

Manuele Bicego

Hidden Markov Models for
Pattern Recognition and Computer
Vision: methodological issues and
applications

Ph.D. Thesis

March 2003

Università degli Studi di Verona
Dipartimento di Informatica

Advisor:
prof. Vittorio Murino

Series N°: TD-01-03

Università di Verona
Dipartimento di Informatica
Strada le Grazie 15, 37134 Verona
Italy

Abstract

Hidden Markov Model (HMM) is an ubiquitous tool for probabilistic modelling of sequential data, whose importance has rapidly increased during the last decade. As shown in several research papers appeared in literature, HMMs are very effective in all the applications involving sequence modelling. Nevertheless there are some drawbacks and some open issues to be addressed; moreover, in some applications, the use of HMMs has not been exhaustively investigated.

This thesis is therefore aimed at reaching a twofold objective: first of all, the individuation and the analysis of the methodological open issues, with the proposal of some original contributions; secondly, the application of the HMM methodology to Computer Vision and Pattern Recognition problems.

From a methodological point of view, the contributions of this thesis regard the following issues: model selection, classification and clustering. In relation to the model selection problem, a formal proof of the equivalence between Gaussian continuous HMMs is first of all presented, able to reduce, in the continuous case, the complexity of the model selection problem. Three original methods are then proposed, able to effectively and accurately find the best model configuration from data.

Concerning the classification issue, some critic considerations about the standard Maximum Likelihood classification scheme are presented. Subsequently an alternative classification scheme is introduced, founded on the similarity-based classification paradigm, able to substantially improve the classification performances.

With regards to the HMM clustering issue, which has been poorly faced in the literature, some contributions are presented in this thesis, mainly regarding the development of new distances between models and new clustering algorithms. Subsequently, an alternative scheme is proposed, founded on the similarity-based representation introduced in the classification context, able to enhance the clustering performances with respect to standard approaches. All of the proposed methodological algorithms developed in this thesis are evaluated and validated by the use of synthetic and real experiments, showing the appropriateness of the proposed methodologies.

From an application point of view, the employment of HMM-based techniques has produced a direct surplus in the following contexts: 2D shape classification, face recognition and video analysis. For the first application, a HMM-based ap-

proach is proposed, able to accurately recognize planar shapes, even in presence of translation, rotation, occlusion, affine projections and noise. Concerning the face recognition problem, a new scheme is introduced, based on HMM and Wavelet coefficients. The obtained classification accuracy outperforms all other results proposed in the literature on standard databases. Finally, a HMM-based clustering approach is used to analyze a static camera video sequence; the proposed method is able to divide the scene into non overlapping regions, each characterized by chromatic, temporal and spatial homogeneity.

The satisfactory results obtained by HMM approaches in these application contexts show the effectiveness and the wide applicability of the proposed techniques.

Contents

Abstract	V
1 Introduction	1
1.1 Motivations	2
1.2 Contributions	3
1.3 Organization of the thesis	4
1.4 Publications	4
2 Hidden Markov Models	7
2.1 Applications of Hidden Markov Models	7
2.2 Fundamentals	8
2.2.1 Types of HMM	9
2.3 The three basic problems of HMM	11
2.3.1 Solution to Problem 1	11
2.3.2 Solution to Problem 2	13
2.3.3 Solution to Problem 3	14

Part I Methodological Issues

3 Model Selection	21
3.1 Introduction	21
3.2 State of the art	22
3.2.1 Deterministic approaches	23
3.2.2 Standard Model Selection criteria	23
3.2.3 Splitting and merging approaches	24
3.2.4 Bayesian approaches	26
3.3 Equivalence between continuous Hidden Markov Models	29
3.4 BIC on initialization	32
3.4.1 Initialization	32
3.4.2 The proposed approach: motivations	32
3.4.3 The Bayesian Inference Criterion	33
3.4.4 The proposed approach	36

3.4.5	Experimental results	36
3.4.6	Conclusions	38
3.5	The Bisimulation approach	40
3.5.1	Motivations	40
3.5.2	Bisimulation	40
3.5.3	The Strategy	43
3.5.4	Experimental results	44
3.5.5	Conclusions	52
3.6	Pruning Model Selection	53
3.6.1	Motivations	53
3.6.2	Model Selection criteria	54
3.6.3	The sequential state pruning strategy	55
3.6.4	Testing	56
3.6.5	Conclusions	61
4	Classification with HMM	63
4.1	Standard classification scheme	63
4.2	Reliability of the classification scheme	64
4.2.1	Experimental evaluation	66
4.3	Classification by similarity	69
4.3.1	State of the Art	72
4.3.2	The Similarity-Based Strategy	73
4.3.3	Results and discussion	76
4.3.4	Conclusions	83
5	Clustering with HMM	85
5.1	Introduction	85
5.1.1	Clustering algorithms	85
5.1.2	Sequential data clustering	86
5.1.3	Chapter outline	88
5.2	Standard HMM-based clustering approach	89
5.2.1	Distance between sequences using HMM	89
5.2.2	Pairwise distance-based clustering algorithms	90
5.2.3	Application to the EEG modelling	91
5.2.4	Experiments	92
5.3	Clustering with the similarity-based representation	94
5.3.1	The proposed approach	95
5.3.2	Experimental results	95
5.3.3	The choice of the representative set \mathcal{R}	98
5.3.4	Future perspectives	99

Part II Applications

6	2D shape classification	105
6.1	Introduction	105
6.2	The strategy	106
6.2.1	Object representation	107
6.2.2	Training	107
6.3	Results and discussion	108
6.3.1	Rotation	109
6.3.2	Occlusion	110
6.3.3	Noise	110
6.3.4	Combined transformations	111
6.3.5	Shearing	112
6.3.6	Results on the second data set	113
6.4	Significance of the classification scheme	115
6.5	Conclusions	116
7	Face recognition	119
7.1	Introduction	119
7.2	The DCT approach	120
7.3	The Wavelet coding	120
7.4	Comparison between DCT and Wavelet coding	122
7.5	Conclusions	124
8	Spatio-temporal segmentation of video sequences	125
8.1	Introduction	125
8.2	The proposed approach	126
8.3	Experimental session	128
8.4	Application to background modelling	133
8.4.1	The background modelling problem	133
8.4.2	The TAPPMOG background modelling	134
8.4.3	The particle filtering tracker	135
8.4.4	The integrated region- and pixel-based approach	137
8.4.5	Results	139
8.5	Conclusions	141
9	Conclusions	143
<hr/>		
Part III Appendix		
<hr/>		
A	Linear dimensionality reduction techniques	149
A.1	Principal Component Analysis (PCA)	150
A.2	Independent Component Analysis (ICA)	151
A.3	Fisher Discriminant Analysis (FDA)	152
	References	153
	Sommario	163

Introduction

Pattern Recognition is an important research area, with a long and fecund history. It encompasses a wide range of information processing systems, all aimed at the resolution of problems of great practical significance. Statistics is the most general and natural framework in which Pattern Recognition problems can be formulated, where both information to be processed and obtained results are expressed in probabilistic form. Among the several statistical techniques proposed in the past, one family has assumed a great importance in the last years: the family of *Probabilistic Graphical Models*. These approaches represent a synergy between the probabilistic and the graph theory. From one hand, graphs permit a real and effective representation of the interrelationship (dependency or independency) occurring between the components of the model (random variables). From the other hand, the probabilistic theory provides a precise mathematical formalization of algorithms and results, allowing their interpretation in a probabilistic context. Some examples of probabilistic approaches belonging to this family are Markov Random Fields [146], Bayesian Networks [95] and Hidden Markov Models [180].

This thesis could be collocated in the aforesaid context, and it concerns the analysis and investigation of the Hidden Markov Model approach for Pattern Recognition and Computer Vision. The Hidden Markov Model (HMM) methodology is a Probabilistic Graphical Model ubiquitously employed for the statistical modelling of sequential data. HMMs can be considered as a stochastic generalization of finite-state automata, where both transitions between states and generation of output symbols are governed by probability distributions [180]. These models are also referred to as Markov sources or probabilistic functions of Markov Chains.

The importance of HMMs has rapidly and impressively grown up only in the last decade, even if they were introduced by Baum and *et al.* in the late 1960s and early 1970s [15, 14, 17, 16, 13]. Hidden Markov Models have universally been used in almost all the applications dealing with sequential data modelling, due to their intrinsic attractive properties. These characteristics were properly and effectively resumed in few sentences by Juang in the introduction of the special issue on HMM in *Vision of the International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)* [115]: “The HMM is a powerful mathematical formalism and yet is computationally straightforward. Most of its related algorithms are linear-

time and intuitive. The model is a complex of thoughts but its reasoning and execution are brilliantly simple. It is a simple complex.”.

1.1 Motivations

As shown in several research papers present in the HMM literature, Hidden Markov Models are very suitable for sequential data modelling. Their application to sequence modelling problems has exponentially grown in the last decade, demonstrating their usefulness and efficacy. Nevertheless, it is impossible to find a technique able to solve *all* problems, and each methodology has its own drawbacks. In this sense, also the HMM methodology presents some defects, and some efforts should be done for fixing them. Moreover, the wide application of these models has revealed the importance of some aspects (e.g. model selection) of this technique disregarded in the past, or to which few attention has been paid.

The first and most important methodological issue, not yet completely solved, is surely the *model selection* problem, regarding the determination of the HMM structure, namely the topology and the number of states. The former regards the possibility of introducing some constraints in the HMM structure, such as forcing the presence or absence of connections between certain states. The latter concerns the determination of the number of states, and is definitely more interesting: it represents the first and fundamental step in the selection of the model, that mainly prevents overtraining situations. Another still unsolved issue derives from the local/greedy behavior of the standard algorithm used to estimate the HMM parameters from training data. This learning procedure, starting from some initial estimates, converges to the nearest local maximum of the likelihood function. The initialization, therefore, crucially affects the obtained model estimate, since the likelihood function is highly multi-modal, and this behavior has a strong influence on the effectiveness of the learning process. It has been demonstrated that appropriate model selection and initialization of the training procedure are crucial for an effective data modelling, and should be carefully addressed for exploiting all the potentialities of this technique.

Another important open issue concerns the classification scheme: the standard HMM-based approach to sequence classification consists of training one HMM for each class, and then using them as class-conditional densities in a standard Bayes classification paradigm. Some questions may arise: is this classification scheme reliable? Does this scheme use all available information? Are alternative schemes possible? At the moment, definitive answers to these questions are missing, this thesis presents a contribute in this sense.

A final question regards the unsupervised classification or clustering of sequences. This topic presents several interesting characteristics, from both a theoretical and an applicative point of view. From a theoretical point of view, two considerations reveal the importance of this issue. First, it is well known that data clustering is inherently a more difficult task if compared to supervised classification, in which classes are already identified, so that a system can be adequately trained. Second, this intrinsic difficulty worsens if sequential data are considered: the structure of the underlying process is often difficult to infer, and typically different length sequences have to be dealt with. The problem of clustering of sequences

represents therefore a real methodological challenge in the Pattern Recognition area, and much effort has been lavished on it by the scientific community.

Moreover, this problem has also a great practical relevance, increasing in importance in recent years, due to its wide applicability in emergent contexts like data mining and bioinformatics problems (like DNA genome modelling and analysis).

These open issues regard methodological aspects of the Hidden Markov Models methodology. Other considerations could be done with concerns to the application context: after reviewing the HMM literature, it has been noted that in some emerging applications, like face recognition, the use of the HMM methodology has not been exhaustively investigated, and some substantial improvements could be obtained with their use.

Starting from these general considerations, the objective of this thesis is therefore twofold: first, investigate some of the presented methodological open issues, with the aim of gaining a better insight into them and finding some original solutions. The second objective is to apply the Hidden Markov Model methodology to Pattern Recognition and Computer Vision problems, showing that these models permit to obtain results that represent relevant contributions also in the specific application domain.

1.2 Contributions

In this section the original contributions proposed in this thesis are summarized. These contributions regard both methodological open issues and applications. All the methodological open issues presented in the previous section are addressed in this thesis. Even if these issues are still not completely resolved, this thesis contributes to get a better insight into these problems, and proposes some original algorithms that clearly improve the state of the art.

More in detail, this thesis largely addresses the model selection and initialization issues. First of all, a formal proof of the equivalence between continuous Gaussian HMMs is derived: this proof permits to reduce, in the continuous case, the problem of selecting the best model (*i.e.* choosing the number of states and number of Gaussians for state) to a simpler one, where only the number of states is needed. Then three original methods are proposed, each one characterized by different features, and all aimed at automatically discovering the most appropriate structure of the model. The first proposed method is directly linked to the initialization step, and results in a very fast approach. The second one makes use of a syntactic equivalence relation, the probabilistic bisimulation [10], and is aimed at reducing an oversized model to a more compact representation. The third one recovers from the drawbacks of standard model selection criteria by performing a sequential pruning learning, able to reduce the impact of the initialization issue and to reduce the needed computational requirements.

Regarding the classification problem, some critic considerations about the reliability of classification results are presented; subsequently, an alternative scheme is proposed, inspired by the similarity-based classification paradigm. This scheme, differently from the standard Bayesian classification approach, makes use of all the available information, resulting in a substantial improvement of the classification accuracy.

Finally, in the present study the use of HMM for unsupervised classification is investigated. The standard approach is summarized, introducing some modifications that permit a more accurate and efficient clustering. Subsequently, an alternative scheme is proposed, founded on the similarity-based representation introduced in the classification context. This scheme permits a great improvements of the clustering results, with respect to the standard approach.

With regards to the applications, the use of Hidden Markov Model methodology has produced relevant results in different application scenarios. First, the use of HMM in the 2D shape classification context is presented: a very robust system is proposed, able to recover from object perturbations as translation, rotation, occlusion, noise and affine projections. Second, the face recognition problem is addressed, proposing a system outperforming all other techniques proposed in literature, using HMM and wavelet features. The last application describes the spatio-temporal segmentation of video sequences, where HMMs are employed for segmenting the scene into regions of chromatic, temporal, and spatial homogeneity. This segmentation is then directly used in a background modelling system, able to recover from sudden not uniform illumination changes in the scene.

1.3 Organization of the thesis

The thesis is divided in an introductory chapter and two main parts. The first chapter formally introduces the HMM methodology, and provides a not exhaustive list of the fields in which HMMs have been successfully employed. The two following main parts regard the methodological issues and applications, respectively.

In the methodological part, Chapter 3 deals with the model selection problem: the state of the art is summarized, and three original methods are proposed. The subsequent Chapter 4 faces the classification problem: after presenting the standard classification rule, some considerations about its reliability are proposed, and an alternative scheme is introduced. Finally, Chapter 5 deals with the clustering task: the standard approach and some variations are presented; an alternative scheme is then proposed, founded on the similarity-based representation paradigm.

Algorithms presented in this part are evaluated using synthetic and real experiments, derived from different applications in Computer Vision and Pattern Recognition, as 2D shape classification, face recognition, EEG segmentation and DNA gene modelling. In the second part of the thesis, with regards to applications, only those systems for which the employment of HMMs produces a surplus in the application context are presented. The others are only briefly explained in the first part, when needed. More in detail, Chapter 6 presents a planar shapes classification system, Chapter 7 addresses the face classification problem with HMMs, and, finally, in Chapter 8, the spatio-temporal segmentation of video sequences using HMM is presented.

1.4 Publications

Some parts of this study have been published in conference proceedings or in international journals. More specifically, the two model selection approaches proposed

in Chapter 3 have been published in [27] and [24], the approach presented in the first part of Chapter 5 in [167], and part of the application described in Chapter 8 in [53]. A preliminary version of the method proposed in Chapter 6 for 2D shape classification appeared in [25].

Other parts of the thesis are still under consideration for publication. More in detail, the alternative classification scheme introduced in Chapter 4 has been submitted to the Pattern Recognition journal [28], the 2D shape classifier described in Chapter 6 to the IEEE Transaction on Pattern Analysis and Machine Intelligence [26], and the HMM-based video segmentation presented in Chapter 8 to the IEEE conference on Computer Vision and Pattern Recognition 2003 [52]. The rest of the thesis is still unpublished: in particular the alternative approach to clustering proposed in Chapter 5 and the the face recognition system introduced in Chapter 7 are works still in progress, which will be submitted when completed.

Hidden Markov Models

This chapter introduces the Hidden Markov Model methodology: the literature is briefly reviewed, applications are described and some technical details are presented.

The literature concerning Hidden Markov Models is very vast. At the risk of unintentional unfairness, the first cited paper is nevertheless the survey that most strongly influenced the author: the Rabiner paper [180], which clearly introduces Hidden Markov Models and their application to speech recognition. Other useful review papers were proposed by Bengio [20], where recent learning algorithms and extensions of the basic model are reviewed, and by Ghahramani [83], in which HMMs are introduced in the context of recent literature on Bayesian Networks [95]. Finally, a very comprehensive list of references on Hidden Markov Models can be found on [41], updated to March 2001.

The chapter is organized as follows: in Section 2.1, a non comprehensive list of applications where HMM were successfully employed is presented. Subsequently, in Section 2.2, HMMs are formally introduced, and the related three common problems are described.

2.1 Applications of Hidden Markov Models

Speech recognition is surely, in order of time, the first and most important application of HMMs. Hundreds of papers appeared on this argument, but, for not unfairly leaving out important works, only few historical papers are reported: [138], [180], and [179], which not only contains information about HMM, but also a very comprehensive review of the speech recognition problem.

In the last decade HMMs were successfully applied to a impressively large number of problems: in the following, a list of applications using HMMs is reported, with some references as example:

- handwriting character recognition: on-line [217,97,136] and off-line [168] recognition;
- computer vision: image classification [145], gesture recognition [69,225], action classification [112], face classification [194,125,70], 2D shape classification [94], texture classification [178] and 3D range object recognition [90];

- signal processing [54];
- finance [190, 188];
- meteorology [188];
- geomagnetism [188];
- neurons signal analysis [182, 39];
- acoustics [184];
- bioinformatics: DNA sequence and protein analysis [68] and identification of ion channel currents [218, 220, 219];
- EEG modelling [174, 175];
- robotics [92];
- communications [129];

This is far from being an exhaustive list, but its aim is simply giving an idea of the several possible applications of this methodology: for a wider list of references, see the very comprehensive [41]. Other examples of applications will be presented in the following chapters of this thesis: in the second part, in particular, some original contributions to HMM applications will be proposed.

2.2 Fundamentals

A discrete-time first-order HMM [180] is a probabilistic model that describes a stochastic sequence $\mathbf{O} = O_1, O_2, \dots, O_T$ as being an indirect observation of an underlying (hidden) random sequence $\mathbf{Q} = Q_1, Q_2, \dots, Q_T$, where this hidden process is Markovian, even the observed process may not be so. Let us briefly recall the concept of Markovian process: a process is said to be Markovian of order p if

$$P(Q_t | Q_{t-1}, Q_{t-2}, \dots, Q_1) = P(Q_t | Q_{t-1}, Q_{t-2}, \dots, Q_{t-p}) \quad (2.1)$$

The process associated with the HMM is Markovian of order one, *i.e.*

$$P(Q_t | Q_{t-1}, Q_{t-2}, \dots, Q_1) = P(Q_t | Q_{t-1}) \quad (2.2)$$

A discrete first-order HMM is formally defined by the following elements:

- A set $S = \{S_1, S_2, \dots, S_k\}$ of (hidden) states. Although these states are hidden, there are some practical applications (especially in speech and handwriting recognition tasks) where some physical significances could be attached to the said states. We denote the state at time t as Q_t .
- A transition matrix $\mathbf{A} = \{a_{ij}\}$, of dimension $k \times k$, where element $a_{ij} \geq 0$ represents the probability of going from state S_i to state S_j :

$$a_{ij} = A(S_i \rightarrow S_j) = P[Q_{t+1} = S_j | Q_t = S_i] \quad 1 \leq i, j \leq k \quad (2.3)$$

Of course

$$a_{ij} \geq 0 \quad \text{and} \quad \sum_{j=1}^k a_{ij} = 1$$

Such a matrix is called a *stochastic matrix*;

- A set $V = \{v_1, v_2, \dots, v_m\}$ of observation symbols: this is the alphabet, and it corresponds to the physical outputs of the process being modelled.
- A $(k \times m)$ emission matrix $\mathbf{B} = \{b(j|S_i)\}$, indicating the probability of emission of symbol v_j from state S_i :

$$b(j|S_i) = P[O_t = v_j | Q_t = S_i] \quad 1 \leq i \leq k, \quad 1 \leq j \leq m \quad (2.4)$$

with

$$b(j|S_i) \geq 0 \quad \text{and} \quad \sum_{j=1}^m b(j|S_i) = 1$$

- An initial state probability distribution $\boldsymbol{\pi} = \{\pi_i\}$,

$$\pi_i = \pi(S_i) = P[q_1 = S_i] \quad 1 \leq i \leq k, \quad (2.5)$$

with,

$$\pi_i \geq 0 \quad \text{and} \quad \sum_{i=1}^k \pi_i = 1$$

An HMM is completely specified by a 5-tuple $\boldsymbol{\lambda} = (S, V, \mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$, and defines a joint probability distribution on the space of hidden and observed sequences, *i.e.*, $P(\mathbf{O} = \mathbf{o}, \mathbf{Q} = \mathbf{q} | \boldsymbol{\lambda})$.

The HMM could also be used as generator model, in order to give an observation sequence $\mathbf{O} = O_1, O_2, \dots, O_T$. This is carried out by following the algorithm [180]:

1. choose an initial state $Q_1 = S_i$ according to the initial state distribution $\boldsymbol{\pi}$;
2. set $t = 1$;
3. chose $O_t = v_j$ according to the symbol probability distribution in state S_i , that is $b(j|S_i)$;
4. transit to a new state $Q_{t+1} = S_j$ according to the state transition probability distribution for state S_i , *i.e.* a_{ij} ;
5. set $t = t+1$; if $t < T$ return to step 3, otherwise terminate the procedure.

2.2.1 Types of HMM

The previous definition specifies *discrete*, *ergodic* and *stationary* Hidden Markov Models. These three characteristics are related to two parameters: the emission matrix (discrete) and the transition matrix (ergodic and stationary). There are different alternatives to them, implying different types of Hidden Markov Models.

With regards to the transition matrix, the ergodic model is the most common type: the HMM has a full state transition matrix, and every state could be reached from every other state of the model (see Fig. 2.1(a)). For some applications, as speech recognition, other topologies have been found to account for the specific problem better than the standard ergodic topology. One example of such a model is the *left-right* HMM [11], presented in Fig. 2.1(b). In this case, the HMM has only a partial state transition matrix: as time increases, the state index increases (or remains the same), *i.e.* the states proceed from left to right. Formally, this implies

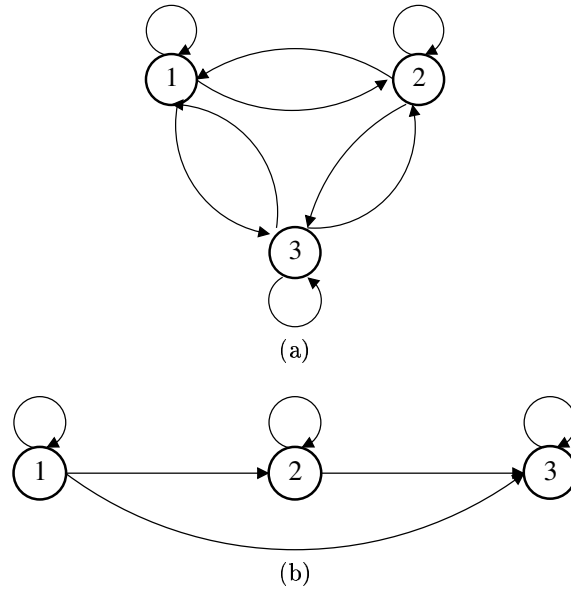


Fig. 2.1. Different topologies: (a) three state ergodic HMM; (b) three state left-right HMM.

$$a_{ij} = 0, \quad \forall j < i$$

Another aspect concerning the transition matrix is its stationarity: if

$$P[Q_{t+1} = S_j | Q_t = S_i] = P[Q_{t+r+1} = S_j | Q_{t+r} = S_i] = a_{ij} \quad \forall r$$

then the Hidden Markov Model is called *stationary* (a_{ij} does not vary over time). Otherwise, HMM is called *non-stationary*.

With regards to the emission probability, it is worth noting that in many interesting applications V is a continuous set, e.g. $V = \mathbb{R}$, and that it is advantageous to use HMMs with continuous observation densities. In this case, instead of a matrix of symbol probabilities \mathbf{B} , for each state S_i we have an emission probability density function (pdf) $b(o|S_i)$, for $o \in V$, and of course with $\int_V b(o|S_i) do = 1$. The most general representation of the pdf is a finite mixture of the form

$$b(o|S_i) = \sum_{m=1}^{M_i} c_{im} \mathcal{F}(o, \boldsymbol{\mu}_{im}, \mathbf{U}_{im}) \quad 1 \leq i \leq k \quad (2.6)$$

where o is the vector being modelled, c_{im} is the mixture coefficient for the m th mixture in state S_i and \mathcal{F} is any log-concave or elliptically symmetric density, with mean vector $\boldsymbol{\mu}_{im}$ and covariance matrix \mathbf{U}_{im} . For real (scalar or vectorial) observations, a very common approach is to model $b(o|S_i)$ as a mixture of Gaussians,

$$b(o|S_i) = \sum_{j=1}^{M_i} c_{ij} \mathcal{N}(o | \boldsymbol{\mu}_{ij}, \boldsymbol{\Sigma}_{ij}). \quad (2.7)$$

In the equation above, $\mathcal{N}(o|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes a Gaussian density of mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, evaluated at o . The observations from state S_i are therefore modelled as samples from a Gaussian mixture with M_i components. In this mixture-based case, for which the adaptation of the Baum-Welch procedure is straightforward [116], \mathbf{B} contains all the mixtures parameters (all the M_i 's, all the $\boldsymbol{\mu}_{ij}$'s, etc.); the HMM, in this case, is completely defined by $\boldsymbol{\lambda} = (S, \mathbf{A}, \boldsymbol{\pi}, \mathbf{B})$.

Although the general formulation of continuous density HMMs could be applied to a wide range of problems, there is another very interesting class of HMMs that seems to be particularly suitable in certain cases, as for example speech recognition or EEG modelling: the autoregressive HMMs [117]. In this case, the observation vectors are drawn from an autoregressive process, and the emission probability $b(O_t|S_i)$ is defined as

$$P(O_t|S_i) = \mathcal{N}(O_t - \mathbf{F}_t \hat{\mathbf{r}}_i, \sigma_i^2) \quad (2.8)$$

where $\mathbf{F}_t = -[O_{t-1}, O_{t-2}, \dots, O_{t-p}]$, $\hat{\mathbf{r}}_i$ is the (column) vector of AR coefficients for the i th state and σ_i^2 is the estimated observation noise for the i -th state. The prediction for the i th state is

$$\hat{O}_t^i = \mathbf{F}_t \hat{\mathbf{r}}_i$$

The order of the AR model is p .

2.3 The three basic problems of HMM

There are three main problems related to the HMM use:

1. Given the HMM $\boldsymbol{\lambda} = (S, V, \mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$, and the observation sequence $\mathbf{O} = O_1, O_2, \dots, O_T$ (with $O_t \in V$, for $t = 1, 2, \dots, T$), compute the marginal probability $P(\mathbf{O}|\boldsymbol{\lambda})$, usually called the *likelihood function*, representing the probability that \mathbf{O} was generated by the model $\boldsymbol{\lambda}$. This is usually defined as the *evaluation problem*.
2. Given $\boldsymbol{\lambda} = (S, V, \mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$, and an observed sequence $\mathbf{O} = O_1, O_2, \dots, O_T$, determine the state sequence $\hat{\mathbf{Q}} = \hat{Q}_1, \hat{Q}_2, \dots, \hat{Q}_T$ (with $\hat{Q}_t \in \{S_1 \dots S_k\}$) that best “explain” the observations, *i.e.* that most probably generated the observation. This is usually called the *decoding problem*.
3. Given a set of L observed strings $\mathcal{O} = \{\mathbf{O}^{(l)}\}$, where $1 \leq l \leq L$, and $\mathbf{O}^{(l)} = O_1, O_2, \dots, O_{T_l}$, adjust the parameters of a HMM $\boldsymbol{\lambda} = (S, V, \mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$ in order to maximize $P(\mathcal{O}|\boldsymbol{\lambda})$. This is usually referred to as the *training problem*.

2.3.1 Solution to Problem 1

Given the HMM $\boldsymbol{\lambda} = (S, V, \mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$, and an observations sequence $\mathbf{O} = O_1, O_2, \dots, O_T$, the goal is the computation of the marginal probability $P(\mathbf{O}|\boldsymbol{\lambda})$. By using marginalization and the Bayes theorem, $P(\mathbf{O}|\boldsymbol{\lambda})$ could be computed as

$$\begin{aligned} P(\mathbf{O}|\boldsymbol{\lambda}) &= \sum_{\text{all } \mathbf{Q}} P(\mathbf{O}, \mathbf{Q}|\boldsymbol{\lambda}) \\ &= \sum_{\text{all } \mathbf{Q}} P(\mathbf{O}|\mathbf{Q}, \boldsymbol{\lambda})P(\mathbf{Q}|\boldsymbol{\lambda}) \end{aligned}$$

Now, assuming statistical independence for observations, we have that

$$\begin{aligned} P(\mathbf{O}|\mathbf{Q}, \boldsymbol{\lambda}) &= \prod_{t=1}^T P(O_t|Q_t, \boldsymbol{\lambda}) \\ &= b(O_1|Q_1)b(O_2|Q_2)\dots b(O_T|Q_T) \end{aligned} \quad (2.9)$$

The other factor $P(\mathbf{Q}|\boldsymbol{\lambda})$ could be computed as

$$P(\mathbf{Q}|\boldsymbol{\lambda}) = \pi_{Q_1} a_{Q_1 Q_2} a_{Q_2 Q_3} \dots a_{Q_{T-1} Q_T}$$

Putting all together, we obtain

$$\begin{aligned} P(\mathbf{O}|\boldsymbol{\lambda}) &= \sum_{\text{all } \mathbf{Q}} P(\mathbf{O}|\mathbf{Q}, \boldsymbol{\lambda})P(\mathbf{Q}|\boldsymbol{\lambda}) \\ &= \sum_{\text{all } Q_1, Q_2, \dots, Q_T} \pi_{Q_1} b(O_1|Q_1) a_{Q_1 Q_2} b(O_2|Q_2) \dots a_{Q_{T-1} Q_T} b(O_T|Q_T) \end{aligned} \quad (2.10)$$

A little thought should convince the reader that the calculation of $P(\mathbf{O}|\boldsymbol{\lambda})$ with (2.10) involves the order of $2T \cdot k^T$ calculation: this computation is unfeasible, and a more efficient procedure is required. Such a method exists, the so-called *forward-backward procedure* [14,17]. This technique is based on two variables, the *forward* variable $\alpha_t(i)$ and the *backward* variable $\beta_t(i)$. The former ($\alpha_t(i)$), is defined as

$$\alpha_t(i) = P(O_1 \dots O_t, Q_t = S_i | \boldsymbol{\lambda}) \quad (2.11)$$

and represents the probability to have observed the sequence $O_1 \dots O_t$ up to time t , and being in state S_i . It is recursively computed by the following formulas

$$\begin{aligned} \alpha_1(i) &= \pi_i b(O_1|S_i) & 1 \leq i \leq k \\ \alpha_{t+1}(i) &= \left[\sum_{j=1}^k \alpha_t(j) a_{ji} \right] b(O_{t+1}|S_i) & 1 \leq t \leq T-1, 1 \leq i \leq k \end{aligned}$$

The *backward* variable is defined as

$$\beta_t(i) = P(O_{t+1} \dots O_T | Q_t = S_i, \boldsymbol{\lambda}) \quad (2.12)$$

and represents the probability to observe the symbols $O_{t+1} \dots O_T$, being in the state S_i at time t . This variable is recursively computed:

$$\begin{aligned} \beta_T(i) &= 1 & 1 \leq i \leq k \\ \beta_t(i) &= \sum_{j=1}^k a_{ij} b(O_{t+1}|S_j) \beta_{t+1}(j) & t = T-1, \dots, 1, 1 \leq i \leq k \end{aligned}$$

$P(\mathbf{O}|\boldsymbol{\lambda})$ is then computed as,

$$P(\mathbf{O}|\boldsymbol{\lambda}) = \sum_{i=1}^k \alpha_t(i)\beta_t(i) \quad \forall t \quad (2.13)$$

By fixing $t = T$ we obtain

$$P(\mathbf{O}|\boldsymbol{\lambda}) = \sum_{i=1}^k \alpha_T(i) \quad (2.14)$$

2.3.2 Solution to Problem 2

Given $\boldsymbol{\lambda} = (S, V, \mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$, and an observed sequence $\mathbf{O} = O_1, O_2, \dots, O_T$, the aim is to determine the “optimal” state sequence that generated \mathbf{O} . Several criteria could be adopted to define the concept of “optimality”. Finding the single best path that maximizes the probability to generate the sequence is the usual one. The goal is therefore to find the state sequence $\hat{\mathbf{Q}} = \hat{Q}_1, \hat{Q}_2, \dots, \hat{Q}_T$, such that

$$\hat{\mathbf{Q}} = \arg \max_{\mathbf{Q}} P(\mathbf{O}, \mathbf{Q}|\boldsymbol{\lambda}).$$

This problem is solved by the *Viterbi algorithm* [222, 79]. This procedure starts by defining the quantity

$$\delta_t(i) = \max_{Q_1 \dots Q_{t-1}} P(Q_1, Q_2, \dots, Q_t = S_i, O_1, O_2, \dots, O_t|\boldsymbol{\lambda}) \quad (2.15)$$

representing the best score (*i.e.* the highest probability) along a single path, at time t , which accounts for the first t observations and ends at state S_i . To retrieve the state sequence, the argument of the δ_i has to be stored for each t and each i : this is obtained using the vector $\psi_t(i)$.

The Viterbi algorithm is then defined by the following recursive steps:

1. Initialization:

$$\begin{aligned} \delta_1(i) &= \pi_i b(O_1|S_i) & 1 \leq i \leq k \\ \psi_1(i) &= \emptyset & 1 \leq i \leq k \end{aligned}$$

2. Recursion:

$$\delta_t(i) = \max_{1 \leq j \leq k} [\delta_{t-1}(j) a_{ij}] b(O_t|S_i) \quad \begin{array}{l} 1 \leq j \leq k \\ 2 \leq t \leq T \end{array}$$

$$\psi_t(i) = \arg \max_{1 \leq j \leq k} [\delta_{t-1}(j) a_{ij}] \quad \begin{array}{l} 1 \leq j \leq k \\ 2 \leq t \leq T \end{array}$$

3. Termination:

$$\begin{aligned} \hat{P} &= \max_{1 \leq i \leq k} [\delta_T(i)] \\ \hat{Q}_T &= \arg \max_{1 \leq i \leq k} [\delta_T(i)] \end{aligned}$$

4. State sequence backtracking:

$$\hat{Q}_t = \psi_{t+1}(\hat{Q}_{t+1}) \quad t = T-1, T-2, \dots, 1$$

2.3.3 Solution to Problem 3

Given a set of L observed strings $\mathcal{O} = \{\mathbf{O}^{(l)}\}$, where $1 \leq l \leq L$, and $\mathbf{O}^{(l)} = O_1, O_2, \dots, O_{T_l}$, assumed to be independent samples taken from a common HMM $\lambda = (S, V, \mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$, the aim is to determine λ . This is the most difficult problem, which is usually solved by adopting the Maximum Likelihood (ML) criterion, that is,

$$\hat{\lambda} = \arg \max_{\lambda} P(\mathcal{O}|\lambda) = \arg \max_{\lambda} \prod_{l=1}^L P(\mathbf{O}^{(l)}|\lambda);$$

The best-known algorithm to implement this ML criterion is the so-called *Baum-Welch re-estimation* technique [15, 14, 17, 16, 13]. This is a particularization of the well known *Expectation-Maximization* (EM) algorithm [57, 226] for ML estimation problems with missing data (the missing data in this case is the hidden sequence \mathbf{Q}).

In order to describe the procedure for the re-estimation of the HMM parameters, two variables have to be introduced:

- $\xi_t(i, j)$: it represents the probability of passing from state S_i at time t to state S_j at time $t + 1$, given the observations and the model, *i.e.*

$$\xi_t(i, j) = P(Q_t = S_i, Q_{t+1} = S_j | \mathbf{O}, \lambda) \quad (2.16)$$

This variable is computed using forward and backward variables as

$$\begin{aligned} \xi_t(i, j) &= P(Q_t = S_i, Q_{t+1} = S_j | \mathbf{O}, \lambda) \\ &= \frac{P(Q_t = S_i, Q_{t+1} = S_j, \mathbf{O} | \lambda)}{P(\mathbf{O} | \lambda)} \\ &= \frac{\alpha_t(i) a_{ij} b(O_{t+1} | S_j) \beta_{t+1}(j)}{P(\mathbf{O} | \lambda)} \\ &= \frac{\alpha_t(i) a_{ij} b(O_{t+1} | S_j) \beta_{t+1}(j)}{\sum_i \sum_j \alpha_t(i) a_{ij} b(O_{t+1} | S_j) \beta_{t+1}(j)} \end{aligned} \quad (2.17)$$

Note that the sum of $\xi_t(i, j)$ over time t could be interpreted as the expected number of transitions from state S_i to state S_j .

- $\gamma_t(i)$: it represents the probability of being in state S_i at time t , given the observation and the model, *i.e.*

$$\gamma_t(i) = P(Q_t = S_i | \mathbf{O}, \lambda) \quad (2.18)$$

This variable is computed as

$$\begin{aligned} \gamma_t(i) &= P(Q_t = S_i | \mathbf{O}, \lambda) \\ &= \frac{P(Q_t = S_i, \mathbf{O} | \lambda)}{P(\mathbf{O} | \lambda)} \\ &= \frac{\alpha_t(i) \beta_t(i)}{\sum_i \alpha_t(i) \beta_t(i)} \end{aligned} \quad (2.19)$$

The variable $\gamma_i(t)$ could be also expressed in terms of the variable $\xi_t(i, j)$, giving

$$\gamma_t(i) = \sum_{j=1}^k \xi_t(i, j) \quad (2.20)$$

In this case too, the sum of $\gamma_t(i)$ over the time t can be interpreted as the expected number of transitions from S_i .

Given those two variables, the re-estimation procedure determines the values of the parameters $\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}$ with the following procedure:

$$\begin{aligned} \bar{\pi}_i &= \text{expected frequency in state } S_i \text{ at time } t = 1 \\ &= \gamma_1(i) \end{aligned} \quad (2.21)$$

$$\begin{aligned} \bar{a}_{ij} &= \frac{\text{expected number of transitions from state } S_i \text{ to state } S_j}{\text{expected number of transitions from state } S_i} \\ &= \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \end{aligned} \quad (2.22)$$

$$\begin{aligned} \bar{b}(v_j|S_i) &= \frac{\text{expected number of times in state } S_i \text{ and observing symbol } v_j}{\text{expected number of times in state } S_i} \\ &= \frac{\sum_{t=1}^{T-1} \gamma_t(i)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad \text{s.t. } O_t = v_j \end{aligned} \quad (2.23)$$

If we define the current model as $\boldsymbol{\lambda} = (S, V, \mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$, and use it to compute the right hand side of (2.21), (2.22) and (2.23), we define the re-estimated model as $\bar{\boldsymbol{\lambda}} = (S, V, \bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\boldsymbol{\pi}})$. It has been shown by Baum and his colleagues that

$$P(\mathbf{O}|\bar{\boldsymbol{\lambda}}) > P(\mathbf{O}|\boldsymbol{\lambda})$$

which means that, at each iteration, the likelihood has increased, until a maximum is reached. The final result of this procedure is the trained HMM, called the *Maximum Likelihood estimate* of the HMM.

Moreover, it has been proved [180] that this formulation could be perfectly casted into the *expectation-maximization* (EM - [57, 226]) framework. Finally, as this problem is an optimization problem, some gradient descent techniques have been employed [138], yielding nevertheless solutions comparable to those of standard re-estimation procedure above presented.

This technique is quite effective and really fast: typically the algorithm converges after about ten iterations. The main drawback is its local behavior, implying

the convergence to the nearest local maximum. Since the likelihood is highly multimodal, the initial conditions could crucially affect the effectiveness of the learning. Another severe limitation is the fact that re-estimation formulas are based on computation of frequencies, or expected times, of events. This implies that a big training set is indispensable in order to obtain reliable estimate of the HMM parameters.

Methodological Issues

Summary

This part contains the description of the methodological contributions to the Hidden Markov Models problem developed in this thesis. More in detail, Chapter 3 introduces the Model Selection problem, regarding the question of choosing some quantitative characteristics of the model, as number of states, type of connections etc. A formal proof on the equivalence between Gaussian continuous HMMs is proposed: this permits to reduce the problem of selecting the best model (i.e. choosing the number of states and the number of Gaussians for state), to a simpler one, where only the number of states is searched for. Subsequently, three original methods are proposed, each one characterized by different features: the first method presented is directly linked to the initialization phase, resulting in a very fast approach; the second one proposes to use a syntactic method, called probabilistic bisimulation, to solve the problem; the third one overcomes drawbacks of standard model selection techniques, improving also the initialization stage.

Subsequently, Chapter 4 deals with the classification scheme intrinsically linked with HMM use: first the standard classification scheme is introduced, together with some quantitative considerations about reliability of this standard rule; subsequently, a new classification scheme is proposed, inspired by the classification by similarity paradigm, able to really improve performance of HMMs. This scheme introduces a new representation space, where sequences are featured by pairwise distances, allowing optimal discrimination with standard point (not sequence) classification techniques.

Finally, in Chapter 5 the clustering problem is addressed: clustering of sequences using HMMs was poorly addressed in the literature, only few papers appeared on this topic. Here two new approaches are introduced: the first represents a slight modification of the standard approach, aimed at giving a more understandable and intuitive measure of distance between HMMs. The second is an alternative scheme, based on the similarity representation introduced in the classification context, able to reduce the complex problem of clustering of sequences to a simpler and more addressable problem of clustering of points, for which numerous methods have appeared.

Model Selection

3.1 Introduction

The core entities of the statistical approach to the Pattern Recognition are represented by the probability density functions (pdf): these quantities are typically unknown, and should be estimated from data. There are several tools used to model an arbitrarily complex pdf, some examples are Gaussian Mixture Models (GMM [154]) and Hidden Markov Models. In the last years, standard techniques have been established for fitting a model to data, such as Expectation-Maximization (EM) [57, 226] or minimum least square methods. These techniques assume the “size” of the fitting model known, finding the most appropriate values for the model parameters. The problem of finding the best model size is usually referred to as the *model selection* problem, and it represents a crucial step in the modelling issue. Its importance could be understood by considering the simple problem of approximating a set of points with a polynomial curve. Once decided the degree of the polynomial, well established techniques for finding its coefficients are available (e.g. least mean square method). The choice of the degree, that represents in this case the model selection issue, is critical: if a too small degree is chosen, the model is not able to capture the significant behavior of the curve; on the other hand, the choice of a too large degree results in a too much detailed representation, unable to adequately generalize, as the model tries to assimilate all the details, also those derived from noise.

In the HMM case, the model selection issue is related to the the determination of the topology and the number of states k . The former aspect concerns the possibility of introducing some constraints in the HMM structure, such as forcing the presence or absence of connections between certain states. The topology is often dependent by the specific application addressed, and typically chosen *a priori*: great advantages could be derived from such tailored topologies, as circular HMM for shape classification [7] or lexeme HMM for handwriting recognition [144], or HM-Network for speech recognition [208]. Nevertheless, in a strict sense, there is actually no selection or search for the best topology. The topology is in fact designed and chosen *ad hoc*, depending on the particular application context.

The second issue is more interesting, and concerns the determination of the number of states: this represents the first and fundamental step in the selection of

the model. Some aspects have to be considered; the first is the complexity of the modelled class: if a large variation between classes is present (consider as example the class of “.” and the class of “W” in the handwriting character recognition), it is obvious that the size of the model could not be the same for all classes, but it has to change, in order to adapt itself to the complexity of the class modelled. On the other hand, too large a model could produce a situation of “overfitting”, where the model is too specifically learned on the training patterns, and is not able to generalize to other test patterns. Choosing accurately the model size for each class is therefore crucial in order to obtain a good modelling.

It is well known that the model selection problem can not be addressed by the ML criterion [75]. The reason for this is that the models are nested, *i.e.*, an HMM with fewer states can always be seen as a particular case of a larger model. Then, the maximized likelihood is a non-decreasing function of the number of states and cannot be used as a model selection criterion. To overcome this problem, typical model selection criteria, as *Bayesian inference criterion* (BIC) [196] - deeply detailed in the following, adds a penalty term to the likelihood, discouraging larger models.

In the case of HMMs with emission densities modelled by Gaussian mixtures, the model selection involves also the choice of the number of components at each state, M_1, \dots, M_k . In this case, nevertheless, there is an additional non-identifiability issue. For example, consider an HMM with two states, such that one of the states has a two-component mixture emission density, and the other state has a single-component Gaussian emission density. This HMM is equivalent to another one with three states characterized by single-Gaussian emission densities. This fact could be generalized: in Section 3.3 it is shown that, for each HMM with more than one Gaussian per state there is another HMM, with more states but only one Gaussian per state, that is equivalent to it. Therefore, the search could be restricted only to the number of states, without any care about the number of Gaussians for state.

The rest of the chapter is organized as follows: in Section 3.2, the state of the art is detailed, together with a brief introduction about Bayesian Theory, while in Section 3.3, the proof of equivalence between continuous HMMs with different number of Gaussians per state is presented. The next three Sections deal with original contributions of the author in the field of model selection, each one featured by different characteristic: the first method, proposed in Section 3.4, is linked to the initialization phase, resulting in a very fast approach; the second one, detailed in Section 3.5, proposes the use of a syntactical method to solve the problem; the last, explained in Section 3.6, overcomes the drawbacks of standard model selection techniques, improving also the initialization stage.

3.2 State of the art

Despite its importance, the state of the art of the HMM model selection problem is quite poor: the mostly used solution is still to fix *a priori* the number of states and the connectivity of transitions, with the help of some heuristic knowledge (for example, one can try with different number of states and look for the best behavior [94]).

The state of the art could be divided in four categories: *deterministic approaches*, where the topology and number of states are merely derived from considerations about specific applications, *standard model selection approaches*, where standard model selection criteria are applied to the HMM context, *splitting and merging approaches*, where the best model is obtained by successive split or merge operations, and *Bayesian approaches*, where the optimal model is determined by using the Bayesian Theory.

3.2.1 Deterministic approaches

These approaches are characterized by a substantial use of the application knowledge, and typically are not employable in different contexts. Some examples are [92, 136, 230]. In [92] HMMs are used for the prediction and the analysis of sensor information recorded during robotic telemanipulation tasks. Each state is then associated with one of the process subtask: the size of the model is therefore determined by the complexity of the process. The transition probabilities encode the ease with which the operator completes the task.

In [136], the authors propose to determine the topology by a *data-driven* approach, in the case of on-line Korean handwriting recognition. Starting from the observation that in the Korean script each sample could be represented as a combination of straight lines, the exploited idea is to assign one state to each straight line class, determined using a clustering approach. A left-to-right topology is employed, and a parameter tying scheme is proposed, whose aim is to avoid the excessive growth of the number of states.

In [230], the authors compared three different approaches to the choice of the length of the HMM for single handwritten character recognition systems: the standard fixed length models, the *Bakis-model*, where the number of states is proportional to the average number of observations of the corresponding training samples, and the *Quantile model*, where the number of states of the HMM is defined as a specified quantile of the character length histogram. The system was deeply tested on isolated words recognition task.

The major drawback of these approaches is that the said methods are very specifically tailored for particular applications, and cannot be used or exported in other contexts.

3.2.2 Standard Model Selection criteria

The standard model selection criteria approaches addressed the problem of determining the best structure of HMM by using standard model selection criteria, *i.e.* criteria that are not finely tuned for HMMs, but are general, and employable in different contexts.

A method that could be used to estimate the number of states is the so called *Cross validation* (CV - [207]): this method is computationally heavy and does not use the available data efficiently. In CV, the observed data is split in two subsets: one becomes the *training set*, the other is called *test set* (the splitting strategy depends on the specific details of the CV technique chosen); different models are then obtained using only the training set (*e.g.*, varying the model

structure) and the one showing best performance on the test set is chosen. CV represents a widely applied method for estimating parameters values (as the K parameter in the K -Nearest Neighbor classification), and is also employed to obtain statistically reliable evaluation of system performances.

More specifically linked to the HMM issue, in [144], the *Bayesian inference criterion* (BIC) [196] is used, in order to determine the number of states or the number of Gaussians for state for continuous HMMs. The topology employed is fixed, and consists in a left-right structure, with an absorbing state allowing the handling of incomplete patterns. The system is tuned for handwriting character recognition, and is compared to application-specific heuristic approaches in terms of resulting model size.

In [91] the problem of choosing the appropriate number of states in speech recognition task is addressed using *Bayesian inference criterion* (BIC) [196] and *Minimum Description Length* (MDL) [185] criteria, together with some heuristics. This approach is tested on simple alphadigits recognition.

3.2.3 Splitting and merging approaches

This class of approaches is characterized by the same idea: starting from an inappropriate but simple model, the correct model is determined by successively applying an operation that modifies the structure, until some stop criterion is reached. The difference between *splitting* and *merging* approaches is evident: the former starts from a concise model and grows it by splitting its states, while the latter starts from a general oversized model and shrinks it by merging or pruning its states.

To the former class belongs the approach proposed in [101], where the problem is addressed by using a splitting strategy together with the Akaike's information criterion (AIC) [5]. This criterion is based on the Kullback-Leibler discrepancy [131, 130], a measure of distance between probability distributions. The proposed approach is incremental, and determines the structure by adding states (by splitting) and transitions to an initial simple model; the process is ended when the AIC criterion reaches a minimum. The state to be split is chosen by measuring the "badness" of each state, measured with the probability to stay in that state times the entropy of the state. The transitions to be added are chosen by evaluating the increase in the likelihood produced by each connection potentially added, and choosing the transition that, if added, maximally increases the likelihood: an approximated procedure is proposed in order to make the likelihood computation feasible. The approach is tested on artificial data and real phone recognition problem.

Two historical and widely cited approaches are [208] and [199]. The former [208] was proposed in the 1992, and consists in a successive state splitting algorithm used in the context of speech recognition for efficient allophone modelling. This method is strictly tuned for Hidden Markov Networks (HM-Net), an *ad hoc* topology specifically designed for speech recognition, illustrated in Fig. 3.1. The algorithm chooses to split the state that presents the larger variability, measured by analyzing the state Gaussians means and variances. The chosen state could be split in the temporal or in the contextual domain, as illustrated in Fig. 3.2, depending on which

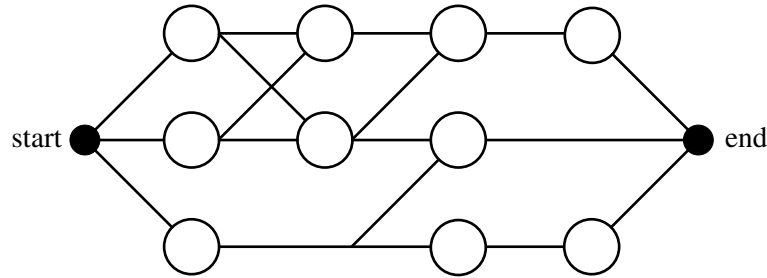


Fig. 3.1. The architecture of the HM-Net.

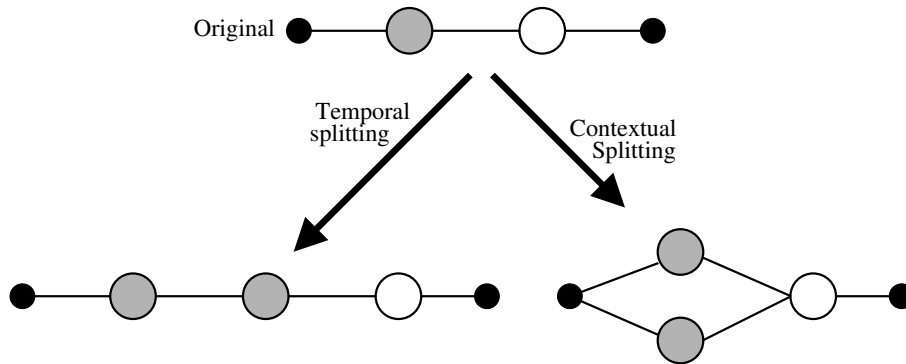


Fig. 3.2. Two types of state splitting proposed in [208].

domain splitting results in the maximum likelihood. The problem of this method is that it works properly only for well defined training topologies, or speaker dependent data. This method, in fact, splits the most “variable” state that, in case of speaker independent data, reflects the speaker’s variability, rather than coarticulations or temporal effects.

This drawback is resolved in [199], where a similar approach is proposed, also aimed at estimating the architecture of a Hidden Markov Network for the speech recognition problem. The approach calculates for each state the expected likelihood gain obtained by the splitting (in both temporal and contextual domains) and splits the state with the highest expected likelihood gain.

A pruning strategy for HMM topology estimation is proposed in [216], in particular aimed at the estimation of the number of states of discrete HMM. The proposed algorithm iteratively prunes state transitions, by training one HMM for each potential pruned connection, and choosing the topology that presents the maximum likelihood. The pruning procedure stops when there is a sudden reduction in the likelihood. The major drawbacks of this approach are the large computational burden required (at each iterations, all potential pruned HMMs are trained) and the empirical content of the stopping criterion.

3.2.4 Bayesian approaches

This section describes some interesting approaches to the model selection issue for Hidden Markov Models, which make use of the Bayesian Theory: this theory is briefly introduced in the following section.

Introduction to Bayes Theory

The Bayes theory represents a particular way of formulating and dealing with statistical decision problems. It is able to formalize the *a priori* knowledge on the problem, and to combine it with the available observations, in order to obtain an optimal (in some sense) decision criteria. In the literature, several books deal with this topic (some examples are [22], [187], and [23]); nevertheless, it is worthwhile to cite the soon available but still unpublished [75], the book that mainly introduced the author into the basics of the Bayesian Theory¹.

In order to understand the Bayesian approach to the parameter estimation problem, let us introduce the difference between the *Maximum Likelihood* (ML) approach (as Baum-Welch re-estimation), and the *Bayesian* approach. Given a set $\mathcal{Y} = \{y_1 \cdots y_n\}$ of observations (supposed independent, identically distributed (i.i.d.)) derived from a fixed but unknown probability distribution $P(y|M)$, the goal is to infer the model M from the data. Supposing that there are different classes M_i from which to choose the model M : M_i could represent different architectures of the model (for example, different topologies, in the HMM case).

Before dealing with the comparison between model architectures M_i , let's introduce how each model is characterized using the two approaches. Fixed a model M_i , the *Maximum Likelihood (ML)* approach finds the parameters $\hat{\theta}_i^{(ML)}$ of the model M_i that maximize the log likelihood $\mathcal{L}(\theta_i) = \log P(\mathcal{Y}|\theta_i)$, *i.e.*

$$\hat{\theta}_i^{(ML)} = \arg \max_{\theta_i} \{\log P(\mathcal{Y}|\theta_i)\} \quad (3.1)$$

In other words, the ML approach characterizes the model M_i by a *point estimate* of its parameters θ_i .

The simplest model selection criteria that intuitively could be used is to compute the ML-estimation parameters $\hat{\theta}_i^{(ML)}$ for all classes M_i , and to choose the class M_{ML} for which the likelihood $P(\mathcal{Y}|M_{ML})$ is maximal. Unfortunately, this criterion could not be used for two reasons: first, it is well known that ML approach tends to overfit the data; secondly, ML prefers complex models, since they have more parameters and fit better the data.

The Bayesian approach provides, in principle, a solution to these problems. The basic idea under this approach is the following: instead of characterizing each model M by computing a *point estimation* of the parameters, this methods computes, for each model class M_i , the *entire* posterior probability $P(M_i|\mathcal{Y})$. In the Bayes theory, the posterior probability $P(M_i|\mathcal{Y})$ represents all the information on M_i derivable from the observations \mathcal{Y} , and is obtained by considering the parameters θ as random variables, and integrating them out. Successively, the model

¹ Thanks to Mario A.T. Figueiredo that gave me a chance to read the draft version.

selection phase, *i.e.* the choice between different classes of models, is performed by comparing directly the posterior distribution $P(M_i|\mathcal{Y})$.

More in detail, the posterior probability $P(M_i|\mathcal{Y})$ is computed using the Bayes rule:

$$P(M_i|\mathcal{Y}) = \frac{P(\mathcal{Y}|M_i)P(M_i)}{P(\mathcal{Y})} \quad (3.2)$$

where $P(\mathcal{Y}|M_i)$ is again the likelihood or *evidence*, and $P(M_i)$ is the prior probability, resuming the *a priori* knowledge on model M_i . The entire posterior probability $P(M_i|\mathcal{Y})$ is obtained by considering the parameters θ_i as random variables, and integrating them out from the formula:

$$\begin{aligned} P(M_i|\mathcal{Y}) &\propto P(M_i)P(\mathcal{Y}|M_i) \\ &= P(M_i) \int P(\mathcal{Y}, \theta_i|M_i)d\theta_i \\ &= P(M_i) \int P(\mathcal{Y}|\theta_i, M_i)P(\theta_i|M_i)d\theta_i \end{aligned} \quad (3.3)$$

where $P(M_i)$ is the *model prior*, summarizing the *a priori* knowledge on the model M_i , and $P(\theta_i|M_i)$ is the *parameter prior*, summarizing the knowledge about parameters θ_i given a fixed model M_i .

The simplest way to use this theory is the following: assuming that the posterior distribution $P(M_i|\mathcal{Y})$ is unimodal, it could be approximated by its maximum $P(\hat{\theta}_i^{(MAP)}|\mathcal{Y})$, where

$$\hat{\theta}_i^{(MAP)} = \arg \max_{\theta_i} \{P(M_i|\mathcal{Y})\} \quad (3.4)$$

In this case, the Bayesian estimate degenerates to a point estimate, as in the ML case; therefore a similar model selection rule could be used, called *Maximum A posteriori* (MAP) rule. This rule chooses the model M_{MAP} that maximizes the posterior probability $P(\hat{\theta}_i^{(MAP)}|\mathcal{Y})$ over all model M_i , computed using Bayes theorem:

$$\begin{aligned} M_{MAP} &= \arg \max_i P(\hat{\theta}_i^{(MAP)})P(\mathcal{Y}|\hat{\theta}_i^{(MAP)}) \\ &= \arg \max_i \{\log P(\mathcal{Y}|\hat{\theta}_i^{(MAP)}) + \log P(\hat{\theta}_i^{(MAP)})\} \end{aligned} \quad (3.5)$$

where the passage to the logarithm function does not affect the computation of the optima, but results in a more manageable function. Looking at the eq. (3.5), observe that the MAP approach maximizes the likelihood of the data plus a regularization term: this term typically assigns a penalty to the model size, increasing for larger models. In other words, by the use of a prior probability, the Bayesian approach is able to penalize complex models, preventing overfitting and allowing optimal structure determination.

Obviously, in this simplified MAP case, not the whole potentialities of the Bayesian approach are used, especially knowing that, in real applications, the posterior probability is typically highly multimodal, and major utility could be derived by using full Bayesian approach. Nevertheless, the major drawback of this

approach is that the computation of the integral in (3.3) is intractable, even for very simple cases (e.g., factor analysis, see [31]). Different approaches have been proposed to approximate it, mainly falling in three general categories: Markov Chain Monte Carlo methods [86, 152], large sample methods (as Laplace approximation, exemplified in the Section 3.4.3) [183], and variational methods [106, 105, 84].

Bayesian Model Selection for Hidden Markov Models

The most famous Bayesian approach to model selection for Hidden Markov Model is without doubts the algorithm proposed by Stolcke and Omohundro in [205, 206], where a merging state strategy was implemented in order to maximize the posterior probability $P(M_i|X)$ of the model M_i , given the data X . The posterior probability $P(M_i|X)$ is computed, as explained in the previous section, using the Bayes rule

$$P(M_i|X) \propto P(M_i)P(X|M_i) \quad (3.6)$$

The merging strategy starts from the most specific model, that reproduces exactly all sequences of the training set: one parallel path for each sequence, and, in each path, one state for each symbol. Then, successive merging operations are applied to this model, where the states to be merged are those maximizing the $P(M_{i+1}|X)$; the merging algorithm is stopped when the posterior $P(M_i|X)$ reaches a local maximum. Several approximations were introduced, in order to make the approach feasible: in particular, the likelihood $P(X|M_i)$ is computed only on the Viterbi path, and it is assumed that the merging operation does not affect the Viterbi path.

Another interesting Bayesian approach for Hidden Markov Model is proposed in [33, 35], where an entropic prior is used to penalize the likelihood, discouraging low entropy parameters. The author introduces a modified version of the standard EM algorithm, able to drive to zero weakly supported parameters, skeletonizing the model and concentrating evidence on surviving parameters. This pruning strategy guarantees the increasing of the posterior probability at each step, and results in compact and sparse models, with interpretable states. The approach was applied to several real problems, as music, handwriting, video time-series, but no to synthetic examples. A similar but more formally justified idea was used in [73] for mixture of Gaussians, where an improper Dirichlet prior and a modified EM were used to enhance the annihilation process.

One variational approach to Bayesian HMM model selection was proposed by Mackay in [151]: in this paper, the goal is to approximate the entire posterior probability, using an *ensemble*. This approximation is obtained by maximizing a variational free energy, which measures the relative entropy between the approximating ensemble and the true distribution. Using the Neal and Hinton [159] observation that the EM algorithm can be viewed as a variational free energy minimization method, they propose an EM-based algorithm to find the optimal approximate posterior distribution, using only the assumption that the approximating function (the so-called *ensemble*) is separable in the HMM parameters \mathbf{A} , \mathbf{B} , $\boldsymbol{\pi}$ and in the unknown state sequence S . Even if very interesting, this approach was not implemented, remaining only a theoretical formulation.

Other Bayesian approaches were recently proposed: in [188], a reversible jump Markov Chain Monte Carlo scheme was used to approximate parameters and number of states, and tested on finance, meteorology and geomagnetism data. In [29], the HBIC (HMM BIC) criterion was introduced, obtained by approximating the Bayesian integral (3.3) with Laplacian approximation: this results in a criterion that takes into account the complexity of the HMM problem, adding more terms in the BIC computation; instead of using the same prior for all parameters, in this approach different priors are used, determined by hand, by fitting them to the data.

3.3 Equivalence between continuous Hidden Markov Models

In this section, we show that, given an HMM λ with k states, where the emission probability of each state S_i is a mixture of (univariate or multivariate) Gaussians, each Gaussian having parameters θ_{im} ,

$$b(O|S_i) = \sum_{m=1}^{M_i} c_{im} \mathcal{N}(O|\theta_{im}) \quad (3.7)$$

there is another HMM λ' with $k' = \sum_{i=1}^k M_i$ states, with only one Gaussian for state, that is equivalent to λ . Here, equivalence is understood in a likelihood sense, that is, $P(\mathbf{O}|\lambda) = P(\mathbf{O}|\lambda')$, for any sequence $\mathbf{O} = O_1, O_2, \dots, O_T$.

First we will describe how the λ' model is built; subsequently we will show that the two models are equivalent. Given $\lambda = (S, \mathbf{A}, \boldsymbol{\pi}, \mathbf{B})$, the equivalent model $\lambda' = (S', \mathbf{A}', \boldsymbol{\pi}', \mathbf{B}')$ is defined as follows:

- **New states:** we split each state S_i into M_i states, one for each of the M_i Gaussians of the mixture of S_i . Thus we obtain $k' = \sum_{i=1}^k M_i$ states and

$$S' = \{S'_1, \dots, S'_{k'}\} = \{S'_{11}, \dots, S'_{1M_1}, S'_{21}, \dots, S'_{2M_2}, S'_{31}, \dots, S'_{kM_k}\}, \quad (3.8)$$

where we have introduced the double index notation in which S'_{im} corresponds to the m -th Gaussian of the original state S_i .

- **Emission probabilities:** naturally, the emission probability of state S'_{im} is the corresponding Gaussian

$$b'(O|S'_{im}) = \mathcal{N}(O|\theta_{im}) \quad (3.9)$$

- **State transition probability:** using the double index notation, where

$$A'_{ik, jm} = P(Q_{t+1} = S'_{jm} | Q_t = S'_{ik}) \quad (3.10)$$

denotes the probability of going from state S'_{ik} to state S'_{jm} , we set

$$A'_{ik, jm} = A_{ij} c_{jm}, \quad (3.11)$$

where c_{jm} is the mixing weight of the m -th component from the original state S_j . Notice that $A'_{ik, jm}$ does not depend on k and that, as required,

$$\sum_{jm} A'_{ik, jm} = \sum_{j=1}^k \sum_{m=1}^{M_j} A'_{ik, jm} = \sum_{j=1}^k A_{ij} \sum_{m=1}^{M_j} c_{jm} = 1.$$

- **Initial state probability:** similarly to the previous definition, we set

$$\pi'(S'_{jm}) = \pi(S_j) c_{jm} \quad (3.12)$$

which is also clearly normalized.

The proof of the equivalence between the two HMMs uses the *forward-backward* procedure (see, e.g., [180]), the standard technique for computing $P(\mathbf{O}|\boldsymbol{\lambda})$. This technique is based on the *forward* variables $\alpha_t(S_i)$, defined as

$$\alpha_t(S_i) = P(O_1, \dots, O_t, q_t = S_i | \boldsymbol{\lambda}) \quad (3.13)$$

which are iteratively computed according to

$$\alpha_1(S_i) = \pi(S_i) b(O_1 | S_i) \quad (3.14)$$

$$\alpha_{t+1}(S_i) = b(O_{t+1} | S_i) \sum_{j=1}^k \alpha_t(S_j) A_{ji} \quad (3.15)$$

Given the sequence $\mathbf{O} = O_1, \dots, O_T$, $P(\mathbf{O}|\boldsymbol{\lambda})$ is computed by marginalization,

$$P(\mathbf{O}|\boldsymbol{\lambda}) = \sum_{i=1}^k P(O_1, \dots, O_T, Q_T = S_i | \boldsymbol{\lambda}) = \sum_{i=1}^k \alpha_T(S_i) \quad (3.16)$$

With the goal of showing that $P(\mathbf{O}|\boldsymbol{\lambda}) = P(\mathbf{O}|\boldsymbol{\lambda}')$, let us rewrite $P(\mathbf{O}|\boldsymbol{\lambda}')$ as

$$P(\mathbf{O}|\boldsymbol{\lambda}') = \sum_{i=1}^{k'} \alpha_T(S'_i) = \sum_{i=1}^k \sum_{m=1}^{M_i} \alpha_T(S'_{im}) \quad (3.17)$$

that is, using the double index notation introduced in (3.8). Let us also define

$$\alpha'_T(S_i) = \sum_{m=1}^{M_i} \alpha_T(S'_{im}) \quad (3.18)$$

Clearly, if we show that, for $i = 1, \dots, k$,

$$\alpha_T(S_i) = \alpha'_T(S_i) \quad (3.19)$$

then, we will be able to conclude, as desired, that

$$\underbrace{\sum_{i=1}^k \alpha_T(S_i)}_{P(\mathbf{O}|\boldsymbol{\lambda})} = \sum_{i=1}^k \alpha'_T(S_i) = \sum_{i=1}^k \sum_{m=1}^{M_i} \alpha_T(S'_{im}) = \underbrace{\sum_{i=1}^{k'} \alpha_T(S'_i)}_{P(\mathbf{O}|\boldsymbol{\lambda}')}.$$

We will now show (3.19) by induction on the length T of the sequence \mathbf{O} .

- We start with $T = 1$. From (3.14), we know that

$$\alpha_1(S_i) = \pi(S_i) b(O_1|S_i) = \pi(S_i) \sum_{m=1}^{M_i} c_{im} \mathcal{N}(O_1|\theta_{im}) \quad (3.20)$$

Now, we can also write

$$\begin{aligned} \alpha'_1(S_i) &= \sum_{m=1}^{M_i} \alpha_1(S'_{im}) = \sum_{m=1}^{M_i} \pi'(S'_{im}) \mathcal{N}(O_1|\theta_{im}) \\ &= \sum_{m=1}^{M_i} \pi(S_i) c_{im} \mathcal{N}(O_1|\theta_{im}) = \pi(S_i) \sum_{m=1}^{M_i} c_{im} \mathcal{N}(O_1|\theta_{im}) \end{aligned}$$

where the first equality is (3.18), the second one is (3.14), the third results from the definitions of the model λ' (3.9) and (3.12). Then we have shown that $\alpha_1(S_i) \equiv \alpha'_1(S_i)$.

- To show the recursion, we have to prove that

$$\alpha_T(S_i) = \alpha'_T(S_i) \Rightarrow \alpha_{T+1}(S_i) = \alpha'_{T+1}(S_i) \quad (3.21)$$

Invoking (3.15), we can write

$$\alpha_{T+1}(S_i) = \left[\sum_{j=1}^k \alpha_T(S_j) A_{ji} \right] \left(\sum_{m=1}^{M_i} c_{im} \mathcal{N}(O_{T+1}|\theta_{im}) \right) \quad (3.22)$$

Also, by using (3.18), and again (3.15), we have

$$\begin{aligned} \alpha'_{T+1}(S_i) &= \sum_{m=1}^{M_i} \alpha_{T+1}(S'_{im}) \\ &= \sum_{m=1}^{M_i} \sum_{j=1}^k \sum_{\ell=1}^{M_\ell} \alpha_T(S'_{j\ell}) A_{j\ell, im} \mathcal{N}(O_{T+1}|\theta_{im}) \\ &= \sum_{m=1}^{M_i} \sum_{j=1}^k \sum_{\ell=1}^{M_\ell} \alpha_T(S'_{j\ell}) c_{im} A_{ji} \mathcal{N}(O_{T+1}|\theta_{im}) \\ &= \sum_{m=1}^{M_i} c_{im} \mathcal{N}(O_{T+1}|\theta_{im}) \sum_{j=1}^k A_{ji} \sum_{\ell=1}^{M_\ell} \alpha_T(S'_{j\ell}) \\ &= \left(\sum_{m=1}^{M_i} c_{im} \mathcal{N}(O_{T+1}|\theta_{im}) \right) \sum_{j=1}^k A_{ji} \alpha'_T(S_j) \end{aligned} \quad (3.23)$$

where the third equality results from (3.11), and the last one from (3.18). Finally, comparing (3.23) with (3.22) clearly shows that the implication in (3.21) is true.

This concludes our proof that $P(\mathbf{O}|\lambda) = P(\mathbf{O}|\lambda')$.

3.4 BIC on initialization

In this section a simple but fast approach to the model selection problem is presented, able to reduce computational burden required by standard model selection techniques. The proposed approach is directly linked to the initialization issue of the training process, so we first introduce this issue.

3.4.1 Initialization

As explained in Section 1.1, the initialization of the training procedure crucially affects the effectiveness of the obtained model parameters, as the learning procedure is a local optimization strategy, and the likelihood function is highly multi-modal. Careless initializations could lead to poor estimates of the model, and this behavior strongly affects the model order selection criteria. A typical solution, used for discrete HMM but deleterious for continuous HMMs, is to use several random initializations and choose as final estimate the one with the highest likelihood.

An alternative approach is to perform a preliminary clustering of the coefficients, using for example a Gaussian Mixture Model (GMM) [154] clustering, in order to initialize the emission matrix of the HMM before starting the training process. In larger detail, given a set of sequences $\{\mathbf{O}^i\} = \{O_1^i \dots O_{T_i}^i\}$, the initialization phase proceeds as follows:

1. Consider the set $D = \{O_1^1, O_2^1, \dots, O_{T_1}^1, O_1^2, O_2^2 \dots\}$, that is the set of values of the *unrolled* sequences; each sequence is considered as a set of scalar values (no matter in which order the coefficients appear).
2. Cluster the set D into k clusters using a GMM clustering approach, *i.e.*, fitting the data using k Gaussian distributions, where k is the number of states of the HMM; the Gaussian parameters are estimated by an EM-based [57, 226] method.
3. The mean and variance of each cluster are then used to initialize the Gaussian of each state, with a direct correspondence between clusters and states.

3.4.2 The proposed approach: motivations

Standard methods for model selection address the problem by training several models, with different structures, and then choosing the model that maximizes a certain selection criterion. Although these approaches perform rather accurately, they require one model training for each model structure, thus involving a considerable computational burden.

In order to reduce the computational load, in our approach the model selection issue is addressed in the initialization phase. In particular, the choice of the model is determined by a model selection analysis of the GMM clustering phase: the number of states of the HMM is set as the number of Gaussians of the mixture that best fits the unrolled sequence. Only one HMM training session is therefore needed, with a noticeable reduction of the computational load. It is worth noting that this model selection scheme determines the model that best fits the *unrolled* sequence: in this sense this is a coarse model selection scheme, as only the sequence values are considered and not the order in which they appear. The dynamics of

the sequence, *i.e.*, the way in which these segments are ordered, is thus encoded into the transition matrix.

To choose the GMM model that best fits the data, the *Bayesian Inference Criterion* (BIC) approach [196] is adopted: this Bayesian criterion derives from the Bayes Theory, presented in Section 3.2.4, and is detailed in the following section.

3.4.3 The Bayesian Inference Criterion

As explained in Section 3.2.4, the goal of the Bayesian approach, given a model M_i and the data set D , is to estimate the *entire* posterior probability $P(M_i|D)$, by integrating out the parameters θ_i . The posterior is computed as

$$P(M_i|D) \propto P(M_i)P(D|M_i) \quad (3.24)$$

where $P(M_i)$ is the model prior, and $P(D|M_i)$ is called the *integrated likelihood*, determined by

$$P(D|M_i) = P(M_i) \int_{\theta_i} P(D|\theta_i, M_i)P(\theta_i|M_i)d\theta_i \quad (3.25)$$

The problem of the Bayesian approach is that this integral is not computable, and some approximations should be done. The BIC criterion is the result of one such approximation, more specifically it derives from the ‘‘Laplace method for integrals’’. Let us consider only one model M_i , so that the equation (3.25) could be rewritten as:

$$P(D) = \int_{\theta} P(D|\theta)P(\theta)d\theta \quad (3.26)$$

Let us suppose, for simplicity, that the data set D is composed by n i.i.d. observations $\{y_1 \dots y_n\}$. Define the function $g(\theta)$ as

$$g(\theta) = \log(P(D|\theta)P(\theta)) \quad (3.27)$$

Let $\tilde{\theta}$ be the value of θ that maximizes the function $g(\theta)$, *i.e.* $\tilde{\theta}$ is the MAP estimate. Consider the Taylor series expansion of $g(\theta)$ in θ :

$$g(\theta) = g(\tilde{\theta}) + (\theta - \tilde{\theta})^T g'(\tilde{\theta}) + \frac{1}{2}(\theta - \tilde{\theta})^T g''(\tilde{\theta})(\theta - \tilde{\theta}) + o(\|\theta - \tilde{\theta}\|^2)$$

where the superscript T denote the matrix transpose, and $g'(\theta)$ and $g''(\theta)$ denotes the first derivatives vector and the Hessian matrix, respectively. These quantities are defined as

$$g'(\theta) = \left(\frac{\partial g(\theta)}{\partial \theta_1}, \frac{\partial g(\theta)}{\partial \theta_2}, \dots \right)$$

and

$$g''(\theta) = \begin{bmatrix} \frac{\partial^2 g(\theta)}{\partial \theta_1 \partial \theta_1} & \frac{\partial^2 g(\theta)}{\partial \theta_1 \partial \theta_2} & \dots \\ \frac{\partial^2 g(\theta)}{\partial \theta_2 \partial \theta_1} & & \\ \vdots & & \end{bmatrix}$$

Now $\tilde{\theta}$ is the maximum of $g(\theta)$, and hence $g'(\tilde{\theta}) = 0$; so we could obtain

$$g(\theta) \approx g(\tilde{\theta}) + \frac{1}{2}(\theta - \tilde{\theta})^T g''(\tilde{\theta})(\theta - \tilde{\theta}) \quad (3.28)$$

Please note that this approximation is good only if θ is close to $\tilde{\theta}$. However there are two considerations that justify this approximation: the first is that, if the number of observations n is large, the maximum of the likelihood $P(D|\theta)$ converges to the maximum of the posterior $\tilde{\theta}$ [30]; the second fact is that, in the case of a large n , the likelihood is sharply peaked around its maximum: only values of θ that are close to $\tilde{\theta}$ will contribute much to the integral (3.26) defining $P(D)$ [211]. Using these two considerations we could consider the quantity in (3.28) as a good approximation of (3.26), if n is large. So,

$$\begin{aligned} P(D) &= \int_{\theta} e^{g(\theta)} d\theta \\ &\approx e^{g(\tilde{\theta})} \int e^{\frac{1}{2}(\theta - \tilde{\theta})^T g''(\tilde{\theta})(\theta - \tilde{\theta})} d\theta \end{aligned} \quad (3.29)$$

Recognizing the integrand in the equation (3.29) as proportional to a multivariate normal density, we could use the ‘‘Laplace method for integrals’’, that approximates $P(D)$ as

$$P(D) \approx e^{g(\tilde{\theta})} 2\pi^{\frac{d}{2}} |A|^{-\frac{1}{2}} \quad (3.30)$$

where d is the number of parameters in the model and $A = -g''(\tilde{\theta})$. The error introduced in the approximation is $O(n^{-1})$ [211], where $O(n^{-1})$ represents any quantity such that $nO(n^{-1}) \rightarrow$ a constant, as $n \rightarrow \infty$. Passing to the logarithm representation,

$$\log P(D) = \log P(D|\tilde{\theta}) + \log P(\tilde{\theta}) + \frac{d}{2} \log 2\pi - \frac{1}{2} \log |A| + O(n^{-1}) \quad (3.31)$$

Now, for large number of samples n , the MAP estimate correspond to the Maximum Likelihood estimate $\hat{\theta}$. Moreover, for i.i.d. observations, $A \approx n\mathbf{I}^1(\theta)$, where $\mathbf{I}^1(\theta)$ is the *Expected Fisher Information Matrix* for one observation. This quantity was introduced by Fisher in 1922 [77], and represents a $d \times d$ matrix, defined as

$$\begin{aligned} \mathbf{I}^1(\theta) &= E_{y_1} \left[\left(\frac{\partial P(X|\theta)}{\partial \theta} \right)^2 \right] \\ &= -E_{y_1} \left[\frac{\partial^2 P(X|\theta)}{\partial \theta^2} \right] \end{aligned}$$

where the expectation is taken over values of y_1 , with θ held fixed. This quantity is really important, as it represents a sort of measure of the ‘‘averaged concavity’’ of the likelihood, *i.e.* measuring how ‘‘prominent’’ the maximum is. A particularly important relation involving the variance of an unbiased² estimator $\hat{\theta}(y)$ and the Fisher Information matrix is the *Cramer-Rao bound* [213], which states that

² Let θ be a parameter to be estimated from observations y which are generated according to some likelihood function $P(y|\theta)$, and let $\hat{\theta}(y)$ be some estimator of θ . Then this

$$E_Y \left[\left(\hat{\theta}(y) - \theta \right)^2 \right] \geq \frac{1}{\mathbf{I}(\theta)}$$

Roughly speaking, this theorem says that the variance of an unbiased estimator is bounded by the inverse of the Fisher Information matrix: it is intuitively acceptable that a parameter can be more accurately estimated if the associated log-likelihood has a clear maximum.

Coming back to the BIC derivation, the determinant of A is then approximated by $|A| \approx n^d |\mathbf{I}^1(\theta)|$. This approximation, together with the $\hat{\theta} \approx \hat{\theta}$, introduce an $O(n^{\frac{1}{2}})$ error into equation (3.31), which becomes

$$\log P(D) = \log P(D|\hat{\theta}) + \log P(\hat{\theta}) + \frac{d}{2} \log 2\pi - \frac{d}{2} \log n - \frac{1}{2} \log |\mathbf{I}^1(\theta)| + O(n^{\frac{1}{2}}) \quad (3.32)$$

Removing terms of order $O(1)$ or less, we obtain

$$\log P(D) = \log P(D|\hat{\theta}) - \frac{d}{2} \log n + O(1) \quad (3.33)$$

This equation says that the log-integrated likelihood is equal to the maximized likelihood, minus a correction term depending on the size of the model.

Practically, given a set of candidate models $M_1 \dots M_R$, for each M_i the BIC criterion is computed, following the formula

$$\text{BIC}(M_i) = \log p(\mathcal{O}|\widehat{M}_i) - \frac{|M_i|}{2} \log(n) \quad (3.34)$$

where \widehat{M}_i denotes the ML estimate of the model M_i , and $|M_i|$ is the total number of free parameters of \widehat{M}_i . The chosen model $M_{i_{\text{BIC}}}$ is then the one showing the maximum BIC value, *i.e.*

$$i_{\text{BIC}} = \arg \max_i \text{BIC}(M_i) \quad (3.35)$$

BIC for Hidden Markov Models

In this section the BIC formulation for the HMM case is presented, in order to exemplify the theoretical formulation of the previous section. Let \mathcal{O} denote the observed data-set, and let n be the total number of observations in \mathcal{O} , *i.e.*, $n = \sum_{l=1}^L T_l$. In the HMM case, fixed the minimum and the maximum number of states k_{\min} and k_{\max} , the BIC criterion is applied as follows:

For $k = k_{\min}$ to k_{\max} , do:

1. initialize the HMM using whatever approach (at least, randomly);
2. train the initialized HMM with k states, obtaining the HMM λ_k ;

estimator is said to be *unbiased* if verifies

$$E_y \left[\hat{\theta}(y) \right] = \int_{-\infty}^{+\infty} \hat{\theta}(y) P(y|\theta) dy = \theta$$

3. compute the BIC value $\text{BIC}(\boldsymbol{\lambda}_k)$, using the formula (3.34). In this case, the number of free parameters is the sum of the following factors:
 - initial state probability: $k - 1$ parameters;
 - transition probability: $k(k - 1)$ parameters;
 - emission probability: $kd + k\frac{d(d+1)}{2}$ parameters, where d is the dimensionality of the observation.

Choose the HMM $\boldsymbol{\lambda}_{\tilde{k}_{\text{BIC}}}$ that maximizes the BIC value $\text{BIC}(\boldsymbol{\lambda}_k)$, *i.e.*

$$\tilde{k}_{\text{BIC}} = \arg \max_k \text{BIC}(\boldsymbol{\lambda}_k) \quad (3.36)$$

3.4.4 The proposed approach

Given a set of sequences $\{\mathbf{O}^i\} = \{O_1^i \dots O_{T_i}^i\}$, the goal is to estimate the number of states k of the HMM that better models the sequences. Fixed the minimum and the maximum number of states, *i.e.* k_{\min} and k_{\max} , the proposed strategy works as follows:

1. Consider the set $D = \{O_1^1, O_2^1, \dots, O_{T_1}^1, O_1^2, O_2^2, \dots\}$, that is the set of values of the *unrolled* sequences;
2. for $k = k_{\min}$ to k_{\max} , do:
 - fit the set D with a k Gaussians mixture model, using the Expectation Maximization (EM - [57, 226]) algorithm; denote by \mathcal{G}_k the Gaussian mixture obtained.
 - compute the $\text{BIC}(\mathcal{G}_k)$ value (3.34).
3. Choose the clustering \mathcal{G}_k that maximizes the BIC criterion (3.34), *i.e.*,

$$\hat{k}_{\text{BOI}} = \arg \max_k \text{BIC}(\mathcal{G}_k)$$

4. Initialize (with $\mathcal{G}_{\hat{k}_{\text{BOI}}}$) and train an HMM with \hat{k}_{BOI} states.

In the following, we call this approach the BOI (Bic On Initialization) approach. It is clear that with the BOI approach only one HMM training is needed, with a notable reduction in the computational load with respect to the standard BIC approach, where $k_{\max} - k_{\min} + 1$ training are needed.

It is evident that this approach could be used not only with the BIC criterion, but with any standard model selection criterions, as Minimum Description Length (MDL - [185]), Minimum Message Length (MML - [165]), Akaike's Information Criterion (AIC - [5]), or others (for a complete list of references on this topics see [74] and [189]).

3.4.5 Experimental results

In this section the proposed approach is compared with standard BIC method, presented in Section 2, where the initialization (step 1) is performed using the GMM clustering approach described in Section 3.4.1. It is worth noting that a comparison in term of accuracy of model order estimation in synthetic experiments is not fair, as the proposed approach is not able to take into account the transition

matrix of the HMM, resulting in a coarser model size estimation. It is indeed interesting to compare the classification accuracy of BOI and BIC approaches in real cases, as, for instance, the face recognition problem.

The face recognition task is addressed as explained in Chapter 7.2, where DCT coefficients are used as features: all details are given in the mentioned chapter, and here only briefly sketched. From a face image, the sequence of sub-images is gathered by a raster scanning, and for each sub-image DCT coefficients are computed. In our experiment, the sub-images are of dimension 16x16, with 50% of overlapping; three experimental sessions are performed, using 10, 8 and 3 DCT coefficients for each sub image, respectively. The database used is the ORL database, composed by 40 subjects, and for each subject, 10 poses are given: 5 are used for training, the remaining for testing. Experiments were repeated 20 times, in order to increase the statistical significance of the results. k_{min} and k_{max} were fixed to 2 and 7, respectively. Classification accuracies for the three experiments are reported in Table 3.1, together with relative standard deviations.

Table 3.1. Averaged classification accuracies in face experiment using BIC and BOI techniques, with (a) 10 DCT coefficients, (b) 8 DCT coefficients and (c) 3 DCT coefficients.

Method	Averaged accuracy	Standard Deviation
BIC	97.50%	1.52%
BOI	98.17%	1.35%

(a)

Method	Averaged accuracy	Standard Deviation
BIC	97.08%	1.66%
BOI	97.93%	1.73%

(b)

Method	Averaged accuracy	Standard Deviation
BIC	96.72%	1.64%
BOI	92.88%	2.71%

(c)

We could note that the two approaches perform equally well in average, even if there is a discrepancy between the results of the two experiments: obviously the performance obtained depends on the specific task. Moreover, the supremacy of the BIC approach in the third example is more evident, as expected. These tables confirm that the proposed approach, even if “coarse” in some way, is really effective in discovering the true structure of the problem, obviously relatively to this case.

In order to get a better insight into the proposed technique, we compare, for each one of the 40 subjects, the number of states estimated by the BOI approach with the number of states estimated and the BIC approach. These quantities,

averaged over all 20 experiments, are plotted in Fig. 3.3, where the 40 subjects are subdivided in 4 plots for augmenting the readability.

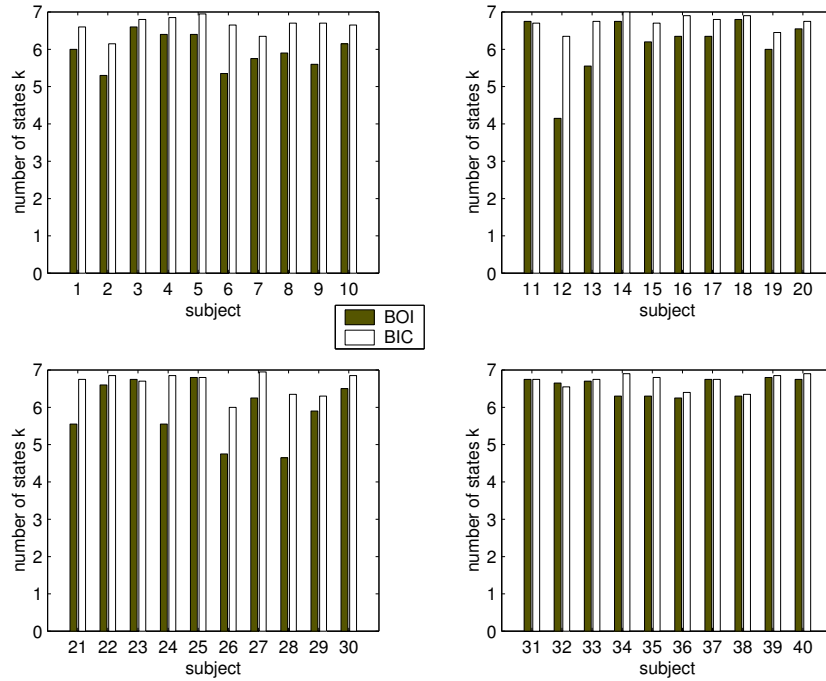


Fig. 3.3. Estimated number of states using BOI (left bar) and BIC (right bar) in the face experiments, using 10 DCT coefficients.

From this figure it is evident that the number of states estimated by the BOI approach is everywhere lower than the BIC estimate. One explanation could be the following: the BOI approach is inherently simpler than the BIC one, since it considers only the unrolled sequence, and bases its decision only on the examination of the Gaussian mixture fitting. The differentiation between states is therefore decided merely on the basis of the static components of the sequence, while the dynamics cannot be taken into consideration. This could lead to an erroneous merging of those states that present similar static behaviors but different dynamic structures. The BIC approach, instead, is based on the HMM training, and is therefore able to take into account both the static and the dynamic behavior of the sequences: this is the reason for the identification of more states.

3.4.6 Conclusions

The proposed approach addresses the model selection issue in the initialization phase, resulting in a quite simple but fast technique. The number of the states of the HMM is estimated by finding the Gaussian mixture that better fits the

unrolled sequence, with respect to some predefined model selection criterion. In this way, only one HMM training is needed, with a great reduction of the computational load needed by a standard model selection approaches. Experimental results on face recognition task show that the the classification accuracies of the proposed approach are comparable with those of the standard BIC technique, even if requiring much lower computational resources.

3.5 The Bisimulation approach

In this section a syntactic approach for addressing the model selection problem is presented, in particular with the aim of determining the number of states in discrete emission HMMs.

3.5.1 Motivations

Most of the standard approaches to HMM model selection are devoted to find the optimal model on the basis of a criterion function by exploring all (or a large part of) the search space, resulting typically in a computationally expensive procedure. The technique here proposed is instead a direct method to identify the model without searching the whole space, resulting less computationally intensive.

The proposed approach consists in eliminating syntactic redundancy of a Hidden Markov Model using a technique called bisimulation. Bisimulation represents a notion of equivalence between graphs (or between nodes in a graph) whose usefulness has been demonstrated in various fields of Computer Science. It is used for the testing process equivalence in Concurrency [155], as a notion of equivalence between Kripke Structures in Model-Checking [21], to provide operational semantics to query languages in Web-like databases [148], and to replace extensionality in the context of non well-founded sets in Set Theory [2].

With this approach, the structure of a HMM is reduced by computing bisimulation equivalence relation between states of the model, so that equivalent states can be collapsed. We employed both the notions of probabilistic and standard bisimulation. It is shown, by experiments on DNA sequence modelling and 2D shape recognition, that bisimulation reduces the number of states without significant loss in term of likelihood and classification accuracy.

3.5.2 Bisimulation

Bisimulation is a notion of equivalence between graphs useful in several fields of Computer Science. The notion was introduced by Park for testing process equivalence, extending a previous notion of automata simulation by Milner. Milner then employed bisimulation as the core for establishing observational equivalence of the Calculus of Communicating Systems [155].

In [121], Kanellakis and Smolka relate the bisimulation problem with the general (relational) coarsest partition problem and pointed out that the partition refinement algorithm in [166] solves this task. More precisely, in [166] Paige and Tarjan solve the problem in which the stability requirement is relative to a relation E (edges) on a set N (nodes) with an algorithm whose complexity is $O(|E| \log |N|)$.

Standard Bisimulation.

Bisimulation can be equivalently formulated as a relation between two graphs and as a relation between nodes of a single graph. Since the interest is in reducing states of a unique graph, the latter definition is adopted.

Definition 3.1. *Given a graph $G = \langle N, E \rangle$ a bisimulation on G is a relation $b \subseteq N \times N$ s.t. for all $u_0, u_1 \in N$ s.t. $u_0 b u_1$ and for $i = 0, 1$: if $\langle u_i, v_i \rangle \in E$, then there exists $\langle u_{1-i}, v_{1-i} \rangle \in E$ s.t. $v_0 b v_1$.*

In order to minimize the number of nodes of a graph, we look for the maximal bisimulation \equiv on G . Such a maximal bisimulation always exists, it is unique, and it is an equivalence relation over the set of nodes of G [2]. The minimal representation of $G = \langle N, E \rangle$ is therefore the graph:

$$\langle N/\equiv, \{ \langle [m]_{\equiv}, [n]_{\equiv} \rangle : \langle m, n \rangle \in E \} \rangle$$

which is usually called the *bisimulation contraction* of G . In the above formula, N/\equiv represents the quotient set of N w.r.t. the equivalence \equiv , $[m]_{\equiv}$ and $[n]_{\equiv}$ are the equivalence class of m and n , respectively.

Using the algorithm in [166] the problem can be solved in time $O(|E| \log |N|)$; for acyclic graphs and for some classes of cyclic graphs it can be solved in linear time w.r.t. $|N| + |E|$ [63, 62].

Bisimulation on labelled graphs.

If the graphs are such that nodes and/or edges are labelled, the notion can be reformulated as follows:

Definition 3.2. *Let $G = \langle N, E, \ell \rangle$ be a graph with a labelling function ℓ for nodes, and labelled edges of the form $m \xrightarrow{a} n$ (a belongs to a set of labels). A bisimulation on G is a relation $b \subseteq N \times N$ s.t. for all $u_0, u_1 \in N$ s.t. $u_0 b u_1$ it holds that: $\ell(u_0) = \ell(u_1)$ and for $i = 0, 1$, if $u_i \xrightarrow{a} v_i \in E$, then there exists $u_{1-i} \xrightarrow{a} v_{1-i} \in E$ s.t. $v_0 b v_1$.*

If exclusively the nodes are labelled, the procedure in [166] can be employed to find the bisimulation contraction, provided that in the initialization phase nodes with the same labels are put in the same class. The case in which edges are labelled can be reduced to the last one by replacing a labelled edge $m \xrightarrow{a} n$ by a new node ν labelled by a and by the edges $\langle m, \nu \rangle$ and $\langle \nu, n \rangle$. Finding the bisimulation contraction can therefore be done also in this case by using the algorithm of [166]; moreover, the procedure of [166] can be modified in order to deal directly (*i.e.*, without preprocessing) with the general case described.

Probabilistic Bisimulation

The notion of bisimulation over labelled graphs (Def. 3.2) has been introduced in a context where labels denote actions executed (e.g. a symbol is emitted) by processes during their run. Labels can also store pairs of values $\langle x, y \rangle$: an action x and a probability value y (that could be read as: this edge can be crossed with probability y and in this case an action x is done). These graphs are called *Fully Probabilistic Labelled Transition System* (FPLTS).

In this case another notion of bisimulation is probably more suitable. Consider, for instance, the graph of Fig. 3.4 (we use n_1-n_8 to refer to the nodes: they are

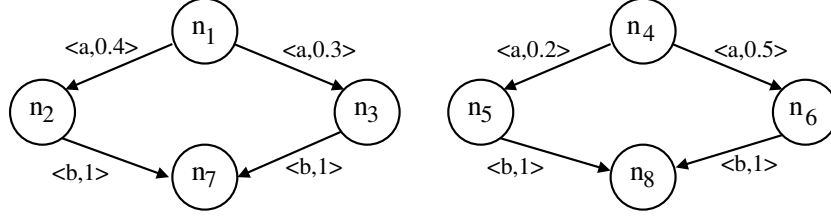


Fig. 3.4. n_1 and n_4 are not bisimilar, but probabilistically bisimilar

not labels). n_7 and n_8 are trivially equivalent since they have no outgoing edges. Nothing can be done in both cases. The four nodes n_2, n_3, n_5, n_6 are in the same equivalence class, since they have equivalent successors (reachable performing the same action b , with probability 1). The nodes n_1 and n_4 are instead not equivalent, since, for instance, there is the edge $n_1 \xrightarrow{\langle a, 0.4 \rangle} n_3$ but no edges labelled $\langle a, 0.4 \rangle$ starts from n_4 . However, one of the equivalent states can be reached both from n_1 and n_4 , performing action a with probability 0.7: the two nodes should be considered equivalent.

The notion of *probabilistic bisimulation* [10] is aimed at formally justifying this intuitive concept. We start by providing two auxiliary notions: given a graph $G = \langle N, E \rangle$ with edge labelled by pairs as above, and $b \subseteq N \times N$ a relation, then for two nodes $m, n \in N$ and a symbol a , we define the functions B and S as follows

$$B(m, n, a) = \{\mu : \exists q (m \xrightarrow{\langle a, q \rangle} \mu \in E \wedge \mu b n)\}$$

$$S(m, n, a) = \sum_{m \xrightarrow{\langle a, q \rangle} \mu \in E, \mu b n} q$$

Definition 3.3. Let $G = \langle N, E \rangle$ be a graph with edge labelled by pairs consisting of symbols and probability values, a probabilistic bisimulation on G is a relation $b \subseteq N \times N$ s.t.: for all $u_0, u_1 \in N$, if $u_0 b u_1$ then for $i = 0, 1$ if $u_i \xrightarrow{\langle a, p \rangle} v_i \in E$, then there exists $v_{1-i} \in N$ s.t.:

- $u_{1-i} \xrightarrow{\langle a, p' \rangle} v_{1-i} \in E$,
- $S(u_i, v_i, a) = S(u_{1-i}, v_{1-i}, a)$, and
- and for all $m \in B(u_i, v_i, a)$ and $n \in B(u_{1-i}, v_{1-i}, a)$ it holds that $m b n$.

In [10] a modification of the Paige-Tarjan procedure is presented for this probabilistic case and proved to correctly return the probabilistic contraction of a graph $G = \langle N, E \rangle$ in time $O(|N||E| \log |N|)$. In the example of Fig. 3.4 the two nodes n_1 and n_4 are put in the same class.

In the proposed approach, the possible labels for edges are further extended. Triplets $\langle p_1, a, p_2 \rangle$ are admitted, where a is a symbol while p_1 and p_2 are probabilistic values. The notion of the above Definition 3.3 is extended point to point. In other words, the reasoning is as if the edge $\langle p_1, a, p_2 \rangle$ is replaced by the two edges $\langle a, p_1 \rangle, \langle \hat{a}, p_2 \rangle$ and \hat{a} can not be confused with a (see Fig. 3.5).

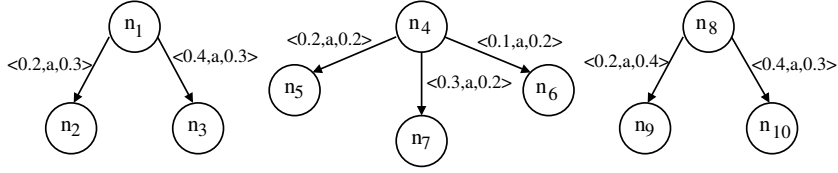


Fig. 3.5. n_1 and n_4 are probabilistically bisimilar. n_1 and n_8 are not.

3.5.3 The Strategy

HMMs as labelled graphs

Probabilistic bisimulation is defined on FPLTS, which are slightly different from HMMs. Neglecting notation, the real problem is represented by the emission probability of each state, which has not counterpart in FPLTS. As described in Sect. 3.5.2, the problem could be solved by choosing an appropriate initial partition, whose sets contain states with the same emission probability and then run the algorithm of [166]. This approach is correct, but it is too restrictive with respect to the concept of probabilistic bisimulation. In other words, using this initialization the classes of bisimulation equivalence are created by using the concept of syntactic labelling, loosing instead the semantic labelling concept, which is the kernel of the probabilistic bisimulation.

Thus, another method is proposed, a bit more expensive in terms of memory allocation and computational cost, but offering a better semantic characterization.

Definition 3.4. *Given a HMM $\lambda = (S, V, \mathbf{A}, \mathbf{B}, \pi)$, trained with a set of strings from the alphabet $V = \{v_1, v_2, \dots, v_m\}$, the equivalent FPLTS is obtained as follows. For each state S_i :*

- Let A_i be the set of edges outgoing from the state S_i , defined as

$$A_i = \{\langle S_i, S_j \rangle : a_{ij} \neq 0, 1 \leq j \leq k\}$$

- each edge e in A_i is replaced by m edges, whose labels are $\langle a_{ij}, v_p, b(p|S_i) \rangle$, where, for $1 \leq i, j \leq k, 1 \leq p \leq m$:
 - a_{ij} is probability of e ;
 - v_p is p -th symbol of V ;
 - $b(p|S_i)$ is probability of emission of v_p from state S_i .

Given an HMM with k states, E edges and m symbols, with this approach the complexity of bisimulation contraction grows from $O(Eklogk)$ to $O(mEklogk)$ for time, and from $O(Ek)$ to $O(mEk)$ for space.

By applying bisimulation to a HMM another important issue has to be considered: the partial control of compression rate of the strategy. To this aim, the concept of *quantization* of probability is introduced: given a set of quantization levels (prototypes) in the interval $[0, 1]$, each probability is approximated with the closest prototype. A uniform quantization is adopted on interval $[0, 1]$. To control

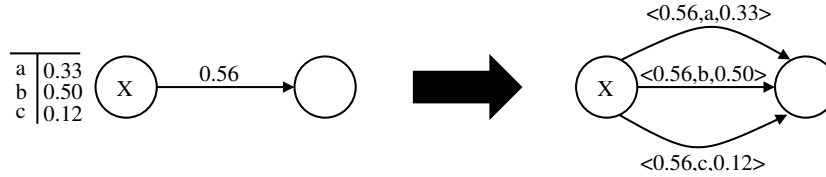


Fig. 3.6. Basic idea of procedure to represent HMM as a FPLTS.

this approximation a *reduction factor* is defined, representing the number of levels that subdivide the interval: it is calculated as $(\text{number of prototypes} - 2)$. For example, reduction factor 3 means that probability are approximated with the values $\{0, 0.25, 0.5, 0.75, 1\}$. Thus, the notion of equivalent labels is governed by the test of equality of their quantization, where $\text{quant}(p)$ is defined as the prototype j closest to p .

As a final consideration, the reduction factor represents a tuning parameter for deciding the degree of the compression adopted. Obviously, for a low value of the factor, the information lost in approximation is high, and the resulting model can be a very poor representation of the original one.

Algorithm

Given a problem, determining the optimal number of HMM states is performed through the following steps:

1. Train the HMM with a number of states that is reasonably large with respect to the problem considered. This number strongly depends from available data, and it can be determined using some heuristics.
2. Transform the HMM in labelled graph (FPLTS), using procedure described in Def. 3.4 of Sect. 3.5.3. In this step the reduction factor has to be chosen, providing a measure of the accuracy adopted in the conversion. It also gives a rough meaning of reduction rate: lower precision likely means higher compression.
3. Run the bisimulation algorithm on such graph, obtaining equivalence classes. Optimal number of states k' is represented by cardinality of the quotient set (*i.e.* the number of different classes determined by bisimulation).
4. Retrain the HMM using k' states.

This method is designed for discrete HMM, but can be generalized for other topologies by working on Step 2 of the procedure.

3.5.4 Experimental results

The aim of the following experiments is to show that this method reduces HMM states without significant loss in terms of likelihood and classification accuracy. We tested these two properties on two distinct problems: DNA modelling, *i.e.* using HMM to model and recognize different DNA sequences (typically, fragments of genes), and 2D shape classification, using the method proposed in [25]. In all tests, the initialization problem was addressed by the following procedure: starting from

random initial estimates of \mathbf{A} , \mathbf{B} and $\boldsymbol{\pi}$, each HMM was trained in three learning sessions, using Baum-Welch re-estimation; the chosen model is the one presenting the maximum likelihood. Each learning ended when likelihood is converged or after 100 training cycles. Performances are measured in terms of the following indices:

- *Compression Rate*, representing a percentage measure of the number of states eliminated by bisimulation:

$$CR = 100 \left(\frac{k_{orig} - k_{reduct}}{k_{orig}} \right)$$

where k_{reduct} are the number of states after bisimulation on a HMM with k_{orig} states; this index measures the degree of the compression obtained by apply the bisimulation;

- *Log Likelihood Loss*, estimating the difference in LL between original and reduced HMM:

$$LLL = 100 \left(\frac{LL_{orig} - LL_{reduct}}{LL_{orig}} \right)$$

where LL_{reduct} and LL_{orig} are log likelihood of HMM with k_{reduct} and k_{orig} number of states, respectively. This index indicates how much likelihood is lost in the reduction: low values indicate that the reduction is able to eliminate redundancy of the model, without affecting the intrinsic characteristics.

DNA modelling

Genomic offers tremendous challenges and opportunities for computational scientists. DNA are sequences of various lengths formed by using 4 symbols: *A*, *T*, *C*, and *G*. Each symbol represent a base, *Adenine*, *Thymine*, *Cytosine*, and *Guanine* respectively. Recent advances in biotechnology have produced enormous volumes of DNA related information, needing suitable computational techniques to manage them [192, 68].

From a machine learning point of view [193], there are three main problems to deal with: *genome annotation*, including identification of genes and classification into functional categories, *computational comparative genomic*, for comparing complete genomic sequences at different levels of detail, and *genomic patterns*, including identification of regular pattern in sequence data. Hidden Markov Models are widely used to resolve these problems, in particular for classification of genes, protein family modelling, and sequence alignment. This is because they are very suitable for modelling strings (as DNA or protein sequences), and can provide useful measures of similarity (LL) in comparing genes.

In this paper, HMMs are employed in order to model gene sequences for classification purposes. This simple example is nevertheless significant to demonstrate HMM ability in recognizing genes, also in conditions of noise (as biological mutations). Data were obtained extracting a 200 bp (base pair) fragment of a *recA* gene sequence of a *Lactobacillus* species. We trained 95 HMMs on this sequence, where k (number of states) grows from 10 to 200 (step 2). We applied the bisimulation contraction algorithm on each HMM, with reduction factor varying from 1 to 9 (step 2), computing the number of resulting states. We then compared Log

Likelihood (LL) of original sequence produced by original and reduced HMMs, obtaining results plotted on Fig. 3.7(a).

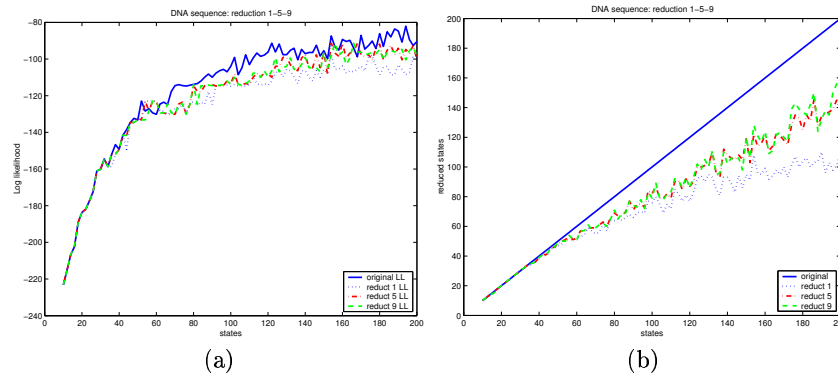


Fig. 3.7. DNA modelling experiment: (a) comparison of Likelihood curve for original and reduced HMM, and (b) compression rate, for reduction factor equal to 1, 5 and 9.

One can notice that the two curves are very similar, in particular when reduction factor is high. In Table 3.2, averaged and maximum loss of likelihood (LLL) are presented for each value of reduction factor, with maximum compression rate: loss of Log Likelihood is fairly low, decreasing when augmenting precision of bisimulation (reduction factor). This kind of analysis is performed to show the satisfying evolution of the HMM likelihood when number of states is decreased using bisimulation.

Table 3.2. Maximum compression rate, average and maximum Log Likelihood loss for DNA modelling experiment at varying reduction factors.

Reduction factor	Max CR (%)	Average LLL (%)	Max LLL (%)
1	50.00	9.57	32.20
3	38.16	6.69	20.07
5	33.14	5.45	20.31
7	34.87	5.91	18.90
9	34.04	5.77	22.30

In Fig. 3.7(b) original number of states vs. reduced number of states are plotted, at varying number of states. More precisely, for a generic value k on the abscissa axis, ordinate axis represents the number of states obtained after running bisimulation on k -states HMM. It is worth noting that compression rate increases when the number of states grows: this is reasonable, because small structures cannot have a large redundancy.

The second part of this experiment tries to exploit the performance of our algorithm regarding the classification accuracy. To perform this step we trained

two HMMs with 150 states on 200 bases fragments of two different *recA* genes: one was from *Corynebacterium glutamicum* and second was from *Mycobacterium tuberculosis*. Each HMM was then reduced using bisimulation, varying reduction factor from 1 to 9 (step 2). Then, HMMs were retrained with reduced number of states, resulting in 10 reduced HMMs (5 for each sequence). Compression rate varies from 32% for reduction factor 1 to 22% for reduction factor 9 (see Table 3.3). We tested classification accuracy of HMMs using 300 sequences, obtained by adding synthetic noise to the original two. The noising procedure is the following: each base is changed with fixed probability p (ranging from 0.3 to 0.4), and following some determined biological rules (for examples, *A* becomes *T* with probability higher than *G*). Each sequence of this set was evaluated using both models, and classified as belonging to the class whose model showed the highest LL. Error rate was then calculated counting misclassified trials and dividing by the total number of trials. Fig. 3.8 shows error rate for original and reduced HMMs, varying the probability of noise. One can notice that error rate trend is quite similar, and that error is very low, always below 5%, proving that HMMs work very well on this type of problems. In Table 3.3 (a–b), average errors on original and reduced HMMs

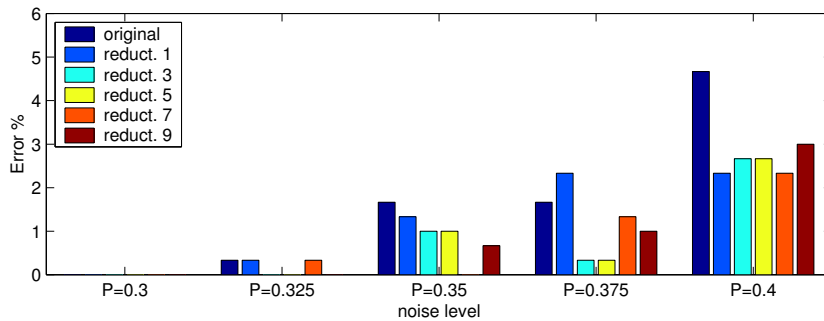


Fig. 3.8. Error rate for different noise level for DNA modelling experiment.

are presented, varying noise level and reduction factor value, respectively. For the latter, maximum compression rate and maximum LL loss are also presented. One can notice that the difference between two errors grows with noise level, *i.e.*, the error value becomes higher when the noise level increases, and differences can be more significant. Nevertheless, LL losses are very low if compared with compression rate and amount of noise. Actually, classification errors remain below 5%, even on experiments with 40% noise level. Moreover, error level seems to be lower in the reduced case than in the original one. Reasonably, HMMs with less states are able to generalize better, so that they recognize also sequences with higher noise, even if we expect a breakdown point, causing a reversing behavior between original and reduced HMMs.

Table 3.3. Error on original and reduced HMMs for DNA modelling experiments in function of (a) varying noise level, and (b) varying reduction factor value.

Noised Level	Error on Original HMM (%)	Error on Reduced HMM (%)
0.300	0.00	0.00
0.325	0.33	0.13
0.350	1.67	0.80
0.375	1.67	1.07
0.400	4.67	2.60

(a)

Reduction Factor	Average CR (%)	Average LLL (%)	Error on Original HMM (%)	Error on Reduced HMM (%)
1	32.00	3.89	1.67	1.27
3	25.33	1.72	1.67	0.80
5	22.00	4.15	1.67	0.80
7	21.33	5.61	1.67	0.80
9	21.66	2.14	1.67	0.93

(b)

2D shape recognition

The 2D shape classification experiment used here is presented in [25], sharing some of the ideas presented in Chapter 6. Briefly, the idea is to characterize an object by its contour, modelled using the chain code. This sequence is then used to train a discrete HMM. Even if the chain code coding is not very accurate (a less rough approach is discussed in Chapter 6), the HMM is well suitable for 2D shape classification: in [25] it was shown that HMM is able to correctly discriminate between objects, even if noisy, scaled or occluded.

In the proposed experiment, although limited to a couple of similar objects, the degree of occlusion is quite large, and noise has been included to affect object coding, (without heavily degrading classification performances).

In our experiment, given an image of 2D objects, data are gathered assigning at each object its chain code, calculated on object contours. Edges are extracted using *Canny edge detector* [40], while chain code is calculated as described in [111]. Fig. 3.9 shows the two simple objects, a stylized hammer and a screwdriver, used in the experiment. One HMM for each object is trained, varying the number of states from 4 to 20. After applying bisimulation contraction, with reduction factor from 1 to 9, HMMs are retrained with reduced number of states and compared in term of Log Likelihood. Average and maximum Log Likelihood loss are calculated, and results are shown in Table 3.4, with maximum compression rate for different reduction factor values. Average LLL values are comfortably low: bisimulation does not seem to affect HMM characteristics, but is able to remove syntactic redundancy from the model. Nevertheless, we can also observe that average loss is very low

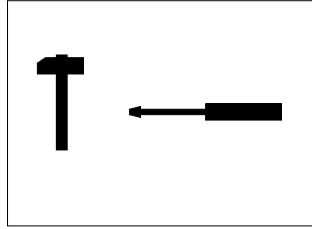


Fig. 3.9. Toy images for 2D shape recognition using Chain Code.

Table 3.4. Maximum compression rate, average and maximum Log Likelihood loss for 2D shape recognition test, for different reduction factors.

Reduction factor	Max CR	Average LLL	Max LLL
1	16.74	4.91	72.20
3	9.43	0.55	29.39
5	6.33	2.30	72.26
7	6.40	1.72	72.85
9	4.71	1.28	68.73

if compared with related maximum LLL. This is because compression is not so strong, as evident in Table 3.4, and therefore some learning session on reduced HMM can produce better results in terms of Log Likelihood. LL of an HMM on a sequence typically grows with k . On the other hand, LL depends on how well the training algorithm worked on the data. Baum-Welch re-estimation ensures to reach the nearest local optimum, without any information about global optimum. So, for closed k_1, k_2 , with $k_1 < k_2$, a HMM with k_1 states can possibly show larger LL than one with k_2 states, because the training algorithm worked better. To partially solve the problem of convergence, each HMM was trained three times, starting with different random initial conditions. The case of so high LL loss may be explained by a low compression rate (the HMMs have the similar number of states) and a very bad training (in this case three trials seems to be insufficient to ensure correct learning).

For testing classification accuracy, two synthetic test sets were created. The first set is obtained considering, for each object, fragments of their chain code of variable length, expressed as percentage rate of the whole length: this simulates the occlusion of the object. The occlusion percentage varies from 10 to 80 percent (20% and 90% of the object is still visible), and the point where fragment starts was randomly chosen. This aspect is important in order to evaluate the independence of the method from which part of the object is actually occluded.

The second set is obtained by adding synthetic noise to the two chain codes, using a procedure similar to that used for DNA noising procedure. Each code is changed with fixed probability P , *i.e.* if cc_i is the original code, with probability P , $((cc_i - 1) \pm 1) \bmod 8 + 1$ is carried out. Probability ranges from 0.05 to 0.35, and, for each value, 60 sequences are generated. As usual, a sequence is assigned to the class whose model shows the highest Log Likelihood, and error

rate is estimated counting misclassified patterns. For each of the two test sets, we calculate performance using original and reduced HMMs and varying reduction factor from 1 to 9. In Table 3.5, averaged errors for original and reduced HMMs on a set of pieces are presented varying reduction factor from 1 to 9. We can see that the difference between the two errors is very low.

Table 3.5. Error on original and reduced HMMs for 2D shape recognition experiment (fragments set): (a) varying resolution factor; (b) varying fragment length.

Reduction factor	Error on Original HMM(%)	Error on Reduced HMM (%)
1	2.52	2.91
3	2.52	2.19
5	2.52	1.51
7	2.52	0.44
9	2.52	2.70

(a)

Fragment Length (%)	Error on Original HMM(%)	Error on Reduced HMM(%)
20 %	4.50	4.33
30 %	3.60	3.28
40 %	2.77	2.32
50 %	3.23	2.31
60 %	3.23	1.75
70 %	2.83	1.36
80 %	0.00	0.23
90 %	0.00	0.01

(b)

The same results are presented in Table 3.6 for a set of noisy sequences, varying reduction factor (Table 3.6(a)) and noise level (Table 3.6(b)).

A consideration can be made on the HMMs performance when applied to this problem: average error in recognizing the fragment sequence is 1.21%, a very low value. This means that a simple HMM can be invariant with respect to some type of object occlusions. Nevertheless, noise seems to be a more serious problem, but working on topology and training algorithms classification accuracy may be less affected by this problem.

Another point regards the similarity of the two objects which may seriously affect performances. This problems may be attenuated by using very different objects.

Comparison with other methods

In this section, the proposed approach is compared with BIC method, described in Section 3.4, with respect to the 2D shape experiment. 18 HMMs were trained,

Table 3.6. Error on original and reduced HMMs for 2D shape recognition experiment (noised set): (a) varying resolution factor; (b) varying noise level (b).

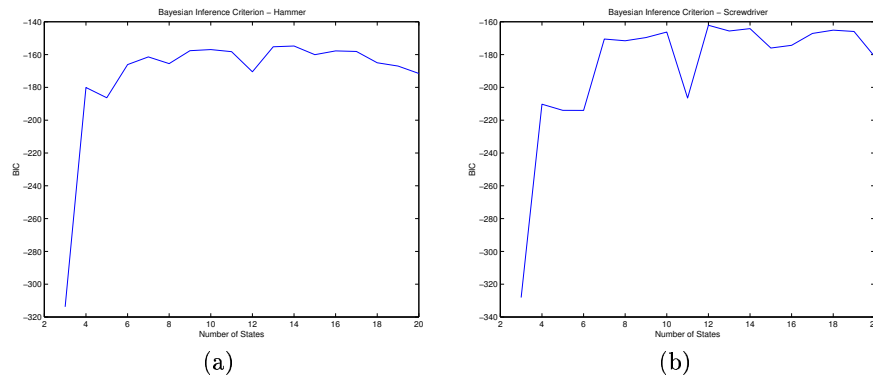
Reduction factor	Error on Original HMM(%)	Error on Reduced HMM(%)
1	29.08	24.83
3	29.08	29.05
5	29.08	21.14
7	29.08	28.23
9	29.08	25.97

(a)

Noise level (%)	Error on Original HMM (%)	Error on Reduced HMM(%)
5	11.33	9.64
10	20.5	17.24
15	27.11	23.61
20	31.67	28.21
25	35.24	31.70
30	37.78	34.16
35	39.95	36.33

(b)

with states number varying from 3 to 20, and for each model the BIC value was computed. BIC vs number of states curves are plotted in Fig. 3.10, for the two objects. The chosen HMMs are those showing the highest BIC value (corresponding

**Fig. 3.10.** BIC value vs number of states curves for the 2D shape recognition experiment with (a) the hammer and (b) the screwdriver.

to 12 and 14 states, for screwdriver and hammer, respectively).

The bisimulation approach trains one HMM with 20 states; then it applies bisimulation, and, finally, it trains another HMM with calculated number of states,

varying reduction factor for 1 to 9 (step 2). To compare the two methods a test set is created by adding synthetic noise (of various entity) to the two chain codes, in a way similar to that presented in the previous section, obtaining, for each noise level, 120 sequences to be classified. The classification errors obtained by applying the two approaches were calculated, and presented in Table 3.7, in function of variable noise level. We can notice that, on the average, classification accuracy

Table 3.7. Comparison between BIC method and our approach: “S.” stands for screw-driver and “H.” stands for hammer.

Method	States		Classification Error						
	S.	H.	Noise 0.05	Noise 0.10	Noise 0.15	Noise 0.20	Noise 0.25	Noise 0.30	Noise 0.35
BIC	12	14	13.33	25.00	32.50	36.88	39.83	41.81	42.98
Bisim RF 1	14	15	20.00	30.00	33.89	36.25	42.67	44.44	48.57
Bisim RF 3	15	16	21.67	34.17	39.44	42.08	43.67	44.72	45.48
Bisim RF 5	18	18	10.00	10.83	22.78	29.17	34.67	40.28	35.71
Bisim RF 7	17	19	28.33	38.33	42.22	44.17	45.33	46.11	46.67
Bisim RF 9	20	20	31.67	37.50	37.22	37.08	37.00	34.17	30.24

is quite similar: in fact BIC method needs 18 training session, while our method only two, plus the time necessary for determining bisimulation contraction (that is $O(mEk \log k)$, given an HMM with k states, E edges and m symbols). In problems with a short alphabet (as DNA modelling and chain code problems), our method is definitively faster than BIC, while it gives approximately the same classification accuracy.

3.5.5 Conclusions

The presented approach, which makes use of the probabilistic bisimulation, is used to estimate the minimal structure of a HMM. It has been shown that starting from a redundant configuration, bisimulation allows to merge equivalent states while preserving classification performances. Redundant and minimal HMM architectures have been tested on two different cases, DNA modelling and 2D shape classification, showing the usefulness of the approach. Moreover, the method has been compared with the BIC criterion, showing comparative performances but with a less computational complexity.

3.6 Pruning Model Selection

3.6.1 Motivations

In this section, a new model selection approach is proposed, aimed at improving standard model selection methods which can be used for HMMs, as the *minimum description length* (MDL) principle ([185]), the *Bayesian inference criterion* (BIC) ([196]), and the *minimum message length* (MML) criterion ([165]). These methods address the model selection problem by training several models, with different structures, and then by choosing the one that maximizes a certain selection criterion. These approaches perform rather accurately, allowing an increase in performance (e.g., [144], [183], and [230]). Although these techniques are less computationally expensive than Cross Validation [207], they still involve a considerable computational burden, since one full training is required for each candidate model structure. Moreover, all these approaches suffer for the already explained problem of the initialization of the parameter in the training, that crucially affects the obtained model estimate: this behavior strongly affects the model order selection criteria.

The approach proposed in this section simultaneously addresses the two issues mentioned above: the computational burden of model selection, and the initialization phase. The key idea is to use a decreasing learning strategy, starting each training session from an informative situation derived from the previous training phase. More specifically, the proposed procedure consists in starting the model training using a large number of states, run the estimation algorithm, and, after convergence, evaluate the chosen model selection criterion for that model. Then, the “least probable” state is pruned, and this configuration is taken as initial situation from which to start again the training procedure. In this way, each training session is started from a “nearly good” estimate. A related approach has been successfully used for Gaussian mixtures in [74]. The key observation supporting this approach is that, when the number of states is extremely large, the initialization dependency of the estimate is much weaker than when the number of states is close to the optimum. Moreover, the “good” initialization drastically reduces the number of iterations required by the learning algorithm, resulting in a less computational demanding procedure. The idea of pruning model selection was successfully employed also in the field of Neural Networks (see [30] and the references herein contained). The proposed method could be applied for all types of HMMs, discrete, continuous, autoregressive and so on. Moreover, it can be used with any model selection criterion: we consider the BIC, and the *mixture minimum description length* (MMDL), a criterion proposed in [74] for Gaussian mixtures and here extended to HMMs. It is worth noting that although Gaussian mixtures can be considered as (simple) special cases of HMMs, applying MMDL and the pruning strategy to HMMs involves additional conceptual and technical difficulties which need to be addressed.

The proposed approach and the normal strategy are largely compared in the experimental session, in terms of accuracy of model selection, classification performance, and computational requirements, using both real and synthetic data.

3.6.2 Model Selection criterions

In the following we will denote an HMM with k states as λ_k . The proposed strategy is applied with the BIC criterion, largely detailed in Section 3.4.3 and with the MMDL criterion, explained in the following.

Mixture Minimum Description Length for HMM

To explain the rationale behind MMDL, we start with the standard MDL criterion [185], which coincides with BIC (equation (3.34)):

$$\text{MDL}(k) = \log p(\mathcal{O}|\hat{\lambda}_k) - \frac{N_k}{2} \log(n) \quad (3.37)$$

where \mathcal{O} denotes the observed data-set, n is the total number of observations in \mathcal{O} , $\hat{\lambda}_k$ is the ML estimate of the model with k states, and N_k is the total number of free parameters of $\hat{\lambda}_k$.

Notice that in the BIC/MDL criterion, each parameter has equal weight in the penalty term, $\log(n)/2$. In the mixture of Gaussians case, MMDL is based on the following observation: the parameters of the j -th component are actually estimated from the observations that are generated by that component, not from all the observed data. Moreover, the expected number of samples obtained from the j -th component is nc_j , where c_j is the probability of the j -th component. The MMDL criterion for mixtures is then obtained by penalizing each parameter of component j by $\log(nc_j)/2$ (instead of the standard $\log(n)/2$), considering the quantity nc_j as an “effective sample size” for the j -th component.

A similar reasoning can be followed in the HMM context, but care must be taken in the definition of the “effective sample size”, because here there is nothing similar to the component probability c_j . We start by decomposing N_k as $N_k^{\mathbf{A}} + N_k^{\boldsymbol{\pi}} + N_k^{\mathbf{B}}$, denoting the number of parameters of the transition matrix \mathbf{A} , of the initial state probability $\boldsymbol{\pi}$, and of the emission probability density function \mathbf{B} , respectively. Following the MMDL rationale, we will weight the emission probability parameters of each state using the “effective sample size” corresponding to that state. The elements of the transition matrix and of the initial state probability vector will be weighted with the standard $\log(n)/2$, since they are estimated from all the samples.

The role of “state probabilities” (equivalent to c_1, \dots, c_k , in the mixture case) will be played by the stationary probability distribution $\mathbf{p}_\infty = [p_\infty(1), \dots, p_\infty(k)]$. This seems to be a natural choice, since \mathbf{p}_∞ represents the “average” occupation of each state, after the Markov chain has achieved the stationary state. This distribution is computed as follows: consider the Markov chain $\mathbf{Q} = Q_1, Q_2, Q_3 \dots$ with the state set $S = \{S_1, \dots, S_k\}$, the stochastic transition matrix \mathbf{A} , and the initial state probability $\boldsymbol{\pi}$. We can define the vector of state probabilities at time t as

$$\mathbf{p}_t = [p_t(1), \dots, p_t(j), \dots, p_t(k)] = [P(Q_t = S_1), P(Q_t = S_2), \dots, P(Q_t = S_k)]$$

Of course, \mathbf{p}_t can be computed recursively from $\mathbf{p}_1 = \boldsymbol{\pi}\mathbf{A}$, $\mathbf{p}_2 = \mathbf{p}_1\mathbf{A} = \boldsymbol{\pi}\mathbf{A}\mathbf{A}$, and so on. That is $\mathbf{p}_t = \boldsymbol{\pi}\mathbf{A}^t$. We are interested in \mathbf{p}_∞ , which characterizes the equilibrium behavior of the Markov chain, *i.e.*, when we let it evolve indefinitely.

Since it is a stationary distribution, \mathbf{p}_∞ has to be a solution of $\mathbf{p}_\infty = \mathbf{p}_\infty \mathbf{A}$, or, in other words, it has to be a left eigenvector of \mathbf{A} associated with the unit eigenvalue. Under some conditions (see, *e.g.*, [36], for details), the Perron-Frobenius theorem states that matrix \mathbf{A} has a unit (left) eigenvalue and the corresponding left eigenvector is \mathbf{p}_∞ . All other eigenvalues of \mathbf{A} are strictly less than 1, in absolute value. Finding \mathbf{p}_∞ for a given \mathbf{A} then amounts to solving the corresponding eigenvalue/eigenvector problem.

Coming back to the MMDL formulation, we have that, for an HMM with k states, the MMDL cost function is

$$\text{MMDL}(k) = \log p(\mathcal{O}|\hat{\lambda}_k) - \frac{N_k^{\mathbf{A}} + N_k^{\boldsymbol{\pi}}}{2} \log(n) - \frac{N_1^{\mathbf{B}}}{2} \sum_{m=1}^k \log(n p_\infty(m))$$

where $N_1^{\mathbf{B}}$ is the number of parameters of the emission density of an HMM with just one state. Finally, notice that \mathbf{A} has $k(k-1)$ free parameters, $\boldsymbol{\pi}$ has $(k-1)$ free parameters, and $N_1^{\mathbf{B}} = d + d(d+1)/2$, if we assume a full covariance matrix for each component and d -dimensional observations. Accordingly, after dropping all terms that do not depend on k ,

$$\text{MMDL}(k) = \log p(\mathcal{O}|\hat{\lambda}_k) - \frac{k^2}{2} \log(n) - \frac{d^2 + 3d}{4} \sum_{m=1}^k \log(n p_\infty(m)) \quad (3.38)$$

Notice that $\mathbf{p}_\infty = [p_\infty(1), \dots, p_\infty(k)]$ is a function of $\hat{\lambda}_k$ via the estimate of the transition matrix.

3.6.3 The sequential state pruning strategy

The strategy is summarized as follows:

1. Choose some model selection criterion, such as BIC/MDL (Eq. (3.34)), or MMDL (Eq. (3.38)); set k_{min} and k_{max} , which are the minimum and maximum number of states allowed.
2. Initialize the HMM estimation algorithm with k_{max} states using some standard heuristic (*e.g.*, randomly, or using clustering). Let us denote as λ_k^I the initial model used in the training procedure for the HMM with k states.
3. While $k \geq k_{min}$, do:
 - a) run the Baum-Welch algorithm until some convergence criterion is met; let $\hat{\lambda}_k$ be the set of estimated parameters.
 - b) compute and store the value of the model selection criterion; let this be denoted as C_k .
 - c) find the least probable state (*i.e.*, the smallest element of \mathbf{p}_∞);
 - d) prune the least probable state and deleting the corresponding elements from \mathbf{A} , \mathbf{B} , obtaining a reduced model $\bar{\lambda}$.
 - e) set $\lambda_{k-1}^I \leftarrow \bar{\lambda}$, and $k \leftarrow k - 1$.
4. The final chosen model, λ^* , is the one yielding the maximum of the selection criterion. Formally:

$$\lambda^* = \hat{\lambda}_{k^*}, \quad \text{where } k^* = \arg \max_k C_k$$

The computational overhead introduced by this procedure is due mostly to the computation of \mathbf{p}_∞ , involving the computation of eigenvalues of \mathbf{A} . However, this is computed only once for each k , at the end of the Baum-Welch training session. For the MMDL approach, there is actually no computational overhead, since \mathbf{p}_∞ is also needed when evaluating the selection criterion (equation (3.38)).

3.6.4 Testing

To assess the performance of the proposed approach, we have performed tests in which we compare two strategies:

- Standard BIC (or MMDL) method: we train one HMM for each k (number of states), with k varying from k_{max} to k_{min} . Each learning session (Baum-Welch algorithm) is initialized using a Gaussian mixture model, which is better than the usual random initialization. Each learning session is stopped when the relative increase of the likelihood function falls below a threshold. For each k , we compute and store the BIC (or MMDL) value, and, finally, we choose the model yielding the best value.
- Pruning BIC (or MMDL) method: as described in Section 3.6.3.

In all the considered HMMs, the emission probability density of each state is a single univariate Gaussian. The two strategies are compared in terms of 1) accuracy of the model size estimation, 2) total computational cost (total number of iterations) required by Baum-Welch procedure, and 3) classification accuracy on three recognition tasks (one synthetic and two real data problems).

Accuracy of model selection

We have tested our procedure on three different problems. For each one, the test set contains 5 sequences, each 400 observations long, synthetically generated from a known HMM. To increase statistical significance, all experiments were repeated 50 times. We set k_{min} and k_{max} to 2 and 10, respectively.

The first model is shown in Fig. 3.11(a): \mathbf{A} is the transition matrix, $\boldsymbol{\pi}$ is the initial state probability, and $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are the means and variances of the Gaussian emission densities of each state. This is a relatively simple model, where Gaussians of different states are very well separated. The results (in Table 3.8(a)) show that, with regards to the accuracy in the selection of the true k , all model selection procedures perform perfectly. Regarding the computational requirements, the pruning strategy is less demanding, requiring about the 77% of the number of Baum-Welch iterations of the normal procedure.

The second model is more challenging, since two of the emission Gaussians overlap, with common mean but different variances (see Fig. 3.11(b)). Also in this case, there is no difference between BIC and MMDL, but there is a great difference between the two training strategies. In Table 3.8(b), the accuracies are reported, showing that the pruning methodology performs perfectly, with 100% accuracy, whereas the accuracy is 54% for the standard algorithm. Nearly an half of the models selected with the normal strategy have a wrong number of states (typically too many). This is confirmed in Fig. 3.12(a), where histograms of the

$$\mathbf{A} = \begin{array}{|c|c|c|c|} \hline 0.25 & 0.25 & 0.25 & 0.25 \\ \hline 0.25 & 0.25 & 0.25 & 0.25 \\ \hline 0.25 & 0.25 & 0.25 & 0.25 \\ \hline 0.25 & 0.25 & 0.25 & 0.25 \\ \hline \end{array} \quad \boldsymbol{\pi} = \begin{array}{|c|} \hline 0.25 \\ \hline 0.25 \\ \hline 0.25 \\ \hline 0.25 \\ \hline \end{array} \quad \mathbf{B} = \begin{array}{|c|c|} \hline \mu_1 = -2 & \sigma_1^2 = 0.5 \\ \hline \mu_2 = 0 & \sigma_2^2 = 0.5 \\ \hline \mu_3 = 2 & \sigma_3^2 = 0.5 \\ \hline \mu_4 = 4 & \sigma_4^2 = 0.5 \\ \hline \end{array}$$

(a)

$$\mathbf{A} = \begin{array}{|c|c|c|c|} \hline 0.25 & 0.25 & 0.25 & 0.25 \\ \hline 0.25 & 0.25 & 0.25 & 0.25 \\ \hline 0.25 & 0.25 & 0.25 & 0.25 \\ \hline 0.25 & 0.25 & 0.25 & 0.25 \\ \hline \end{array} \quad \boldsymbol{\pi} = \begin{array}{|c|} \hline 0.25 \\ \hline 0.25 \\ \hline 0.25 \\ \hline 0.25 \\ \hline \end{array} \quad \mathbf{B} = \begin{array}{|c|c|} \hline \mu_1 = 0 & \sigma_1^2 = 0.2 \\ \hline \mu_2 = 0 & \sigma_2^2 = 2 \\ \hline \mu_3 = -4 & \sigma_3^2 = 0.5 \\ \hline \mu_4 = 4 & \sigma_4^2 = 0.5 \\ \hline \end{array}$$

(b)

$$\mathbf{A} = \begin{array}{|c|c|c|c|} \hline 0.85 & 0.05 & 0.05 & 0.05 \\ \hline 0.05 & 0.85 & 0.05 & 0.05 \\ \hline 0.05 & 0.05 & 0.85 & 0.05 \\ \hline 0.05 & 0.05 & 0.05 & 0.85 \\ \hline \end{array} \quad \boldsymbol{\pi} = \begin{array}{|c|} \hline 0.25 \\ \hline 0.25 \\ \hline 0.25 \\ \hline 0.25 \\ \hline \end{array} \quad \mathbf{B} = \begin{array}{|c|c|} \hline \mu_1 = 0 & \sigma_1^2 = 0.2 \\ \hline \mu_2 = 0 & \sigma_2^2 = 2 \\ \hline \mu_3 = -4 & \sigma_3^2 = 0.5 \\ \hline \mu_4 = 4 & \sigma_4^2 = 0.5 \\ \hline \end{array}$$

(c)

Fig. 3.11. Three models for the synthetic data test: \mathbf{A} is the transition matrix, $\boldsymbol{\pi}$ is the initial state probability, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ the parameters of the emission density.

Table 3.8. Results on synthetic data. (a) First, (b) second, and (c) third experiments.

	Selection accuracy	Avg. iterations
Standard BIC	50/50 (100%)	110
Standard MMDL	50/50 (100%)	110
Pruning BIC	50/50 (100%)	84
Pruning MMDL	50/50 (100%)	84

(a)

	Selection accuracy	Avg. iterations
Standard BIC	27/50 (54%)	175
Standard MMDL	27/50 (54%)	175
Pruning BIC	50/50 (100%)	103
Pruning MMDL	50/50 (100%)	103

(b)

	Selection accuracy	Avg. iterations
Standard BIC	43/50 (86%)	186
Standard MMDL	43/50 (86%)	186
Pruning BIC	49/50 (98%)	98
Pruning MMDL	49/50 (98%)	98

(c)

selected numbers of states are shown. Also in this case, the average number of iterations required by the pruning method is significantly lower.

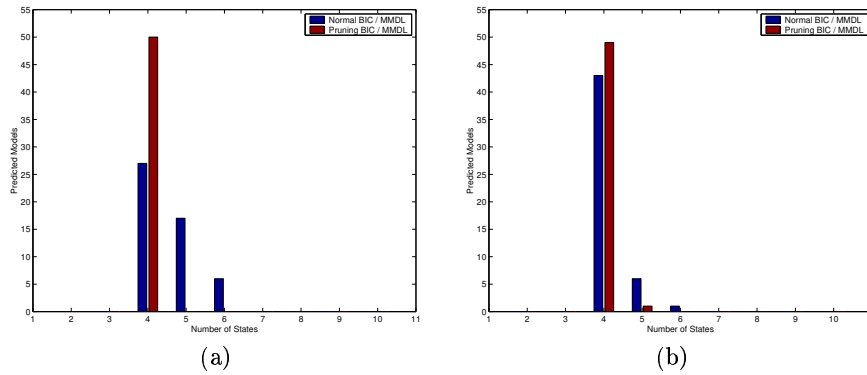


Fig. 3.12. Histograms of the selected number of states for the standard and the pruning strategies; the correct number of states is 4. (a) Second and (b) third experiments.

The third model is obtained from the second one by changing the transition matrix (see Fig. 3.11(c)). From Table 3.8(c) and Fig. 3.12(b), it is clear that, also in this example, the pruning strategy performs better, with a nearly perfect accuracy, versus about 86% for the standard method. The average number of iterations required by the pruning strategy is 52.7% of that required by the normal procedure.

Classification Accuracy

We now study the performances of the proposed method in terms of classification accuracy on recognition tasks, using both synthetic and real data.

Synthetic Data

In order to test the classification accuracy of the two methods, we have used the following testing procedure.

- Two training sets are generated, according to two models, each corresponding to one of two different classes.
- Two HMMs, one for each class, are trained using both methods (pruning and standard) and both model selection criteria (BIC and MMDL).
- Two test sets from the same true models are then generated.
- The classification accuracies using these test sets (a sequence is assigned to the class whose model has the highest likelihood), are finally estimated.

For each model, the training set contained 5 sequences of length 400. The test set was composed by 20 sequences, 10 from the first class and 10 from the second. To increase statistical significance, experiments were repeated 25 times. As before, $k_{min} = 2$ and $k_{max} = 10$.

In the first experiment, the HMM models used for each class are those shown in Figs. 3.11(b) and 3.11(c), only differing in the transition matrix \mathbf{A} . The experimental results are shown in Table 3.9(a). Both techniques perform perfectly, with the pruning method requiring fewer Baum-Welch iterations.

Table 3.9. Classification accuracies on synthetic data. (a) First and (b) second experiments.

	Classification accuracy		Avg. iterations
	Mean	Std. Dev.	
Normal BIC	20/20 (100%)	0/20	110
Normal MMDL	20/20 (100%)	0/20	110
Pruning BIC	20/20 (100%)	0/20	84
Pruning MMDL	20/20 (100%)	0/20	84

(a)

	Classification accuracy		Avg. iterations
	Mean	Std. Dev.	
Normal BIC	18.44/20 (92.2%)	2.31/20	163
Normal MMDL	18.44/20 (92.2%)	2.31/20	163
Pruning BIC	19.60/20 (98.0%)	0.76/20	107
Pruning MMDL	19.60/20 (98.0%)	0.76/20	107

(b)

The second classification task considered was a very difficult one: the first model is the one shown in Fig. 3.11(b), the second one is almost the same, the only difference being the variance of the Gaussian of the first state: 0.4 instead of 0.2. The two HMMs are quite similar, but, as we can see in Table 3.9(b), the classification performance is very good. More in detail, the pruning strategy is better, with an accuracy of 98%, *i.e.*, 6% larger than that of the standard procedure. In this case, the effectiveness of the learning is crucial for the correct discrimination. Moreover, the number of iterations required in the training phase is reduced for the pruning method, nearly 65% of the standard method.

Real Data

Finally, we have conducted two classification experiments with real data. The first one involves a 2D shape recognition problem, using HMMs as described in Chapter 6. The second is a face recognition experiment, using HMMs as proposed in [125], described in Section 7.2.

The 2D shape recognition test is performed on a part of the data set described in [197], with four classes, each containing 12 different shapes. An object from each class is shown in Fig. 3.13. Just as in the synthetic experiments above reported, the pruning method performs better on this real-data problem (see Table 3.10), and involves a smaller computational burden. The classification accuracies reported are computed using the leave-one-out (LOO) method. This means that the training set is each time different, composed by all sequences except one, while the remaining is left out and used for testing. The left out sequence changes until all sequences have been tested, and results are averaged. Experiments were repeated 10 times to increase the statistical significance.

Face recognition is addressed using the method proposed in [125], highly detailed in Section 7.2, and here briefly resumed. The approach considers DCT (Discrete Cosine Transform) coefficients as features; given a sequence of sub images of



Fig. 3.13. Examples of shapes from database used.

Table 3.10. Classification accuracies on real data considering 2D shape classification.

	Classification accuracy		Avg. iterations
	Mean	Std. Dev.	
Normal BIC	44.4/48 (92.5%)	1.26/48	94.1
Normal MMDL	45.3/48 (94.37%)	0.95/48	94.1
Pruning BIC	45.7/48 (95.21%)	0.48/48	76.6
Pruning MMDL	45.7/48 (95.21%)	0.67/48	76.6

the face image, obtained with a raster scanning, the DCT coefficients of each sub image are computed, and vectorized using a *zig-zag* scan. The chosen number of coefficients determines the dimensionality of the observation, and 10 coefficients are used in our experiment. Subimages were of dimension 16x16, with an overlap of 50%. The experiments have been conducted on the ORL database³, which consists in 40 subjects with 10 faces each. For each subject, five faces were used for training and the others for testing. The results, shown in Table 3.11, were obtained by repeating the experiments 25 times and averaging the results. They are very satisfactory: the classification accuracies are similar, but our method reduces substantially the number of the iterations required.

Table 3.11. Classification accuracies on real data: face recognition.

	Classification accuracy		Avg. iterations
	Mean	Std. Dev.	
Normal BIC	194.35/200 (97.17%)	1.54/200	86.2
Normal MMDL	194.85/200 (97.42%)	1.64/200	86.2
Pruning BIC	195.85/200 (97.42%)	0.95/200	51.4
Pruning MMDL	195.21/200 (97.61%)	0.92/200	51.4

A general consideration could be done looking at the standard deviations presented in all results tables: performances of the proposed approach are more stable, as the corresponding standard deviations are lower than those obtained with standard techniques. This confirms the fact that with our method the initialization is better addressed, resulting in a more stable and initialization-independent training process.

³ Downloaded from <http://www.uk.research.att.com/facedatabase.html>.

Comparison between BIC and MMDL criteria

In all synthetic experiments, the BIC and MMDL criteria chose the same topology, leading to the same model selection accuracy. Nevertheless, in the real-data case, the MMDL criterion slightly outperforms BIC in the resulting classification accuracy, showing that, as claimed in [74], in some cases this criterion is better able to select a more suitable model structure.

3.6.5 Conclusions

The key idea of the proposed approach is to perform a decreasing learning strategy, starting each training session from a “nearly good” configuration, derived from previous training by pruning the “least probable” state. The proposed strategy can be applied for all types of HMMs and can be used with any model selection criterion. We have considered the Bayesian inference criterion (BIC), and we have adapted the mixture minimum description length (MMDL) criterion to the HMM case. Experimental results on synthetic and real problems are really promising, since the proposed approach, in these experiments, is more accurate in finding the true model, more effective in classification accuracy, while having reduced computational requirements. Moreover, the performances of the proposed approach are more stable, as the corresponding standard deviations are lower than those obtained with standard techniques. This suggests that with the proposed method the initialization is better addressed, resulting in a more stable and initialization-independent training process.

Classification with HMM

The classification of sequential data is an interesting and important research area. Its importance has rapidly grown in the last years for both methodological and applicative reasons. From a methodological point of view, probabilistic modelling and classification of sequences is a challenging problem, because intrinsically more difficult than in the standard scenario, where each observation is a set (vector) of features. In fact, since the sequences length may vary, it is not possible to directly use standard pattern recognition techniques. With regards to application, some sequence classification problems have become very popular in recent years, as DNA and protein modelling or data mining: these problems, moreover, usually involve very large data sets.

Hidden Markov Models (HMMs) are one of the widest employed probabilistic models for sequential data, mostly applied to classification problems, for which a standard and well established protocol exists. This standard scheme, which is directly derived from the Bayesian approach to the classification problem, is presented in the next section.

Some questions could arise about this classification scheme: is this scheme reliable? Is it possible to measure the trustworthiness of a classification? Are there alternative schemes that could be adopted? Some considerations about the first two questions are presented in Section 4.2; the last question is directly addressed in Section 4.3, where an alternative scheme is proposed, inspired by the similarity-based classification paradigm.

4.1 Standard classification scheme

The standard HMM-based approach to sequence classification is directly derived from the standard Bayes classification paradigm [65]. This paradigm is strictly linked to the Bayesian theory explained in Section 3.2.4, which refers to parameter estimation. The Bayesian classification paradigm is simple: it assigns an unclassified item x to the class showing the maximum *a posteriori* probability, *i.e.*

$$\ell(x) = \arg \max_i P(C_i|x) \tag{4.1}$$

where $\ell(x)$ is the class (or label) assigned to the pattern x by the classifier, $\ell(x) \in \{1..C\}$, with C is the number of classes. The main problem of this approach is that posterior probability is usually not known, and also difficult to estimate from data in a direct manner; fortunately, there is a theorem that provides a simpler way to compute this quantity, the Bayes theorem [65]:

$$P(C_i|x) = \frac{P(x|C_i)P(C_i)}{P(x)} \quad (4.2)$$

where $P(C_i)$ represents the *a priori* probability of the class C_i , and $P(x|C_i)$ is the *likelihood*, representing the probability that the pattern x has been generated from the class C_i . These quantities are more easily estimable from data than the posterior probability. Since $P(x)$ is independent from the class, the criterion (4.1) could be rewritten as

$$\ell(x) = \arg \max_i P(x|C_i)P(C_i) \quad (4.3)$$

In the HMM case, the conditional class probabilities $P(x|C_i)$ are modelled with HMMs, one for each class. In this way the class likelihood of an unknown sequence \mathbf{O} could be computed as $P(\mathbf{O}|C_i) = P(\mathbf{O}|\lambda_i)$. By applying the Bayesian classification rule (4.3), and by assuming *a priori* equiprobable classes, so that the prior probability $P(C_i)$ could be dropped from the criterion (4.3), we have that the unknown sequence \mathbf{O} will be assigned to the class whose model shows the highest likelihood, *i.e.*

$$\ell(\mathbf{O}) = \arg \max_i P(\mathbf{O}|\lambda_i) \quad (4.4)$$

where λ_i is the HMM modelling the i -th class. This rule is called the *Maximum Likelihood* (ML) classification rule, and represents the typical HMM-based classification scheme. In the sequel, in order to distinguish it from the next method, we will call this the ML_{OPC} approach (with OPC standing for “one per class”).

A somewhat different rule could also be used in some contexts (see for example Chapter 6). Instead of training one HMM for each class, we could train one model for each training sequence, and assign an unknown sequence \mathbf{O} to the class of the model showing the highest likelihood. More formally, let $\lambda_i^{(k)}$ denote the HMM model trained on sequence $\mathbf{O}_i^{(k)}$, which belongs to class k . The classification rule, under this approach, is then

$$\ell(\mathbf{O}) = \arg \max_{i,k} P(\mathbf{O}|\lambda_i^{(k)}) \quad (4.5)$$

We call this the ML_{OPS} approach (with OPS standing for “one per sequence”). Notice that this may be seen as a 1-nearest-neighbor (1-NN) classifier, with the proximity measure defined by the likelihood function.

4.2 Reliability of the classification scheme

Both classification schemes are Maximum Likelihood approaches, where the model showing the highest likelihood “wins”, *i.e.* decides the class to be assigned. Unfortunately, this criterion is not able *per se* to provide a “reliability” measure of the

classification decision. For instance, consider a two-class problem, with two models, λ_1 and λ_2 , an unknown sequence \mathbf{O} , and two possible different situations. In the first situation, we have $P(\mathbf{O}|\lambda_1) > P(\mathbf{O}|\lambda_2)$, with $P(\mathbf{O}|\lambda_1) \approx P(\mathbf{O}|\lambda_2)$, whereas in the second $P(\mathbf{O}|\lambda_1) \gg P(\mathbf{O}|\lambda_2)$. In both cases, the classification scheme assigns \mathbf{O} to the class C_1 , but the “reliability” of the second classification is without doubt much higher, as the system is more “sure” that the class is C_1 .

Starting from this consideration, two measures of the “reliability” of the classification are proposed, able to quantify, in a likelihood sense, the potential robustness of the classification. The two introduced measures are defined for the correctly classified patterns and for the misclassified patterns, respectively. In particular, the former, called RCC (Reliability in Correct Classification), is determined by computing, in each correctly classified experiment, the log-likelihood difference between the “winning” model and the second choice, and then averaging between all correctly classified patterns. More in detail, given the HMMs $\lambda_1.. \lambda_C$, modelling the C classes, and the set of sequences to be classified $\mathcal{T} = \{\mathbf{O}_1.. \mathbf{O}_N\}$, let us denote as $\ell(\mathbf{O}_i)$ the class assigned to the sequence \mathbf{O}_i by the classifier, and as $c(\mathbf{O}_i)$ the corresponding true class. The $\text{RCC}(\mathcal{T})$ factor is then defined as

$$\text{RCC}(\mathcal{T}) = \frac{1}{\epsilon_1(\mathcal{T})} \sum_{\mathbf{O}_i \in \mathcal{T} \text{ s.t. } \ell(\mathbf{O}_i) \equiv c(\mathbf{O}_i)} \text{rcc}(i) \quad (4.6)$$

where $\epsilon_1(\mathcal{T})$ is the number of correctly classified patters in \mathcal{T} , and $\text{rcc}(i)$ is defined as

$$\text{rcc}(i) = \left| P(\mathbf{O}_i | \lambda_{\ell(\mathbf{O}_i)}) - \max_{j \neq \ell(\mathbf{O}_i)} P(\mathbf{O}_i | \lambda_j) \right| \quad (4.7)$$

This value could be considered as a likelihood-based measure of the “safety” or “reliability” of the classification, because the larger this measure, the more reliable the classification result.

In the case of misclassified patterns, the complementary reasoning could be adopted: we could compute the log-likelihood difference between the chosen model and the model of the actual object class. This results in defining the factor REC (Reliability in Erroneous Classification), defined as

$$\text{REC}(\mathcal{T}) = \frac{1}{\epsilon_2(\mathcal{T})} \sum_{\mathbf{O}_i \in \mathcal{T} \text{ s.t. } \ell(\mathbf{O}_i) \neq c(\mathbf{O}_i)} \text{rec}(i) \quad (4.8)$$

where $\epsilon_2(\mathcal{T})$ is the number of misclassified patterns in \mathcal{T} , and $\text{rec}(i)$ is defined as

$$\text{rec}(i) = \left| P(\mathbf{O}_i | \lambda_{\ell(\mathbf{O}_i)}) - P(\mathbf{O}_i | \lambda_{c(\mathbf{O}_i)}) \right| \quad (4.9)$$

This value quantifies the distance, in terms of likelihood, between the correct choice and the (wrong) classifier choice, providing a kind of measure of the size of the “classification error”.

These measures are aimed at quantifying the reliability of the decision taken by the classification scheme. It is worth noting that these measures do not represent a rejection rule, and do not provide any precise information about the accuracy-rejection tradeoff of the classifier system. Rejection is a widely investigated concept

in the Pattern Recognition area, and consists of a rule that decides not to classify an object if there is not a sufficient confidence that the decision will be correct. In the case of rejection the classification is delegated to other more sophisticated procedures. However, high rejection rate could lead to large time consuming efforts (due to the sophisticated procedures), therefore a tradeoff between accuracy and rejection is mandatory. The formulation of the best accuracy-reject tradeoff is presented in the seminal work of Chow [48,49], where the related optimal reject rule is also derived. In this rule, a thresholding is applied to the posterior probability (or to the likelihood, if *a priori* equiprobable classes are assumed):

$$\text{if } \max_i P(C_i | \mathbf{O}) \begin{cases} \geq \theta, & \text{then classify } \mathbf{O} \\ < \theta, & \text{then reject } \mathbf{O} \end{cases} \quad (4.10)$$

where θ is a threshold in the range $(0, 1)$. The large the value of θ , the fewer points will be classified.

The measures introduced in this thesis do not match this formulation, since no rejection rule is proposed here. The presented quantities represent only a measure of the “robustness” of a classification, *i.e.* a measure of how much “sure” the system decisions are. In this sense, our measures share the same philosophy of those presented in [85], where the introduced quantities estimate the aptitude of a classifier to reject errors without rejecting correct classification. As in our case, these two measures are defined in the correct classification case and in the misclassification case. The elegant basic concept under these measures is the entropy of the classification probability. Also in that case, if required, a rejection rule should be further derived. In our case, to obtain the rejection rule, the proposed measures should be reformulated in the Chow’s work context, in which Chow’s rejection rule, or other more complicated ones (e.g. [82]), could be applied.

4.2.1 Experimental evaluation

The objective of this section is to assess the appropriateness of the defined measures, by analyzing their experimental behavior in difficulty-controllable real and synthetic tasks.

Synthetic case

In this case, the goal is the following: given a classification task, with an increasing and controllable degree of difficulty, we want to see if the difficulty is captured by the proposed REC and RCC factors. If these factors are well defined, the expectation is that the RCC factor (regarding correctly classified patterns) will increase as difficulty decreases, whereas the REC factor (regarding misclassified patterns) will show the opposite behavior, *i.e.* it will grow as the complexity of the task increases.

In order to partially control the difficulty of the task we propose a three-class problem, when the originating HMMs are three states models, proposed in Fig. 4.1.

One can notice that the three models are very similar: they shares the same transition matrix and the same initial state probability. Moreover the emission

$$\begin{aligned}
\mathbf{A} &= \begin{array}{|c|c|c|} \hline 1/3 & 1/3 & 1/3 \\ \hline 1/3 & 1/3 & 1/3 \\ \hline 1/3 & 1/3 & 1/3 \\ \hline \end{array} \quad \boldsymbol{\pi} = \begin{array}{|c|} \hline 1/3 \\ \hline 1/3 \\ \hline 1/3 \\ \hline \end{array} \quad \mathbf{B} = \begin{array}{|c|c|} \hline \mu_1 = 1 & \sigma_1^2 = 0.4 \\ \hline \mu_2 = 3 & \sigma_2^2 = 0.4 \\ \hline \mu_3 = 5 & \sigma_3^2 = 0.4 \\ \hline \end{array} \\
&\text{(a)} \\
\mathbf{A} &= \begin{array}{|c|c|c|} \hline 1/3 & 1/3 & 1/3 \\ \hline 1/3 & 1/3 & 1/3 \\ \hline 1/3 & 1/3 & 1/3 \\ \hline \end{array} \quad \boldsymbol{\pi} = \begin{array}{|c|} \hline 1/3 \\ \hline 1/3 \\ \hline 1/3 \\ \hline \end{array} \quad \mathbf{B} = \begin{array}{|c|c|} \hline \mu_1 = 1 & \sigma_1^2 = 0.4 + \delta \\ \hline \mu_2 = 3 & \sigma_2^2 = 0.4 + \delta \\ \hline \mu_3 = 5 & \sigma_3^2 = 0.4 + \delta \\ \hline \end{array} \\
&\text{(b)} \\
\mathbf{A} &= \begin{array}{|c|c|c|} \hline 1/3 & 1/3 & 1/3 \\ \hline 1/3 & 1/3 & 1/3 \\ \hline 1/3 & 1/3 & 1/3 \\ \hline \end{array} \quad \boldsymbol{\pi} = \begin{array}{|c|} \hline 1/3 \\ \hline 1/3 \\ \hline 1/3 \\ \hline \end{array} \quad \mathbf{B} = \begin{array}{|c|c|} \hline \mu_1 = 1 & \sigma_1^2 = 0.4 + 2\delta \\ \hline \mu_2 = 3 & \sigma_2^2 = 0.4 + 2\delta \\ \hline \mu_3 = 5 & \sigma_3^2 = 0.4 + 2\delta \\ \hline \end{array} \\
&\text{(c)}
\end{aligned}$$

Fig. 4.1. Three generative models used for synthetic testing: (a) class 1, (b) class 2, and (c) class 3.

probabilities of the states are Gaussians, where means are equal: the only difference between the three models lies in the variance of the Gaussians, differing of δ . Delta (δ) is the parameter used to drive the difficulty of the classification task: the larger the δ , the easier the task.

The experimental setup is the following: 30 sequences (with the length of 400) were generated for each class. HMMs are trained using standard Baum-Welch re-estimation procedure, initialized with Gaussian Mixture Models as described in Section 3.4.1. The training process was stopped after likelihood convergence. In order to guarantee statistical significance to the experiments, each session was repeated 20 times, averaging REC and RCC factors. The variable δ varies from 0.025 to 0.25. REC and RCC were plotted in Fig. 4.2, together with the classification results.

From this plot it could be seen that the expected behavior was confirmed: the RCC factor, computed for correctly classified patterns, increases as difficult increases, *i.e.* as the parameter δ increases. On the other hand, the REC factor, computed on misclassified patterns, does not show a particularly expressive pattern. The only interesting consideration is that it is really low if compared to the RCC factor: so we could infer that the classification is not reliable.

One important observation has to be made: these indices, especially the RCC one, are able to quantitatively describe the accuracy of the system in a classification task, also in case of perfect classification. In fact, we could note that, for $\delta \geq 0.125$, the system reaches an almost perfect accuracy. Nevertheless, the RCC index increases for decreasing difficulty, allowing to understand that the task is easier: this information could not be deduced by merely analyzing the classification accuracies, that are almost the same. This is more evident in the real case proposed in the next section, where HMM accuracy is 100% in some situations.

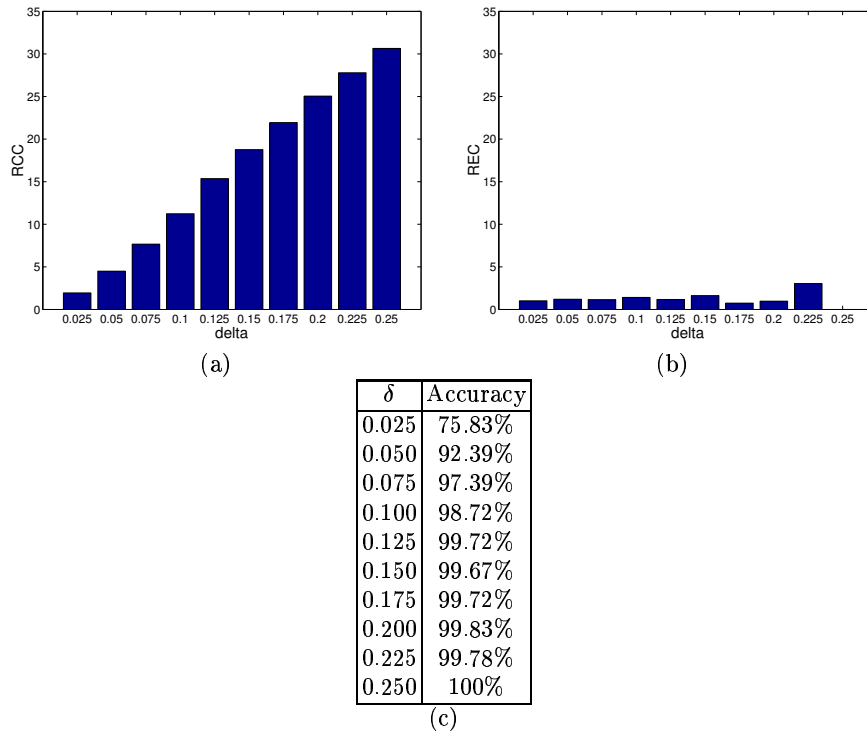


Fig. 4.2. Plots of RCC (a) and REC (b) factors in the synthetic experiments, together with the correspondent classification accuracy (c). All plots are in the same scale.

Real case

The proposed measures have been computed also in the 2D shape recognition task, using the method proposed in Chapter 6, in the presence of occlusions and in the presence of noise separately. For the occlusion experiments, the two proposed measures are plotted, using the same scale, in Fig. 4.3, together with the corresponding classification accuracies.

In Fig. 4.3(a), as expected, the RCC factor increases as occlusion level decreases. In Fig. 4.3(b), only REC values for occlusion levels higher than 35% are plotted, as no errors are made for lower occlusion levels. From these values, it can be observed that the REC factor is very low, even lower than the margin estimated in the case of correct classification (Fig. 4.3(a)). Here it is more evident the fact that the perfect classification rates, proposed by the system for occlusion lower than 35%, do not correspond to the same task ease. This information could be evinced only by looking at the RCC index values.

The same behavior can be noticed looking at the reliability analysis for the experiments in presence of noise (Fig. 4.4(a) and (b)): the margin between the log-likelihood difference in the two cases is narrowed, but it is still possible to discriminate between correct and wrong classifications.

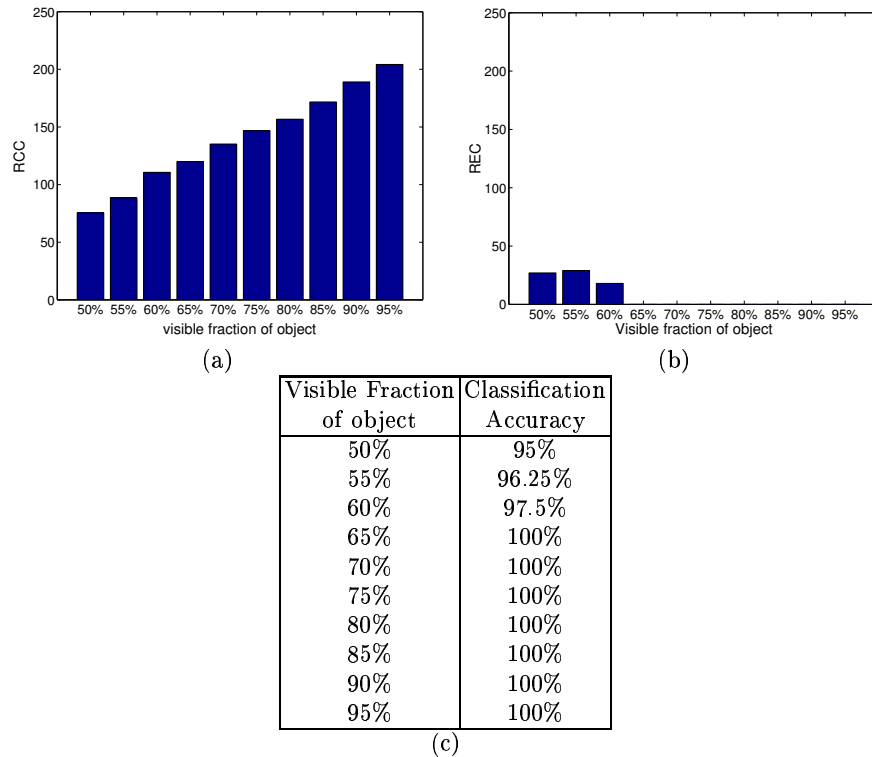


Fig. 4.3. Analysis of the reliability of the classification rule in occlusion experiments, at different occlusion levels: (a) RCC factor, (b) REC factor, and (c) the corresponding classification accuracy. All plots are on the same scale.

4.3 Classification by similarity

As explained in Section 4.1, the standard HMM-based approach to sequence classification consists in training one HMM for each class, and to classify an unknown sequence into the class whose model shows the highest probability (*likelihood*) of having generated this sequence (*Maximum Likelihood* (ML) classification rule).

In this section, an alternative classification scheme is proposed, by extending the similarity-based paradigm [110, 87, 107, 169, 173, 170] to HMM-based classification. This paradigm, which has recently been introduced, differs from typical pattern recognition approaches where objects to be classified are represented by sets (vectors) of features. In the similarity-based paradigm, objects are described using pairwise (dis)similarities, *i.e.*, distances from other objects in the data set. Thus, objects are not constrained to be explicitly represented in a feature space; all that is necessary is a way to compute (dis)similarities between pairs of objects. The goal is then to learn a classifier only from these relational data.

The literature on similarity-based classification is not vast [110, 87, 107, 169, 173, 170], and is summarized in Section 4.3.1. The general idea behind all these approaches is basically the same: given a set of pairwise dissimilarity values, a

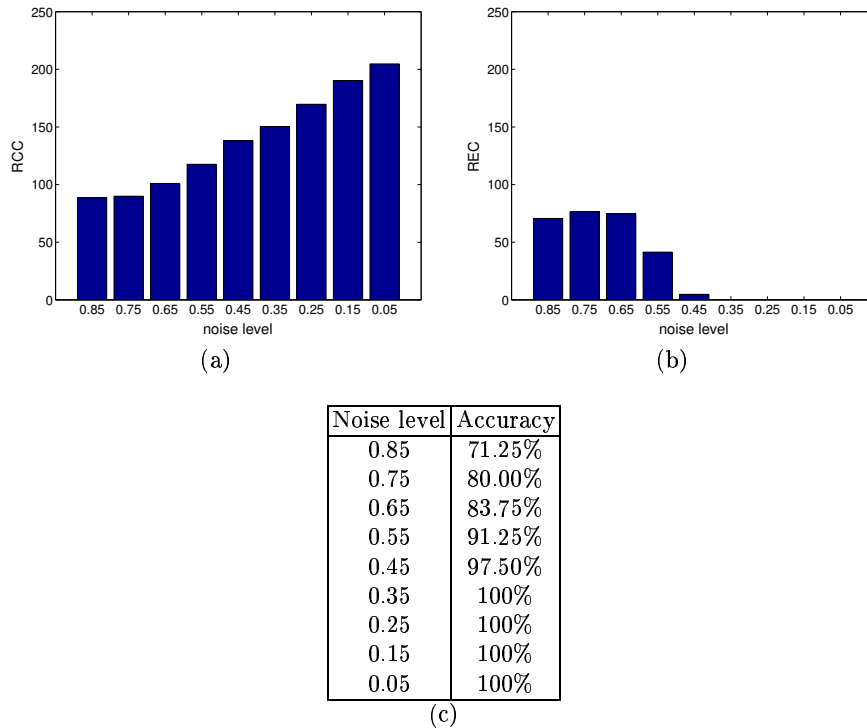


Fig. 4.4. Analysis of the reliabilities of the classification rule in noise experiments, at different noise levels: (a) RCC factor, (b) REC factor, and (c) the corresponding classification accuracy. All plots are on the same scale.

new representation space can be built, in which each object is described by these values. In [173], a simple synthetic experiment shows that a complex problem in a 2D space (requiring a quadratic classifier to achieve almost correct separation), becomes a linearly separable problem in a dissimilarity space.

The proposed approach extends this dissimilarity-based classification paradigm to HMM-based sequences classification problems. We propose to build a similarity¹ space, representing each object (sequence) by the vector of its similarities with respect to a predetermine set of objects (this can be the whole data set, in the simplest approach), called the *representatives set*; the classification is then performed in this new representation space. Similarities are derived by considering the likelihood $P(\mathbf{O}|\boldsymbol{\lambda})$ as a measure of the *similarity* between the sequence \mathbf{O} and the HMM specified by the set of parameters $\boldsymbol{\lambda}$. This similarity measure was previously used in sequence clustering applications [201, 167] (see also Chapter 5).

The similarity-based classification paradigm seems to be particularly well suited to HMMs, as it can be seen as a natural extension of the standard HMM classification scheme. Specifically, the standard Maximum Likelihood criterion assigns an unknown sequence \mathbf{O} to the class whose model shows the highest likeli-

¹ Note that we refer indifferently to similarity or dissimilarity.

hood. To this aim, the likelihoods of \mathbf{O} with respect to the HMMs of all classes are evaluated, each stating a *likelihood-based* measure of the similarity between that class and the observed sequence. In other words, HMMs are used to compute *similarities* between sequences and classes, with each class being represented by a single HMM. Subsequently, only the maximum of these values is used to take the classification decision. In the similarity-based approach, the classification decision is taken using the *whole* set of similarities between each observed sequence and all the other sequences. We will show that this strategy results in a substantial improvement in the classification performance, compared to standard HMM-based approaches. Moreover, with the use of HMMs and the similarity representation, the problem of sequences classification is reduced to a more standard classification task (where each object is described by a fixed-length feature vector), for which arbitrarily sophisticated techniques can be used, allowing to increase even more the classification performance.

The proposed approach was successfully tested on both synthetic and real data, involving 2D shape recognition and face recognition problems. In comparison with the standard HMM-based Maximum Likelihood classification criterion, our method showed a significant performance improvement in these problems, confirming all of the potentialities of the similarity-based classification approach.

The main problem of the similarity-based approach, of particular relevance in practical applications, is the high dimensionality of the resulting similarity space. Actually, in the basic approach, this dimensionality is equal to the cardinality of the whole training data set, possibly leading to a huge computational burden. In the literature, two types of solutions of this problem could be identified, summarized in Section 5. Here, three methods to face this problem are proposed. The first one aims at removing redundancy from the data by applying linear dimensionality reduction techniques, such as *Fisher Discriminant Analysis* (FDA) [81] and *Principal Component Analysis* (PCA) [113]. The second proposed method is based on a greedy strategy known as *matching pursuit* [153], which selects a subset of representatives based on which the similarity values are computed. These two approaches are very general, and can be applied in all distance-based classification contexts. The third proposed approach is more specific to the HMM case, and is based on a simple adaptation of the similarity-based classification approach to the standard HMM learning procedure. All these approaches were experimentally evaluated, showing the promising discriminative power of the similarity space, even when the dimensionality is reduced to a more manageable size.

Briefly, the main contribution is the introduction of the similarity-based recognition paradigm in an HMM context, resulting in a significant performance improvement with respect to standard HMM-based classification. The mapping to the similarity space, proposed in our approach, allows to reduce complex problem of sequence classification to a more standard point classification problem, for which arbitrarily complex techniques could be used.

From the point of view of similarity-based recognition, we propose two different approaches to deal with the high dimensionality of the similarity space, which is one of the main problems of the method. First, the potential of linear reduction techniques, as PCA and FDA, is exploited, showing that they are able to reduce the curse of dimensionality impact on the classification process. Second, we address the

choice of a set of appropriate representatives using the matching pursuit algorithm, which seems to be a robust and effective approach.

4.3.1 State of the Art

Similarity-based classification

The literature on similarity-based classification is not vast. Jain and Zongker [110] have obtained a dissimilarity measure, for a handwritten digit recognition problem, based on deformable templates; a multidimensional scaling approach was then used to project this dissimilarity space onto a low-dimensional space, where a 1-nearest-neighbor (1-NN) classifier was employed to classify new objects. In [87], Graepel *et al.* investigate the problem of learning a classifier based on data represented in terms of their pairwise proximities, using an approach based on Vapnik's structural risk minimization [215]. Jacobs and Weinshall [107] studied the use of distance-based classification with non metric distance functions (*i.e.*, that do not verify the triangle inequality). Duin and Pekalska are very active authors in this area,² having recently produced several papers [169, 173, 170]. Motivation and basic features of similarity-based methods were first described in [169]; it was shown, by experiments in two real applications, that a Bayesian classifier (the RLNC - Regularized Linear Normal density-based Classifier) in the dissimilarity space outperforms the nearest neighbor rule. These aspects were more thoroughly investigated in [170], where other classifiers in the dissimilarity space were studied, namely on digit recognition and bioinformatics problems. Finally, in [173], a generalized kernel approach was introduced, dealing with classification aspects of the dissimilarity kernels.

The dimensionality issue

The main problem of the similarity-based approach, of particular relevance in practical applications, is the high dimensionality of the resulting similarity space. In the literature, two types of solutions have been proposed in order to address this problem. The first consists in building the similarity space using all available patterns, and subsequently applying some standard dimensionality reduction techniques. One example of this kind of approach is the multidimensional scaling method used by Jain and Zongker [110]. Another recent example is presented by Pekalska and Duin in [172], where a reduction of the dimensionality of the dissimilarity space is obtained by a modified multidimensional scaling scheme, able to reduce the computational burden and to allow generalization to new data. The second type of solution to the dimensionality problem performs by directly choosing a small set of representatives. An example of this type of solution can be found in [170], where random selection, most-dissimilar rule, and the *Condensed Nearest Neighbor* (CNN) rule were employed. Other examples could be also found in [107], where a new type of CNN method is proposed, or more recently in [171], where a greedy approach is proposed, able to find prototypes encoding the principal components of the similarity space.

² See <http://www.ph.tn.tudelft.nl/Research/neural/index.html>

4.3.2 The Similarity-Based Strategy

Introduction

The basic issue of a similarity-based strategy is how to define similarities in an HMM framework. It has to be remembered that, given an HMM λ and a sequence \mathbf{O} , there is a standard procedure (*forward-backward procedure* [16]) to compute $P(\mathbf{O}|\lambda)$, *i.e.*, the probability (density) that the sequence \mathbf{O} has been generated by model λ . This quantity is called *likelihood*, and measures how well the sequence \mathbf{O} “fits” the model λ . A natural choice is then to define the similarity $D_{ij} = \mathcal{D}(\mathbf{O}_i, \mathbf{O}_j)$ between two sequences \mathbf{O}_i and \mathbf{O}_j as

$$D_{ij} = \mathcal{D}(\mathbf{O}_i, \mathbf{O}_j) = \frac{\log P(\mathbf{O}_i|\lambda_j)}{T_i} \quad (4.11)$$

where λ_j is the HMM trained on sequence \mathbf{O}_j , and T_i is the length of the sequence \mathbf{O}_i . The $1/T_i$ is a normalization coefficient introduced to take into account sequences of different length. Notice that this similarity is not symmetric.

The idea at the basis of the proposed approach is conceptually simple: building a new representation space, using the similarity values between sequences obtained via the HMMs according to (4.11), and constructing a classifier in that space. One of the justifications for this approach lies in the fact that similarity is high for similar objects/sequences, *i.e.*, belonging to the same class, and low for objects of different classes, making discrimination possible [173]. Therefore, we can interpret the similarity measure $\mathcal{D}(\mathbf{O}, \mathbf{O}_i)$ between a sequence \mathbf{O} and another “reference” sequence \mathbf{O}_i as a “feature” of the sequence \mathbf{O} . This fact suggests the construction of a feature vector for \mathbf{O} by taking the similarities between \mathbf{O} and a set of reference sequences $\mathcal{R} = \{\mathbf{O}_k\}$, so that \mathbf{O} is characterized by a *pattern* (*i.e.*, a set of features) $\{\mathcal{D}(\mathbf{O}, \mathbf{O}_k), \mathbf{O}_k \in \mathcal{R}\}$.

This approach is well suited for HMMs. With the classical approach, given a sequence \mathbf{O} , the rule defined by (4.5) uses HMMs to compute the similarities between \mathbf{O} and all the sequences in the training set. Subsequently, it seeks the most similar training sequence, and classifies \mathbf{O} as belonging to the class of this sequence (exactly as in a 1-NN classifier). Therefore, this process does not use all the information contained in the complete set of similarities. In our approach, instead, all this information is used. Notice that the fact that two sequences, say \mathbf{O}_i and \mathbf{O}_j , present similar degrees of similarity to several other sequences (*e.g.*, they are both very similar to some sequences, and also both very dissimilar to some other sequences) enforces the hypothesis that \mathbf{O}_i and \mathbf{O}_j belong to the same class.

Formal Definition

Formally, the proposed strategy is defined as follows. Consider a classification problem with C classes; for each class $k \in \{1, 2, \dots, C\}$, we have a set of N_k training sequences $\mathcal{T}_k = \{\mathbf{O}_1^{(k)} \dots \mathbf{O}_{N_k}^{(k)}\}$; thus $N = \sum_k N_k$ is the total size of the training set $\mathcal{T} = \bigcup_{k=1}^C \mathcal{T}_k$.

Let $\mathcal{R} = \{\mathbf{P}_1, \dots, \mathbf{P}_R\}$ be a set of R “reference” or “representative” objects; these objects may belong to the set of training sequences ($\mathcal{R} \subseteq \mathcal{T}$) or may be otherwise defined. Now let $\mathcal{D}_{\mathcal{R}}(\mathbf{O})$ be a function that returns the vector of similarities between an arbitrary sequence \mathbf{O} and all the sequences in \mathcal{R} , that is

$$\mathcal{D}_{\mathcal{R}}(\mathbf{O}) = \begin{bmatrix} \mathcal{D}(\mathbf{O}, \mathbf{P}_1) \\ \vdots \\ \mathcal{D}(\mathbf{O}, \mathbf{P}_R) \end{bmatrix} \in \mathbb{R}^R \quad (4.12)$$

We will designate the space \mathbb{R}^R in which the dissimilarity vector exists as the “similarity space” and denote it as $\mathcal{S}_{\mathcal{R}}$, where the subscript \mathcal{R} is used to emphasize the dependance of the similarity space on the set \mathcal{R} . Once this similarity space is defined, any standard classifier can, in principle, be used.

Regarding the choice of \mathcal{R} , different approaches can be adopted; the basic one, described in next subsection, is to choose $\mathcal{R} = \mathcal{T}$, the whole training set. With this choice, the dimensionality of $\mathcal{S}_{\mathcal{R}} = \mathcal{S}_{\mathcal{T}}$ is equal to N , the cardinality of the training set \mathcal{T} . Obviously, this represents a problem, because it makes the proposed method unapplicable in most cases; nevertheless, it is interesting to investigate the discrimination ability of this space.

Subsequently, the problem of reducing the dimensionality of the space is addressed by three different approaches: in the first one, linear projection techniques are applied to the whole similarity space $\mathcal{S}_{\mathcal{T}}$; in the second one, we will modify the strategy used to compute the distance $\mathcal{D}(\cdot, \cdot)$; in the third one, we finally use a greedy strategy, based on a *matching pursuit* algorithm, in order to choose a “good” set of representatives.

Basic Approach: $\mathcal{R} = \mathcal{T}$

When we take $\mathcal{R} = \mathcal{T}$, the dimensionality of $\mathcal{S}_{\mathcal{R}}$ is equal to N , the cardinality of \mathcal{T} . Notice that in this case we are required to design a classifier on a N -dimensional space using only N training sequences; this is an extreme case of the curse of dimensionality, suggesting that some dimensionality reduction technique should be adopted. Linear transformations, such as *Principal Component Analysis* (PCA) [113] (see appendix A.1) or *Fisher Discriminant Analysis* (FDA) [81] (see appendix A.3), were conceived as means of reducing the dimensionality of a space while preserving almost all the “relevant information” contained in a data set. The reduction of the space dimensionality absorbs some of the impact of the curse of dimensionality; moreover, it could sometimes eliminate some redundancy present in the data (as shown in the experiments), leading to a better classification performance.

Choice of the Set of Representatives \mathcal{R}

If we want to avoid the curse of dimensionality without having to resort to PCA or FDA, smarter ways of choosing \mathcal{R} have to be devised. Clearly, the choice of \mathcal{R} is critical since only if this set is adequately chosen, the discrimination power of the space $\mathcal{S}_{\mathcal{R}}$ will be large. Here, two methods are proposed, namely, the *One Per Class* (OPC), and the *Matching Pursuit* (MP) procedures.

The “One Per Class” Approach.

In this approach, which is similar to the ML_{OPC} scheme described in Section 4.1, instead of training one HMM for each sequence, a model is trained for each class using all sequences of that class. Using these HMMs, the feature vector of a sequence \mathbf{O} is a C -dimensional (for a C -class problem) vector given by

$$\mathcal{D}_{OPC}(\mathbf{O}) = \frac{1}{T} \begin{bmatrix} \log P(\mathbf{O}|\lambda_1) \\ \vdots \\ \log P(\mathbf{O}|\lambda_C) \end{bmatrix} \quad (4.13)$$

where λ_j is the HMM estimated from the set of all training sequences from class j , and T is the length of sequence \mathbf{O} . In this case, $\mathcal{D}_{OPC}(\mathbf{O})$ can be seen as containing the similarities between \mathbf{O} and each of the C classes. We can imagine the set \mathcal{R} as containing C sequences $\{\mathbf{P}_1, \dots, \mathbf{P}_C\}$, such that \mathbf{P}_j is an (imaginary) sequence such that if we applied the learning algorithm to \mathbf{P}_j we would still obtain λ_j . In the following, we will denote the similarity space obtained with this approach as \mathcal{S}_{OPC} .

The Matching Pursuit Approach.

The MP approach is based on the following idea: instead of using all sequences of the training set, one can choose those that are more “useful” in classification, *i.e.*, more discriminant in some sense. This choice is made incrementally, starting with an empty set, and adding at each step the object that yields the largest “performance improvement”. The process is stopped by some convergence criterion.

The MP algorithm was introduced in the signal processing community as an algorithm to decompose a signal into a linear combination of basis functions from a redundant dictionary [153]. It is a general, greedy, approximation scheme that works by sequentially appending functions to an initially empty set. At each step, the basis function appended is the one that produces the largest decrease in the approximation error. Recently, Vincent and Bengio [221] used MP to obtain kernel-based solutions to machine-learning problems.

Formally, the MP algorithm is defined as:

- Set $\mathcal{R} = \emptyset$ (the empty set);
- Until some stopping criterion is met, repeat:
 - For each sequence $\mathbf{O}_i^{(k)} \notin \mathcal{R}$, compute the *Leave One Out* (LOO) classification error rate of the 1-NN classifier using the feature vector $\mathcal{D}_{\{\mathcal{R} \cup \mathbf{O}_i^{(k)}\}}(\cdot)$.

Let’s denote this error as $E_{\mathcal{R}}(\mathbf{O}_i^{(k)})$.

- The new representative set is $\mathcal{R} = \mathcal{R} \cup \{\mathbf{O}_{i^*}^{(k^*)}\}$, where

$$(i^*, k^*) = \arg \min_{(i,k): \mathbf{O}_i^{(k)} \notin \mathcal{R}} E_{\mathcal{R}}(\mathbf{O}_i^{(k)}).$$

In the following, we denote the similarity space obtained with this approach as \mathcal{S}_{MP} . Note that, unlike the OPC approach, this scheme is very general, and can be used in all other instances of similarity-based classification.

4.3.3 Results and discussion

In this section, experimental results are reported, in order to validate the proposed approach. Firstly, we investigate the discriminative power of the space $\mathcal{S}_{\mathcal{R}}$ with $\mathcal{R} = \mathcal{T}$, *i.e.* using, as reference set, the whole training set \mathcal{T} . The standard ML classification scheme and the proposed approach are compared, with both synthetic and real data. The use of PCA and FDA is investigated in this context, also with the aim of visualizing the data. Secondly, experimental results concerning the two different choices of \mathcal{R} (OPC and MP) are reported. All the experiments are repeated 10 times and the results are averaged, so as to increase the independence of the results from the training of the HMMs.

Basic Approach: $\mathcal{R} = \mathcal{T}$

Synthetic Data.

We consider a 3-class synthetic problem, defined by the parameters given in Fig. 4.5. The training set is composed of 30 sequences (of length 400) from each of

$$\begin{aligned}
 \mathbf{A} &= \begin{array}{|c|c|c|} \hline 1/3 & 1/3 & 1/3 \\ \hline 1/3 & 1/3 & 1/3 \\ \hline 1/3 & 1/3 & 1/3 \\ \hline \end{array} & \boldsymbol{\pi} &= \begin{array}{|c|} \hline 1/3 \\ \hline 1/3 \\ \hline 1/3 \\ \hline \end{array} & \mathbf{B} &= \begin{array}{|c|c|} \hline \mu_1 = 1 & \sigma_1^2 = 0.6 \\ \hline \mu_2 = 3 & \sigma_2^2 = 0.6 \\ \hline \mu_3 = 5 & \sigma_3^2 = 0.6 \\ \hline \end{array} \\
 & & & & \text{(a)} \\
 \\
 \mathbf{A} &= \begin{array}{|c|c|c|} \hline 1/3 & 1/3 & 1/3 \\ \hline 1/3 & 1/3 & 1/3 \\ \hline 1/3 & 1/3 & 1/3 \\ \hline \end{array} & \boldsymbol{\pi} &= \begin{array}{|c|} \hline 1/3 \\ \hline 1/3 \\ \hline 1/3 \\ \hline \end{array} & \mathbf{B} &= \begin{array}{|c|c|} \hline \mu_1 = 1 & \sigma_1^2 = 0.5 \\ \hline \mu_2 = 3 & \sigma_2^2 = 0.5 \\ \hline \mu_3 = 5 & \sigma_3^2 = 0.5 \\ \hline \end{array} \\
 & & & & \text{(b)} \\
 \\
 \mathbf{A} &= \begin{array}{|c|c|c|} \hline 1/3 & 1/3 & 1/3 \\ \hline 1/3 & 1/3 & 1/3 \\ \hline 1/3 & 1/3 & 1/3 \\ \hline \end{array} & \boldsymbol{\pi} &= \begin{array}{|c|} \hline 1/3 \\ \hline 1/3 \\ \hline 1/3 \\ \hline \end{array} & \mathbf{B} &= \begin{array}{|c|c|} \hline \mu_1 = 1 & \sigma_1^2 = 0.4 \\ \hline \mu_2 = 3 & \sigma_2^2 = 0.4 \\ \hline \mu_3 = 5 & \sigma_3^2 = 0.4 \\ \hline \end{array} \\
 & & & & \text{(c)}
 \end{aligned}$$

Fig. 4.5. Generative HMMs for synthetic data testing: \mathbf{A} is the transition matrix, $\boldsymbol{\pi}$ is the initial state probability, and \mathbf{B} contains the parameters of the emission density (Gaussians with the indicated means and variances).

the three classes; the dimensionality of the similarity space $\mathcal{S}_{\mathcal{R}}$ is thus $N = 90$. Notice that this classification task is not easy, as the three HMMs are very similar to each other, only differing slightly in the variances of the emission densities.

We compare the standard ML classification criterion with a simple classifier in the similarity space $\mathcal{S}_{\mathcal{R}}$, the k -nearest-neighbor (k -NN), for $k = 1$ (1-NN) and $k = 3$ (3-NN), using Euclidean distance. This classical technique assigns a given object \mathbf{O} to the class having the largest number of representatives in the set of the k objects in the training set that are nearest to \mathbf{O} . This classifier is widely used,

as it is simple, fast, and reasonably accurate. The major drawback of nearest-neighbor classifiers is their sensitivity to noisy patterns on the training set, and the need to store all the training samples.

Accuracies were computed using the Leave One Out (LOO) procedure. This means that the dissimilarity space $\mathcal{S}_{\mathcal{R}}$ is actually built by using the representatives set \mathcal{R} consisting of 89 sequences, while one sequence is left out and used for testing. The procedure is repeated until all sequences have been tested (*i.e.* 90 times), and results are averaged. Results of different classifiers are shown in Table 4.1. We

Table 4.1. Classification accuracies using basic approach on synthetic data.

Classifier	Accuracy
ML_{OPS}	95.67%
1-NN on $\mathcal{S}_{\mathcal{T}}$	98.89%
3-NN on $\mathcal{S}_{\mathcal{T}}$	98.89%

can observe that there is an improvement when using the simple classifier in the similarity space. It is worth recalling that, as mentioned above, the three classes are very similar and the classification task is very difficult.

In order to get a better insight into the structure of our similarity space, we have applied PCA and FDA to the space $\mathcal{S}_{\mathcal{T}}$. Plots of the 2D projections of the training set thus obtained are shown in Fig. 4.6. It is clear that FDA is really effective in separating the classes, and even PCA leads to a satisfactory result, even if it ignores the class labels. In both cases, the three classes in the training set would be easily separable, although generalization would clearly be better with the FDA projection.

Classification accuracies were also obtained in these reduced spaces, in order to investigate discrimination ability of the similarity space. In this case, we use 1-NN and the Mahalanobis classifier (MC), which classifies an unknown observation as belonging to the class whose mean is nearest, using a Mahalanobis distance [65]. Accuracies (again computed with the LOO procedure) are presented in Table 4.2. For FDA, the maximum dimensionality allowed is $C - 1$, where C is the number of

Table 4.2. LOO accuracies on synthetic data, projected using PCA and FDA.

	Dimensionality			
	2	3	4	5
1-NN on PCA space	98.89%	98.89%	98.89%	98.89%
MC on PCA space	98.89%	97.78%	97.78%	96.67%
1-NN on FDA space	100%	-	-	-
MC on FDA space	100%	-	-	-

classes [81]. In this case, therefore, the maximum dimensionality is two. Comparing Table 4.2 with Table 4.1(a) we can note that the performances on the FDA reduced

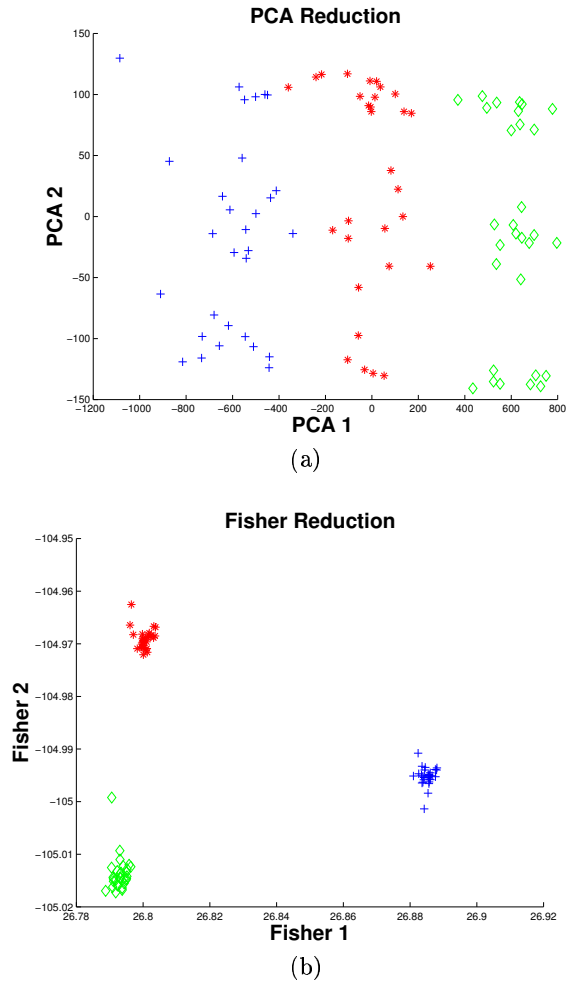


Fig. 4.6. 2D projections of the synthetic training set using (a) PCA and (b) FDA.

space is increased, reaching a perfect classification rate (which is not surprising in view of Fig. 4.6 (b)).

Real Data.

The proposed approach has been tested on two real applications: a 2D shape recognition task, described in Chapter 6, and a face recognition problem, using HMMs as proposed in [125] (briefly resumed in Chapter 7). For the former application, the shape contours are represented by their curvature, modelled using continuous HMMs. Differently from the approach proposed in Chapter 6, we do not use here any model selection technique. Testing was performed on part of the object set used in [197], composed by seven classes, each containing 12 different shapes.

As before, accuracies are computed using the LOO scheme. The database used is shown in Fig. 4.7.

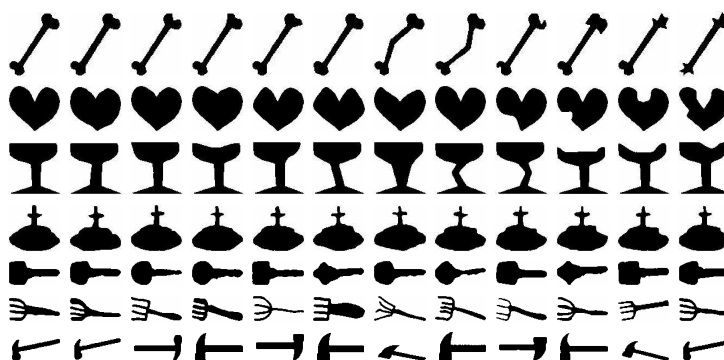


Fig. 4.7. Objects set used for testing.

For the face recognition task, HMMs were used as proposed in [125], described in Chapter 7, considering DCT coefficients as features. Given a sequence of sub-images of the face image, the DCT coefficients of each sub-image are computed, and vectorized using a *zig-zag* scan. The number of coefficients chosen determines the dimensionality of the observation, and 10 coefficients are used in our experiment. The sequence of sub-images is obtained by sliding over the face image a square fixed size window, in a raster scan fashion, with a predefined overlap. The window size and the overlap ratio were fixed to 8% and 50% respectively. Testing was performed using the Bern face database³, which consists of 30 subjects with 10 face images each. For each subject, five faces were used for training and the others for testing. We have chosen to use this database, instead of the ORL used in [125], because with that database HMMs are able to reach an almost perfect classification, so without any possibility of improvement.

Also in this case, the classical ML classification criterion was compared with the similarity-based approach, using a k -NN rule (for $k = 1$ and $k = 3$) in the similarity space $\mathcal{S}_{\mathcal{T}}$. Accuracies are presented in Table 4.3(a) and (b), for 2D shape recognition and for face classification tasks, respectively.

In the 2D shape case, the improvement in classification rates is even larger, of about 18% for the 1-NN classifier and of about 13% for the 3-NN. This shows that this similarity-based feature space is very well suited for this real case; the explanation could be the following. Looking at Fig. 4.7, we can note that there are many differences among items in the same class; the use of all similarities between items may add a lot of discriminative power to the method. This additional discriminative power increases more when the differences among items of same class are large. Also in the face recognition case there is a noticeable improvement in the accuracies of classification, confirming the wide applicability of this method to real cases.

³ Downloaded from <ftp://iamftp.unibe.ch/pub/Images/FaceImages>

Table 4.3. Classification accuracies using basic approach on real data: (a) 2D shape recognition, and (b) face recognition.

Classifier	Accuracy	Classifier	Accuracy
ML_{OPS}	80.95%	ML_{OPS}	50.60%
1-NN on $\mathcal{S}_{\mathcal{T}}$	98.81%	1-NN on $\mathcal{S}_{\mathcal{T}}$	72.07%
3-NN on $\mathcal{S}_{\mathcal{T}}$	93.21%	3-NN on $\mathcal{S}_{\mathcal{T}}$	60.53%

(a) (b)

FDA and PCA were also studied in the case of the 2D shape recognition experiments. Plots of projected training set are shown in Fig. 4.8. As in the previous

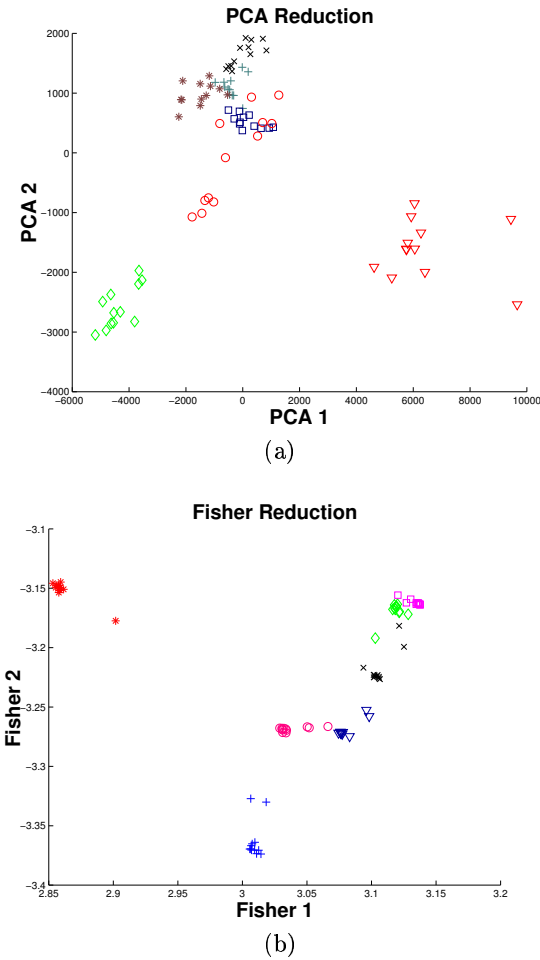


Fig. 4.8. Data set for 2D shape recognition experiment, reduced and plotted using: (a) Principal Component Analysis; (b) Fisher Discriminant Analysis.

subsection, classification accuracies were calculated for different dimensionalities, using the LOO procedure, and the results are reported in Table 4.4. In this case, the

Table 4.4. LOO accuracies on 2D shape recognition task, after PCA and FDA projections.

	Dimensionality			
	2	3	4	5
1-NN on PCA space	80.12%	97.74%	98.21%	98.33%
MC on PCA space	81.19%	92.86%	91.67%	93.33%
1-NN on FDA space	92.5%	95.05%	96.31%	97.14%
MC on FDA space	86.55%	92.62%	90.12%	91.31%

reduction of dimensionality to 2 decreases the classification performance, which, in any case, is still better than the results obtained using the standard ML criterion. The similarity feature space is complex in this case, due to the presence of very dissimilar elements in the same class. For low dimensionality, part of this information is lost, but, by slightly increasing the dimensionality, this information is correctly recovered, and the performance returns to a very good level.

To investigate the robustness of the approach, we have also tested the behavior of the method in the presence of shape occlusions. Occlusion is one of the most severe limitations to the application of typical object recognition techniques. As proposed in Chapter 6, HMMs are very effective in dealing with object occlusions. Here we show that the approach proposed in that chapter can be further improved by using the similarity space representation.

Object occlusion is simulated by considering a fragment of the object boundary, starting at a randomly chosen location. Each object was occluded 5 times, resulting in 420 sequences. Occlusion percentages considered were 10%, 30% and 50%; notice that in the last case, one half of the whole boundary is missing. Also in this case a LOO scheme was adopted: note that this results in a really complex task, as the left out sequence (the occluded one) was not used for building the similarity space. This choice makes all experiments uniform throughout the section, even if it can be seen as somewhat strange, since typically to recognize an occluded object, also the original shape is available (this obviously results in a great improvement in the performances, see Chapter 6).

Results for the different occlusion levels considered, using 1-NN and 3-NN classifiers, are shown in Table 4.5. We observe a clear improvement in the classification accuracies of the classifiers in the similarity space.

Choice of representatives set \mathcal{R}

In this section, the two approaches for the choice of \mathcal{R} described in Section 4.3.2 are tested. These approaches were applied to the 2D shape recognition (using both the entire and occluded shapes) and to the face classification experiments. Classification accuracies were calculated as in the previous section. We used 1-NN classifiers in the similarity spaces \mathcal{S}_{OPC} and \mathcal{S}_{MP} .

Table 4.5. Classification accuracies on data set formed by occluded shapes, at different occlusion levels.

	Occlusion level		
	10%	30%	50%
ML_{OPS}	76.90%	71.50%	60.95%
NN on $\mathcal{S}_{\mathcal{T}}$	91.50%	76.43%	64.19%
3-NN on $\mathcal{S}_{\mathcal{T}}$	91.90%	73.05%	64.10%

The comparison between the proposed approaches and ML classification is reported in Tables 4.6(a) and (b), for the entire and occluded shapes, respectively, and in Table 4.7, for the face experiment. For the sake of clarity, results for 1-NN on $\mathcal{S}_{\mathcal{T}}$ (entire similarity space) are also shown, in order to quantify the loss in classification accuracy determined by the reduction. Moreover, the dimension of the resulting similarity space \mathcal{S} is included in the tables, in order to emphasize the amount of the reduction obtained. In summary, we can conclude that both ap-

Table 4.6. Accuracies on 2D shape experiments of the OPC and MP approaches for the choice of the representatives set \mathcal{R} , in two different experimental conditions: (a) entire shapes; (b) occluded shapes, for different occlusion levels (O.L.).

Classifier	Accuracy	Dim. of \mathcal{S}
ML_{OPC}	89.29%	-
ML_{OPS}	80.95%	-
1-NN on $\mathcal{S}_{\mathcal{T}}$	98.81%	84
1-NN on \mathcal{S}_{OPC}	97.38%	7
1-NN on \mathcal{S}_{MP}	92.86%	4.06

(a)

Classifier	O.L. = 10%	O.L. = 30%	O.L. = 50%	Dim. of \mathcal{S}
ML_{OPS}	76.90%	71.50%	60.95%	-
ML_{OPC}	83.05%	77.98%	69.19%	-
1-NN on $\mathcal{S}_{\mathcal{T}}$	91.50%	76.43%	64.19%	84
1-NN on \mathcal{S}_{OPC}	86.10%	71.11%	57.81%	7
1-NN on \mathcal{S}_{MP}	85.90%	72.10%	56.14%	4.26

(b)

proaches seem to be able to preserve most of the performance of the basic approach (classification on the whole similarity space $\mathcal{S}_{\mathcal{T}}$), while achieving a drastic dimensionality reduction. Regarding the 2D shape recognition experiment, by comparing the performance of the ML_{OPC} method, with the standard ML_{OPS} criterion, we can notice that the use of all sequences to learn each HMM enhances the accuracy of the standard ML classification. HMM is really suitable to be trained using many sequences, as it is able to deal with their possible different lengths. Nevertheless,

Table 4.7. Accuracies on face recognition experiments of the OPC and MP approaches for the choice of the representatives set \mathcal{R} .

Classifier	Accuracy	Dim. of \mathcal{S}
ML_{OPC}	51.67%	-
ML_{OPS}	50.60%	-
1-NN on $\mathcal{S}_{\mathcal{T}}$	72.07%	150
1-NN on \mathcal{S}_{OPC}	69.40%	30
1-NN on \mathcal{S}_{MP}	68.87%	10.1

this could reduce the expressivity of the resulting similarity space, especially in some real cases, where items of the same class present remarkable differences between each other. From Table 4.6(b) we can also notice that when increasing too much the occlusion level, the performances on reduced similarity spaces (MP and OPC approaches) are lower than standard ML classification level. This is probably due to the fact that when the percentage of occlusion increases, the HMMs are less accurately estimated. The obtained similarity space is thus noisy, and the 1-NN rule (which is the simplest classifier) is not able to perform well in such a noisy space. To verify this explanation, we recomputed the LOO classification accuracies on the experiment with the occluded shapes, with occlusion level 50%. We used a carefully trained multi layer feed forward neural network on the MP reduced similarity space: 1-NN accuracies were about 56% in that reduced space. Accuracies obtained with the neural network is around 88%, confirming the large potentialities of this approach: the mapping onto the similarity space allows to reduce complex sequence classification into easier standard point classification, for which one could use arbitrarily sophisticated techniques.

In conclusion, the two approaches for the choice of representatives set \mathcal{R} are both effective. OPC seems to be more interesting, as it results directly from the standard HMM training, without any need to postprocess the space. Nevertheless, the resulting dimensionality is equal to the number of classes, reducing the usefulness of the approach in problems with many classes (*e.g.*, face recognition). Moreover, the training of one HMM for each class can drastically reduce the discrimination ability of the similarity space when items of the same class are very different. On the other hand, the MP approach seems to be better in identifying the representatives that are *really* useful for the similarity-based classification purpose. The higher computational burden introduced with this approach is its major drawback.

4.3.4 Conclusions

In this section we have proposed a novel sequence classification scheme by combining Hidden Markov Models with the similarity-based paradigm. This approach creates a representation space for sequences in which standard feature-based classification techniques can be used. In the investigated applications, we showed that a simple classifier in a such space outperforms standard HMM-based classification schemes. Three approaches to deal with the high dimensionality of resulting space

were also considered and investigated, showing that the similarity-based representation seems to be still effective when its dimensionality is reduced in order to make it more manageable.

Future directions consist in applying and investigating more *ad hoc* similarity space classifiers, as those proposed in [173, 170], and in studying novel techniques for reducing space dimensionality.

Clustering with HMM

5.1 Introduction

Unsupervised classification (or clustering) of data [108, 109] is undoubtedly an interesting and challenging research area: it could be defined as the organization of a collection of patterns into groups, based on similarity. It is well known that data clustering is inherently a more difficult task if compared to supervised classification, in which classes are already identified, so that a system can be adequately trained. Clustering has been applied in several contexts, as, for example, data mining, DNA modelling, information retrieval, image segmentation, signal compression and coding, and machine learning. Hundreds of clustering algorithms have been proposed in the literature, mostly divided in two categories: iterative partitional techniques and agglomerative hierarchical techniques. The former class of algorithm attempts to obtain that partition which minimizes the within-cluster scatter or the between-cluster matrix. The latter class organizes data in a nested sequence of groups which can be displayed in a form of a dendrogram or a tree. This tree is then cut at the chosen depth level in order to obtain the desired clustering.

5.1.1 Clustering algorithms

This subsection is aimed at briefly introducing some details about these basilar approaches to clustering, since some of them will be employed through the rest of this chapter.

As claimed in the previous section, the most part of the clustering algorithms proposed in the literature could be divided in two categories [108, 109]: agglomerative hierarchical and iterative partitional techniques. The former class produces a sequence of clustering with a decreasing number of clusters. The clustering produced at each step typically results from the previous one by merging the two most similar clusters into one.

Most hierarchical clustering algorithms are variants of the Single Link (SL) [202] and the Complete Link (CL) [124] algorithms. These two algorithms differs in the way they characterize the similarity between a pair of clusters. In the Single Link method, the distance between two clusters is the *minimum* of the distances

between all pairs of patterns drawn from the two clusters. In the Complete Link algorithm, instead, the distance between two clusters is the *maximum* of all pairwise distances between patterns in the two clusters. Both these approaches have some advantages and some disadvantages: in particular, Complete Link is able to produce compact clusters, while Single Link has the tendency to find elongated clusters. On the other hand, the Single Link algorithm is more versatile, as it could extract also concentric clusters, while the Complete Link cannot. It should be noted, nevertheless, that Complete Link algorithm is able to produce more useful dendrograms, with respect to several application contexts, as shown in [108].

Another interesting variant of these methods is the the minimum-variance (Ward) [224,158] scheme. In this case, the distance $d(i, j)$ between two clusters C_i and C_j is defined as

$$d(i, j) = \frac{n_i n_j}{n_i + n_j} \| \mathbf{m}_i - \mathbf{m}_j \|^2 \quad (5.1)$$

where n_ℓ and \mathbf{m}_ℓ are the cardinality and the centroid of the cluster C_ℓ , respectively. It has been shown in [210] that this approach merges the two clusters that lead to the smallest possible increase in the total variance.

Another typology of clustering algorithms, which is in contrast with the hierarchical clustering class, is represented by the partitional clustering family: these methods produce a single partition of the data instead of a clustering structure, such as a dendrogram produced by hierarchical technique. The partitional algorithms usually perform clustering by optimizing a criterion function that could be defined locally or globally. Starting from an initial cluster assignment, the algorithm optimizes this function by iteratively re-assigning patterns to clusters, until a stable situation has been achieved. A typical problem of these approaches is the sensitivity to the initial clustering. The most used solution is to run the algorithm several times, and to choose the best obtained clustering configuration. An example of such techniques is the well known k-means method [108,12], a fast and effective clustering approach. This approach finds the optimal partition by evaluating, at each iteration, the distance between each item and each cluster descriptor, and by assigning it to the nearest class. At each step, the descriptor of each cluster is re-evaluated by averaging its cluster items. The system stops when no changes are present in the clustering. A simple variation of the method, called partition around medoid (PAM) [122], determines each cluster representative by choosing the point nearest to the centroid.

A final consideration can be formulated about these two methodologies: partitional methods have the advantage of being very fast, as not the whole dendrogram tree needs to be deduced: this could be a crucial factor in those applications involving large data sets, for which the construction of a dendrogram is computationally prohibitive. On the other hand, hierarchical algorithms are more versatile, as able to more properly perform also in cases of non-isotropic clusters.

5.1.2 Sequential data clustering

The intrinsic difficulty of the unsupervised classification of patterns with respect to supervised classification worsens if sequential data are considered: the structure

of the underlying process is often difficult to infer, and typically different length sequences have to be dealt with. Clustering of sequences has assumed an increasing importance in recent years, due to its wide applicability in emergent contexts like data mining and DNA genome modelling and analysis.

Sequence data clustering methods could be generally classified into three categories: *proximity-based* methods, *feature-based* methods and *model-based* methods. In the *proximity-based* approaches, the main effort of the clustering process is in devising similarity or distance measures between sequences. Once determined such measures, any standard distance-based method (as agglomerative) could be applied. Examples of such methods include time series correlation measures, string distance metrics (as the Hamming distance [108] or string edit distance [186]) and dynamic time warping method [163]. *Feature-based* methods extract a set of features from individual data that capture the temporal information. By way of this, the problem of sequence clustering is reduced to a more addressable static point (vector of features) clustering. Standard examples of these methods use Fourier descriptors [4] and wavelet coefficients [98]. Finally, *model-based* approaches assume an analytical model for each cluster, and the aim of clustering is to find a set of such models that best fit the data. Examples of models that could be employed include time series models, spectral models and finite state automata (as Hidden Markov Model).

HMM-based clustering of sequences

Related to sequence clustering, HMMs have not been extensively used, and only a few papers are present in the literature. The proposed approaches mainly falls into the first (proximity-based) and in the third (model-based) category. More specifically, early works, related to speech recognition, were proposed in [181,137,127]. All these methods belong to the first category, the proximity-based clustering class. HMMs were employed to compute similarity between sequences, and standard pairwise distance matrix-based approaches were then used to obtain clustering.

The first interesting approach, not directly linked to speech issues, was presented by Smyth [201] (see also the more general and more recent [37]). This approach consists in two steps. First, it devises a pairwise distance between observation sequences, by computing a symmetrized similarity. This similarity is obtained by training an HMM for each sequence, so that the log-likelihood (LL) of each model, given each sequence, can be computed. This information is used to build a LL distance matrix to be used to cluster the sequences in K groups, using a hierarchical algorithm. In the second step, one HMM is trained for each cluster; the resulting K HMMs are then merged in a “composite” global HMM, where each HMM is used to design a disjoint part of this “composite” model. This initial estimate is then refined using standard Baum Welch procedure. As a result, a global HMM modelling all the data is obtained. The number of clusters is calculated using a cross validation technique. With respect to the aforesaid taxonomy, this approach could be classified as belonging to both the proximity-based class (a pairwise distance is derived to initialize the model) and the model-based class (a model for clustering data is finally obtained).

An example of a model-based method for sequence clustering with HMMs is proposed in [134], where these models are used as cluster prototypes. The cluster-

ing is obtained by employing the Rival Penalized Competitive Learning (RPCL) approach [227], originally developed for point clustering, together with a state merging strategy, aimed at finding smaller HMMs.

A relevant contribute to the model-based HMM clustering methodology has been provided by Li and Biswas [139, 140, 141, 142, 143]). Basically, in their approach (resumed in the Li PhD. thesis [139]), the clustering problem is addressed by focusing on the model selection issue, *i.e.* the search of the HMM topology best representing data, and the clustering structure issue, *i.e.* finding the most likely number of clusters. In [140], the former issue is addressed using the Bayesian Information Criterion [196], and extending to the continuous case the Bayesian Model Merging approach [205]. Regarding the latter issue, the sequence-to-HMM likelihood measure is used to enforce the within-group similarity criterion. The optimal number of clusters is then determined maximizing the Partition Mutual Information (PMI), which is a measure of the inter-cluster distances. In [141], the same problems are addressed in terms of Bayesian model selection, using the Bayesian Information Criterion (BIC) [196], and the Cheesman-Stutz (CS) approximation [45]. A more comprehensive version of this paper has appeared in [143], where the method is also tested on real world ecological data. These clustering methodologies have been applied to specific domains, as physiology, ecology and social science, where the dynamic model structure is not readily available. Obtained results have been published in [142].

5.1.3 Chapter outline

In this chapter the problem of clustering of sequences using HMMs is analyzed, and some contributions are presented. More in detail, in Section 5.2, after reviewing the standard proximity-based algorithm used for sequence clustering with HMMs, a new metric to measure the distance between sequences is proposed. A new partitional method, variation of the Partition Around Medoid (PAM) [122] strategy, is proposed in Section 5.2.2, able to perform pairwise distance-based clustering in a partitional manner. All these schemes are then evaluated using real data sequences, *i.e.* the electroencephalographic (EEG) signals. Analysis of this kind of signals has become very important in the last years, due to the growing interest in the field of *Brain Computer Interface* (BCI) [175]. In this case, autoregressive HMMs were employed, with particular care to the HMM training initialization, with the use of a Kalman filtering and a mixture of Gaussians clustering method.

Subsequently, in Section 5.3, an alternative HMM clustering scheme is proposed, based on the similarity representation introduced in Section 4.3. Also in this case, we propose to build a new feature space, where each sequence is characterized by its similarity to all other sequences, similarities being computed with HMMs. In that space, clustering is then performed using some standard point techniques: the difficult task of sequence clustering is thus recovered to a more manageable clustering of points (vectors of features). With respect to the taxonomy before introduced, this method could be classified as a feature-based method, and represents an innovative approach in the HMM literature. Experimental evaluation on synthetic and real problems shows that this alternative approach largely outperforms standard HMM clustering schemes in the addressed problems.

5.2 Standard HMM-based clustering approach

The standard proximity-based method for clustering of sequences using HMMs could be depicted by the following algorithm: given a set of N sequences $\{\mathbf{O}_1 \dots \mathbf{O}_N\}$ to be clustered:

1. Train one HMM λ_i for each sequence \mathbf{O}_i .
2. Compute the distance matrix $D = \{D(\mathbf{O}_i, \mathbf{O}_j)\}$, representing a similarity measure between sequences or between models; this is typically obtained from the forward probability $P(\mathbf{O}_j|\lambda_i)$, or by devising a measure of distances between models.
3. Use a pairwise distance matrix-based method to obtain the clustering, as agglomerative methods.

5.2.1 Distance between sequences using HMM

In the second step the HMMs computed in the the first step are used to determine distances between sequences. In the past few authors have proposed approaches to computing these distances: early approaches were based on the Euclidean distance of the discrete observation probability [138], others on entropy [114, 71], or on co-emission probability of two models [150], or, very recently, on the Bayes probability of error [9].

In this section, the appropriateness of three kinds of measures is investigated. These measures are all based on the likelihood matrix L_{ij} , defined on the basis of the HMMs $\{\lambda_i\}$ trained on the sequences $\{\mathbf{O}_i\}$:

$$L_{ij} = P(\mathbf{O}_j|\lambda_i), \quad 1 \leq i, j \leq N \quad (5.2)$$

The first measure, denoted as L_S and proposed in [201], is obtained by merely symmetrizing the L matrix:

$$L_S^{ij} = \frac{1}{2} [L_{ij} + L_{ji}] \quad (5.3)$$

The second measure, which reminds the Kullback-Leibler information number [131, 130], defines the distance L_{KL} between two sequences \mathbf{O}_i and \mathbf{O}_j , and its symmetrized version L_{KLS} , as

$$L_{KL}^{ij} = L_{ii} \left[\ln \frac{L_{ii}}{L_{ji}} \right] + L_{ij} \left[\ln \frac{L_{ij}}{L_{jj}} \right] \quad (5.4)$$

$$L_{KLS}^{ij} = \frac{1}{2} [L_{KL}^{ij} + L_{KL}^{ji}] \quad (5.5)$$

The Kullback-Leibler information number could be an useful quantity in this context, since it represents a measure of distance between probability distribution functions.

Finally, we introduced another measure, called BP metric, defined as

$$L_{BP}^{ij} = \frac{1}{2} \left\{ \frac{L_{ij} - L_{ii}}{L_{ii}} + \frac{L_{ji} - L_{jj}}{L_{jj}} \right\} \quad (5.6)$$

motivated by the following considerations: the measure (5.3) defines the similarity between two sequences \mathbf{O}_i and \mathbf{O}_j as the likelihood of the sequence \mathbf{O}_i with respect to the model λ_j (trained on \mathbf{O}_j) plus the likelihood of the sequence \mathbf{O}_j with respect to the model λ_i (trained on \mathbf{O}_i); it does not really take into account the effectiveness of the HMM learning. In other words this kind of measure assumes that all sequences are modelled with the same quality, without considering how well each sequence is modelled by the HMM: this could not always be true. Our proposed distance also considers the modelling goodness by evaluating the relative normalized difference between the sequence and the training likelihoods.

5.2.2 Pairwise distance-based clustering algorithms

In the third step, a pairwise clustering technique is needed. The main characteristic of this kind of methods is that they determine the clustering on the basis of a pairwise distance matrix, containing the dissimilarity between each pair of patterns in the data set. The natural choice, in this case, is to use hierarchical algorithms, already detailed in Section 5.1.1. The disadvantage of this kind of algorithms is that the whole dendrogram tree needs to be deduced, and this could be a crucial factor in those applications involving large data sets, for which the construction of a dendrogram is computationally prohibitive.

In such cases, partitional algorithms should be preferred. In our distance matrix-based context, nevertheless, the partitional algorithms cannot be applied, since typically they are not able to deal with only distance matrices. Recently, a partitional algorithm called “Clustering by friends” has been proposed by Dubnov *et al.* in [64], able to obtain a partition of the data from only the distance matrix. This algorithm, which is nonparametric, iteratively employs a two steps transformation on the proximity matrix. The first step of the transformation represents each point by its relation to all other data points, and the second step re-estimates the pairwise distances using a proximity measure on these representations. Using these transformations, the algorithm iteratively partitions the data points, until it finally converges to two clusters.

In this section we propose a new partitional method, called “DPAM” (Distance matrix Partition Around Medoid), able to directly deal with the pairwise distance matrix. The proposed approach shares the ideas of the PAM technique, that could not be used in this context. In fact, it is not possible to evaluate the centroid of each cluster when having distances between items only, and not their values. The proposed method is able to determine cluster descriptors in a PAM paradigm, using items distances instead of their values. The basic idea is to fix as a representative of the cluster the more “central” element, *i.e.* the element which has the minimum distance to all other elements of the cluster. Moreover, the choice of the initial descriptors could affect algorithm performances. To overcome this problem we have adopted a multiple initialization procedure, where the best resulting partition is determined by a sort of Davies-Bouldin criterion [56].

The DPAM Algorithm

Fixed η as the number of tested initializations, N the number of sequences $\{\mathbf{O}_1 \dots \mathbf{O}_N\}$, k the number of clusters (supposed known) and $D(\mathbf{O}_i, \mathbf{O}_j)$ the proximity matrix, the resulting algorithm is the following:

- for $t=1$ to η
 - Initial cluster representatives θ_j are randomly chosen ($j = 1, \dots, k$, $\theta_j \in \{\mathbf{O}_1, \dots, \mathbf{O}_N\}$).
 - Repeat:
 - *Partition evaluation step:*
Compute the cluster to which each sequence \mathbf{O}_i belongs; \mathbf{O}_i lies in the j cluster for which the distance $D(\mathbf{O}_i, \theta_j)$ is minimum.
 - *Parameters upgrade:*
 - Compute the sum of the distance of each element of cluster C_j^t from each other element of the j th cluster
 - Determine the element in C_j^t for which this sum is minimal
 - Use that element as new descriptor for cluster C_j^t
 - Until the representatives θ_j values between two successive iterations do not change.
 - $\mathcal{R}_t = \{C_1^t, C_2^t, \dots, C_k^t\}$
 - Compute the Davies–Bouldin-like index defined as:

$$DB\mathcal{L}^{(t)} = \frac{1}{k} \sum_{r=1}^k \max_{s \neq r} \left\{ \frac{S_c(C_r^t, \theta_r) + S_c(C_s^t, \theta_s)}{D(\theta_r, \theta_s)} \right\}$$

where S_c is an intra-cluster measure defined by:

$$S_c(C_r, \theta_r) = \frac{\sum_{\mathbf{O}_i \in C_r^t} D(\mathbf{O}_i, \theta_r)}{|C_r^t|}$$

- endfor t
- *Final solution:* The best clustering \mathcal{R}_p has the minimum Davies–Bouldin index, *i.e.*

$$p = \arg \min_{t=1, \dots, \eta} \{DB\mathcal{L}^{(t)}\}$$

5.2.3 Application to the EEG modelling

The proposed measures and the proposed algorithm were then tested on a real complex problem, concerning the modelling of Electroencephalographic (EEG) signals. These signals represent the brain activity of a subject and give an objective mode of recording brain stimulation. EEGs are useful tools used for understanding several aspects of the brain, from diseases detection to sleep analysis and evoked potential analysis. The system used to model EEG signal is largely based on Penny and Roberts paper [174]: the key idea under this approach is to train an autoregressive HMM (already described in Section 2.2.1) directly on the EEG signal, rather than use an intermediate AR representation. Each HMM state can

be associated with a different dynamic regime of the signal, determined using a Kalman Filter approach [119]. Kalman filter is used to preliminary segment the signal into different dynamic regimes: these estimates are then fine-tuned with the HMM. Great attention was paid to the initialization of the HMM training procedure: first, a Kalman filter AR model is passed over the data, obtaining a sequence of AR coefficients; then coefficients corresponding to low evidence are discarded and the remaining are clustered with Gaussian Mixture Models [154]. Finally, the center of each Gaussian cluster is used to initialize the AR coefficients in each state of the HMM-AR model. To initialize the transition matrix we used prior knowledge from the problem domain about average state duration densities. We use the equation $a_{ii} = 1 - \frac{1}{d}$ to let HMM remain in state i for d samples. This number is computed knowing that EEG data is stationary for a period of the order of half a second [162].

The number of clusters (*i.e.* the number of HMM states) and the order of the autoregressive model were decided by performing a preliminary analysis of classification accuracy. By varying the number of states from 4 to 10, and by varying the order of autoregressive model from 4 to 8, we have found that best configuration was $k = 4$ and $p = 6$.

5.2.4 Experiments

In order to validate the exposed modelling technique we worked primarily on EEG data recorded by Zak Keirn at Purdue University [123]. The dataset contains EEG signals recorded from different subjects which were asked to perform five mental tasks: a *baseline task*, for which the subjects were asked to relax as much as possible; the *math task*, for which the subjects were given nontrivial multiplications problems, such as $27*36$, and were asked to solve them without vocalizing or making any other physical movements; the *letter task*, for which the subjects were instructed to mentally compose a letter to a friend without vocalizing; the *geometric figure rotation*, for which the subjects were asked to visualize a particular 3D block figure being rotated about an axis; and a *visual counting task*, for which the subjects were asked to image a blackboard and to visualize numbers being written on the board sequentially. We applied the method on a segment-by-segment basis, 1s signals sampled at 250Hz and drawn from a dataset of cardinality varying from 190 (two mental states) to 473 sequences (five mental states), where we removed segments biased by signal spikes derived from human artifacts (e.g. ocular blinks).

First of all, some classification analysis were performed, in order to choose the best parameter configuration. Classification accuracies were computed for four different subjects in the database, using the baseline and the math task (we choose four subjects in order to compare our approach with the literature). Obtained results, compared with those derived using a Neural Network [6] varying the number of hidden units, are proposed in Table 5.1: the averaged classification accuracy obtained with HMM is about 2% superior than that obtained using Neural Networks, showing that Hidden Markov Models are more effective on this data set.

Subsequently, the proposed HMM clustering algorithm has been tested, by evaluating signals obtained from subject two. Experiments were performed with number of clusters varying from 2 to 5. For each number of clusters, all combinations were experimented, and only best results are displayed. Accuracies were

Table 5.1. Classification accuracies in EEG modelling using Autoregressive HMM and Neural Networks as in [6] (for different number of hidden units), for different subjects.

	Hidden units in NN			Autoregressive HMM
	1	2	5	
Subject 1	93.4% \pm 1.5	93.6% \pm 1.5	93.6% \pm 1.5	95.8% \pm 1.4
Subject 2	96.7% \pm 1.7	96.7% \pm 1.6	96.1% \pm 1.7	97.4% \pm 1.3
Subject 3	80.9% \pm 2.5	80.7% \pm 2.6	82.5% \pm 2.6	83.7% \pm 2.8
Subject 4	91.0% \pm 2.0	91.0% \pm 2.0	90.0% \pm 2.1	94.0% \pm 1.8
mean	90.5% \pm 1.9	90.5% \pm 1.9	90.5% \pm 2.0	92.7% \pm 2.0

computed by comparing the clustering results with real segment labels; percentages are merely the ratio of correct assigned label with respect to the total number of segments. First we applied the hierarchical complete link technique, varying the proximity measure: results are shown in Table 5.2, with number of mental states growing from two to five. We note that accuracies are quite satisfactory. None of

Table 5.2. Results for the application of the Hierarchical Complete Link varying the distances: BP, defined in (5.6), KL in (5.5) and SM in (5.3).

	BP	KL	SM
2 natural clusters	97.37%	97.89%	97.37%
3 natural clusters	71.23%	79.30%	81.40%
4 natural clusters	62.63%	57.36%	65.81%
5 natural clusters	46.74%	54.10%	49.69%

the experimented method can be considered the best one; nevertheless, measures (5.3) and (5.5) seem to be more effective. In particular, measure (5.3) seems to be especially suited for dealing with few clusters, while measure (5.5) performs better when numerous clusters are present.

Subsequently we applied the partitional algorithm to the same set, setting the number of initializations η to 5 during all the experiments. Results are presented in Table 5.3: in this last case the BP distance is overall slightly better than the

Table 5.3. Results for the application of the DPAM algorithm varying the distances: BP, defined in (5.6), KL in (5.5) and SM in (5.3).

	BP	KL	SM
2 natural clusters	95.79%	96.32%	95.79%
3 natural clusters	75.44%	72.98%	65.61%
4 natural clusters	64.21%	62.04%	50.52%
5 natural clusters	57.04%	46.74%	44.80%

others experimented measures. A final comparison of partitional and agglomerative hierarchical algorithms underlines that there are no remarkable differences between the proposed approaches. Clearly, partitional approaches alleviates computational burden, and they should therefore be preferred when dealing with complex signals clustering (e.g. EEG).

The comparison of the clustering accuracies with the correspondent classification accuracies on the second subject, proposed in Table 5.1, showed that there is only a slight difference, while the classification results get better as expected. This fact strengthens the quality of the proposed method, since unsupervised classification is inherently a more difficult task than the supervised one.

5.3 Clustering with the similarity-based representation

As presented in the previous section, clustering of sequences using HMM is typically addressed in two steps: first, HMMs are used for obtaining pairwise distances between sequences, in a likelihood sense; secondly, a standard distance matrix-based method is applied to the resulting distances matrix to obtain the clustering.

In this section, a novel and alternative scheme is proposed; with respect to the taxonomy presented in the introduction, it could be classified as a feature-based method for clustering. This method is mainly based on the similarity space representation introduced in Section 4.3. As shown in [110, 87, 107, 173, 169, 170] and in Section 4.3, the similarity-based representation is a powerful tool for extracting features from data. In the similarity-based representation, each pattern is represented as a vector of distances from a predetermined set of patterns; it has been shown that this representation is really effective and discriminant. In the case of sequences the major advantage of this approach is that the problem of (supervised or unsupervised) classification of sequences is reduced to a more standard and addressable point (or vector) classification, for which several techniques have been proposed. The problem is to find a suitable metrics for measuring the distance between sequences, and, as shown above in this chapter, HMMs can represent a suitable tool for that target.

The main idea under the proposed approach is to map the sequences of the data set onto the HMM-based similarity space introduced in the previous chapter, and to perform some standard point clustering techniques in that space; by way of this, the difficult task of clustering of sequences is recovered to a more manageable clustering of points. Experimental evaluation on synthetic and real experiment will show that this approach largely outperforms standard clustering techniques.

Also in this case it is necessary to deal with the difficult problem of reducing the dimensionality of the resulting similarity space: this task is here even harder than in the classification case, since labels are not available and classification accuracy could not be used as a driving criterion. Two possible solutions to this problem have been proposed here: the Principal Component Analysis (PCA) [113] and the Independent Component Analysis (ICA) [99], both approaching the reduction of the dimensionality of the similarity space by the use of a linear data projection technique. These techniques are briefly resumed in the appendix A.

5.3.1 The proposed approach

Given a set of sequences $\mathcal{T} = \{\mathbf{O}^1 \dots \mathbf{O}^N\}$ to be clusterized, the proposed approach maps each sequence in the similarity space defined in Section 4.3, performing subsequently the clustering in that space. The approach can be briefly resumed as follows:

- let $\mathcal{R} = \{\mathbf{P}_1, \dots, \mathbf{P}_R\}$ be a set of R “reference” or “representative” objects; these objects may belong to the set of sequences ($\mathcal{R} \subseteq \mathcal{T}$) or may be otherwise defined. In a basic case it could be $\mathcal{R} = \mathcal{T}$.
- train one HMM λ_r for each sequence $\mathbf{P}_r \in \mathcal{R}$;
- represent each sequence \mathbf{O}^i of the data set with the distance $\mathcal{D}_{\mathcal{R}}(\mathbf{O}^i)$ to the representative set \mathcal{R} , computed with the HMMs $\lambda_1 \dots \lambda_R$ as:

$$\mathcal{D}_{\mathcal{R}}(\mathbf{O}^i) = \begin{bmatrix} \mathcal{D}(\mathbf{O}^i, \mathbf{P}_1) \\ \mathcal{D}(\mathbf{O}^i, \mathbf{P}_2) \\ \vdots \\ \mathcal{D}(\mathbf{O}^i, \mathbf{P}_R) \end{bmatrix} = \frac{1}{T} \begin{bmatrix} P(\mathbf{O}^i | \lambda_1) \\ P(\mathbf{O}^i | \lambda_2) \\ \vdots \\ P(\mathbf{O}^i | \lambda_R) \end{bmatrix} \quad (5.7)$$

where T is the length of the sequence \mathbf{O}^i . As in Section 4.3, let us call this space the similarity space $\mathcal{S}_{\mathcal{R}} \in \mathbb{R}^R$;

- perform clustering in this space, using a general technique (not only hierarchical clustering, but also k-means or others).

In the most general case, the representative set \mathcal{R} is the whole data set \mathcal{T} , resulting in a similarity space of dimensionality equal to the cardinality of the set \mathcal{T} . Even if unapplicable for large data set, it is interesting to analyze the discriminative power of such a space.

5.3.2 Experimental results

In this section the proposed technique is compared with the standard HMM clustering presented in Section 5.2. Once obtained the likelihood distance matrix, the clustering (step 3 of Section 5.2) is obtained by using three algorithms:

- two variants of the agglomerative hierarchical clustering techniques: the Complete Link scheme, and the Ward scheme, already described in Section 5.1.1.
- a non parametric, pairwise distance-based clustering technique, called “Clustering by friends” [64], described in the previous section.

Regarding the proposed approach, once obtained the similarity representation with $\mathcal{R} = \mathcal{T}$ (*i.e.* by using all sequences as representatives), we used three clustering algorithms:

- again the hierarchical agglomerative complete link and Ward methods, where distance is the Euclidean metrics in the similarity space: this is performed to compare the two representations with the same algorithms;
- standard K-means algorithm [108, 12], already presented in Section 5.1.1.

Clustering accuracies were measured by using synthetic and real experiments. Regarding the synthetic case, we consider a 3-class synthetic problem, where sequences were synthetically generated from the three generative HMM defined in Fig. 5.1. The data set is composed of 30 sequences (with the length of 400) from

$$\begin{aligned}
 \mathbf{A} &= \begin{array}{|c|c|c|} \hline 1/3 & 1/3 & 1/3 \\ \hline 1/3 & 1/3 & 1/3 \\ \hline 1/3 & 1/3 & 1/3 \\ \hline \end{array} \quad \boldsymbol{\pi} = \begin{array}{|c|} \hline 1/3 \\ \hline 1/3 \\ \hline 1/3 \\ \hline \end{array} \quad \mathbf{B} = \begin{array}{|c|c|} \hline \mu_1 = 1 & \sigma_1^2 = 0.6 \\ \hline \mu_2 = 3 & \sigma_2^2 = 0.6 \\ \hline \mu_3 = 5 & \sigma_3^2 = 0.6 \\ \hline \end{array} \\
 & \text{(a)} \\
 \\
 \mathbf{A} &= \begin{array}{|c|c|c|} \hline 1/3 & 1/3 & 1/3 \\ \hline 1/3 & 1/3 & 1/3 \\ \hline 1/3 & 1/3 & 1/3 \\ \hline \end{array} \quad \boldsymbol{\pi} = \begin{array}{|c|} \hline 1/3 \\ \hline 1/3 \\ \hline 1/3 \\ \hline \end{array} \quad \mathbf{B} = \begin{array}{|c|c|} \hline \mu_1 = 1 & \sigma_1^2 = 0.5 \\ \hline \mu_2 = 3 & \sigma_2^2 = 0.5 \\ \hline \mu_3 = 5 & \sigma_3^2 = 0.5 \\ \hline \end{array} \\
 & \text{(b)} \\
 \\
 \mathbf{A} &= \begin{array}{|c|c|c|} \hline 1/3 & 1/3 & 1/3 \\ \hline 1/3 & 1/3 & 1/3 \\ \hline 1/3 & 1/3 & 1/3 \\ \hline \end{array} \quad \boldsymbol{\pi} = \begin{array}{|c|} \hline 1/3 \\ \hline 1/3 \\ \hline 1/3 \\ \hline \end{array} \quad \mathbf{B} = \begin{array}{|c|c|} \hline \mu_1 = 1 & \sigma_1^2 = 0.4 \\ \hline \mu_2 = 3 & \sigma_2^2 = 0.4 \\ \hline \mu_3 = 5 & \sigma_3^2 = 0.4 \\ \hline \end{array} \\
 & \text{(c)}
 \end{aligned}$$

Fig. 5.1. Generative HMMs for synthetic data testing: \mathbf{A} is the transition matrix, $\boldsymbol{\pi}$ is the initial state probability, and \mathbf{B} contains the parameters of the emission density (Gaussians with the indicated means and variances).

each of the three classes; the dimensionality of the similarity space $\mathcal{S}_{\mathcal{R}}$ is thus $N = 90$. Notice that this clustering task is not easy, as the three HMMs are very similar to each other, only differing slightly in the variances of the emission densities. The accuracy of clustering could be quantitatively assessed, by computing the number of wrongly composed clusters: a clustering error occurs if a sequence is assigned to a cluster in which the majority of the sequences are from another source. Results are proposed in Table 5.4, averaged over 10 repetitions. The proposed methodology largely outperforms standard clustering approaches in

Table 5.4. Clustering results on synthetic experiments.

Standard classification	
ML classification	94.78%
Standard clustering	
Aggl. complete link	64.89%
Aggl. Ward	71.33%
Clus. by Friends	70.11%
Clustering on similarity space $\mathcal{S}_{\mathcal{T}}$	
Aggl. complete link	95.44%
Aggl. Ward	97.89%
k-means	98.33%

this experiment: the best performing algorithm is the partitional k-means on the similarity space, which produces an almost perfect clustering. It is important to note that clustering results in the similarity space are better than the standard ML classification results, confirming the fact, showed in the previous chapter, that similarity space is an highly discriminant feature space.

The real experiment regards 2D shape recognition, where shapes were modelled as proposed in Chapter 6; the database was provided by Sebastian *et al.* [197], and is shown in Fig. 5.2. In this case, shapes are given without any label; only the num-

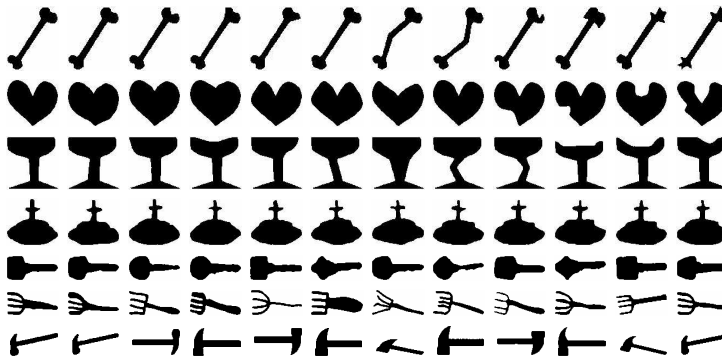


Fig. 5.2. Objects set used for testing.

ber of clusters is known: the algorithm tries to group them into different clusters, basing on their similarity. Results, averaged over 10 repetitions, are proposed in Table 5.5. We can note that there is a big improvement with the use of the simple k-means algorithm in our similarity space. From these tables it is suggested that

Table 5.5. Clustering results on real experiments.

Standard classification	
ML classification	81.55%
Standard clustering	
Aggl. complete link	78.69%
Aggl. Ward	22.86%
Clus. by Friends	70.0%
Clustering on the similarity space $\mathcal{S}\mathcal{T}$	
Aggl. complete link	63.10%
Aggl. Ward	77.62%
k-means	88.21%

the proposed representation seems to be able to provide a great discrimination, resulting in a great increasing of the clustering accuracies.

5.3.3 The choice of the representative set \mathcal{R}

The main problem of similarity-based representation is the resulting high dimensionality of the similarity space, which is equal, in the basic approach, to the cardinality of the data set. Thus, a method for reducing the dimensionality of the similarity space is needed. Unfortunately, in this case it is not possible to use the techniques proposed for the classification case in Section 4.3, as for example Fisher Discriminant Analysis or Matching Pursuit. This is in fact an unsupervised case, in which no labels are attached to patterns; it is therefore impossible to use this information, which represents the leading criterion of such methodologies.

Here, the problem is addressed by the use of two similar strategies: the Principal Component Analysis [113] and the Independent Component Analysis [99], briefly resumed in the appendix A. These techniques try to reduce the space dimensionality by performing a linear projection of the data onto a lower dimensional space.

The strategy is indeed simple: the computed space $\mathcal{S}_{\mathcal{T}}$ is reduced using PCA and ICA, and clustering is performed in such a reduced space. Results on real experiments, for different reduction levels, are shown in Table 5.6. For clarity reasons, the results obtained in the whole space $\mathcal{S}_{\mathcal{T}}$ are also presented, together with the dimensionality of the resulting space, in order to give an idea of the reduction obtained. The use of the Principal Component Analysis and the Independent

Table 5.6. PCA and ICA reduction on the 2D experiment, together with resulting dimensionality.

Clustering on the similarity space $\mathcal{S}_{\mathcal{T}}$				
Dimensionality	k-means		Agg. Ward	
84	88.21%		77.62%	
Clustering on the reduced similarity space				
Dimensionality	k-means		Agg. Ward	
	PCA	ICA	PCA	ICA
2	76.79%	76.55%	67.14%	70.71%
3	84.59%	89.05%	77.38%	84.52%
4	85.59%	78.75%	77.62%	70.83%
5	84.11%	81.31%	77.38%	83.21%
6	84.35%	83.69%	77.38%	82.62%
7	83.51%	84.05%	77.38%	81.31%

Component Analysis seems to be able to remove most of the redundancy present in the data, in general not decreasing too much the accuracy of the clustering, and in some case increasing it. The reduction of the dimensionality is notable, passing from 84 to very few directions.

A comparison between the two reduction techniques reveals that there are not relevant differences between their accuracy, even if some considerations can be made:

- ICA and PCA show the same good behavior when using K-means algorithm; the best performance, nevertheless, is achieved for different dimensionality: 4 for the PCA and 3 for the ICA;
- ICA seems to be very suited to be used with the Ward agglomerative clustering, presenting a great improving in the results with respect to the PCA approach; moreover, the use of ICA permits to enhance very sensibly the performance of this algorithm, which performs better in this reduced space than in the original one;
- the best result is obtained by reducing the space to dimensionality 4 by using ICA and the K-means method; in this case a lot of redundancy is removed, resulting in an accuracy greater than in the not reduced case.

5.3.4 Future perspectives

A possible problem of these linear reduction techniques is the computational burden needed: training one HMM for each sequence, building the whole space, and then computing the transformation, could be quite computationally expensive. Moreover, the system could not generalize to novel patterns, as components have to be (as principle) re-computed. Thus, other techniques should be derived, and this will be the topic of the future research. The first idea is to use a greedy approach similar to the Matching Pursuit algorithm presented in the previous chapter, where a measure of the clustering goodness has to be substituted for the classification accuracy. One candidate is the Davies-Bouldin index [56], presented in Section 5.2.2. The second idea is to cast this problem in a feature selection context, where the prototypes to be chosen are the features to be selected. Nevertheless, these topics will be object of future investigations.

Applications

Summary

In this part, some applications of the Hidden Markov Models approach to practical Computer Vision and Pattern Recognition problems are presented. More specifically, only those systems for which the employment of HMMs produces a surplus in the application context are presented.

In particular, in Chapter 6 the use of the HMM for 2D shape recognition is investigated, analyzing its robustness against rotations, translations, occlusions, affine projections and noise. Chapter 7 presents the use of HMMs for face recognition, showing that this method outperforms all other methods in standard database as ORL. These two applications are important in this thesis, as they have been extensively used through the first part, in order to test most of the techniques proposed in Chapter 3, 4, and 5.

Another application, presented in Chapter 8, regards the use of HMM modelling for video sequence understanding. In particular, HMMs are used to summarize a video sequence, gathered from a static camera, in order to obtain a spatial segmentation of the scene: the resulting regions are featured by a chromatic and temporal homogeneity. An application of this spatio-temporal segmentation to the background modelling problem is also presented, showing that the proposed approach could drastically improve the management of sudden not uniform illumination changes.

2D shape classification

6.1 Introduction

Object recognition, shape modelling and classification constitute active research areas in computer vision. Moreover, these issues are receiving growing attention thanks to the availability of visual databases and the related necessity for retrieving information not only by textual queries but also on the basis of the image content.

Three-dimensional (3D) object recognition has been faced by a large number of different approaches [72]. Among these, many techniques are based on the analysis of two-dimensional (2D) aspects (images) of objects, and many studies deal with 2D shape classification or *planar* object recognition. A basic issue to be resolved first consists in the type of representation of an object, *i.e.*, the features to be used to describe it. Object contours are widely selected features, as they are easily estimated from an image and well represent the semantic information also from a perceptual point of view. Different types of approaches have been proposed in the last few years, such as Fourier descriptors, chain code, curvature-based techniques, invariants, auto-regressive coefficients, Hough-based transforms, associative memories, B-splines, and many others, each with different characteristics, like robustness to noise and occlusions, invariance to translation, rotation and scale, computational requirements, and accuracy [42].

In this context, this chapter investigates the capabilities of Hidden Markov Models for 2D shape classification, where shapes are represented by contours and described using a curvature approach [72]. The use of HMMs for shape analysis has not been widely addressed. To our knowledge, only a few papers have been found to exhibit some similarities to our approach [94, 7, 80, 38]. He and Kundu [94] were the first to employ HMMs for 2D shape recognition. In their approach, contours were represented by auto regressive coefficients computed on segments extracted from the shape boundary. Results were quite interesting and presented as a function of the number of HMM states ranging from 2 to 6, using both stationary and non-stationary models. In [7], the 2D shape classification task was addressed by use of circular HMMs: this particular HMM topology allows one to achieve good classification accuracy with respect to scaling and deformations, and also presents useful characteristics for model training and testing. However, in both works, no examples using shapes explicitly affected by noise or affine object transformations

are reported. Moreover, even if sensitivity to small occlusions is analyzed, shapes are always constrained to be closed contours. Another research study [80] addressed shape recognition: it compared HMMs with a syntactic modelling technique based on stochastic context-free grammars. No original solutions were proposed for HMM design, and the goal of this study was to show the advantages of HMMs over the other method. Recently, another interesting approach was proposed in [38], in which Fourier spectral features were used to classify 2D shapes. A particular HMM topology was introduced in order to deal directly with these features, but, also in this case, shapes were constrained to be closed, and occluded and noisy views were not explicitly analyzed.

In this chapter, the capabilities of HMMs in recognizing planar objects are investigated, showing HMMs performances in the cases of translation, rotation, noise, occlusions, shearing transformations, and combined perturbations. It is worth noting that our approach does not rely on any specific HMM topology or particular training algorithm; moreover, object shapes are *not* constrained to be closed, or represented by using a specific number of symbols. Actually, when objects are occluded, the resulting boundaries are not necessarily closed; in this sense, our algorithm classifies any (closed or open) symbol string.

In training HMMs, particular attention was devoted to the initialization of the training session, using a Gaussian Mixture Model clustering approach. As explained above in this thesis, the initialization issue is a crucial step, because of the local behavior of the standard procedure used to estimate HMM parameters. Another practical but fundamental issue to be resolved when using HMMs is the determination of their structure, namely, the topology and the number of states. The choice of a good model structure is basic to the effectiveness of the learning. In our approach, no constraining assumptions are formulated about the HMM topology, whereas, concerning the number of states, the BIC On Initialization (BOI) approach detailed in Section 3.4 is applied.

Further, the classification scheme is also evaluated in order to assess and objectively quantify the *reliability* of the shape recognition. Using the measure proposed in Section 4.2, the reliability of the classification is tested under various conditions.

The proposed approach was tested using two different databases, in order to assess the robustness of the method to different transformations of objects, such as translation, rotation, occlusion, noise, shearing, and combined perturbations. The resulting high performances on two standard database make the proposed method an interesting alternative to typical shape classification algorithms.

The chapter is organized as follows. In Section 6.2, the global description of the strategy used is presented. In Section 6.3, experimental procedures and results are reported, and Section 6.4 contains the proposed analysis of the reliability of the classification scheme. Finally, in Section 6.5 conclusions are drawn and future developments are suggested.

6.2 The strategy

In this section, the proposed strategy is explained. After describing the object representation, we detail the classification system, focusing briefly on the initialization and model selection issues. The whole strategy is summarized in Fig. 6.1.

6.2.1 Object representation

In our approach, object contours are modelled using curvature [72] coefficients: the curvature is a scalar value that could be computed locally and represents an estimate of the second derivative of the boundary. In this way, starting from a generic boundary point, an object is represented by a sequence of curvature coefficients, namely, a curvature signal. This method is widely used in representing contours, thanks to its attractive intrinsic properties: first, this representation is invariant to object translation; second, object rotation is equivalent to phase translation of the curvature signal. In other words, the scalar curvature value computed at each boundary point is rotationally invariant, but the sequence of these values depends on the initial point. Object rotation implies a change in the initial point, so the curvature signal, in general, turns out to be shifted. The third, and most important point, lies in the fact that the curvature value can be computed for open contours, thus allowing one to deal with occlusions of objects. The main drawback of this method is sensitivity to noise. One possible solution is to apply a real, with quite large variance, Gaussian filter to the (X,Y) contour coordinates, so reducing the noise impact on the signal computation. Moreover, the use of Hidden Markov Models is able to recover from some noisy situation quite well, as can be noticed below.

In our approach, the curvature is computed as follows:

1. Contours are extracted by using the *Canny* edge detector [40], a well-known edge extraction technique.
2. The boundary is approximated by segments of approximately fixed length d_L .
3. Finally, the curvature value at point x is computed as the angle between the two consecutive segments intersecting at x .

For a not occluded object, the initial point is the rightmost point lying on the horizontal line passing through the object centroid, following the boundary in an anticlockwise manner. If the object is occluded, the endpoint allowing the contour to be followed in an anticlockwise way is considered as the initial point.

6.2.2 Training

The obtained curvature representation is then used to train a continuous HMM, where the emission probability of each state is represented by a one-dimensional Gaussian function. Training is performed using the standard Baum-Welch re-estimation method, which is stopped at likelihood convergence. Each HMM is carefully initialized, using the GMM clustering strategy described in detail in Section 3.4.

The number of states is roughly estimated using the BOI method described in Section 3.4. In few words, this method chooses the model by performing a model selection analysis of the GMM clustering phase, *i.e.*, choosing the mixture that best fits the data is chosen. The number of states of the HMM is therefore set as the number of Gaussians of the best mixture so that only one HMM training session is needed. It is worth noting that this model selection scheme determines the model that best fits the *unrolled* sequence: in this sense this is a coarse model selection

scheme, as only the curvature values are considered and not the order in which they appear. Nevertheless, this is quite a reasonable assumption, which considers a shape as being made up of approximating segments with nearly similar curvatures, each group being assigned to a single state. The dynamics of the sequence, *i.e.*, the way in which these segments are ordered, is thus encoded into the transition matrix.

At the end of the training phase, we have one HMM λ_i for each object obj_i .

<p><u>Training:</u> for any object obj_i:</p> <ul style="list-style-type: none"> • extract edges with the Canny edge detector; • calculate the related curvature signature $C(obj_i)$; • train an HMM λ_i on $C(obj_i)$: the HMM is initialized with the GMM clustering; the number of HMM states is estimated by using the BIC criterion in the initialization phase.
<p><u>Classification:</u> given an unknown sequence \mathbf{O}:</p> <ul style="list-style-type: none"> • for each model λ_i, compute the probability $P(\mathbf{O} \lambda_i)$; • classify \mathbf{O} as belonging to class C_ℓ, where $\ell = \arg \max_i P(\mathbf{O} \lambda_i) \quad (6.1)$

Fig. 6.1. The global strategy.

6.3 Results and discussion

The proposed method was tested by using two sets of shapes found in the literature. The first data set was employed by He and Kundu in [94], and is shown in Fig. 6.2. The second data set was used by Sebastian, Klein, and Kimia in [197], and is plotted in Fig. 6.5. In the first case, one HMM for each object has been trained, following the strategy proposed in Section 6.2. It is worth noting that each HMM is trained using the *only* object model present in the data set (Fig. 6.2), without any variation, so that the following results are obtained by training a single shape. As an example, the HMMs of the first two objects, obtained after the training session, are shown in Fig. 6.3. We can notice that the HMM trained on the second object (below), appearing visually less complex than the first one (up), presents a smaller number of states.

Invariance to rotation, occlusion, noise, shearing and a combination of these transformations is tested, whereas invariance to translation is automatically managed by the curvature representation.

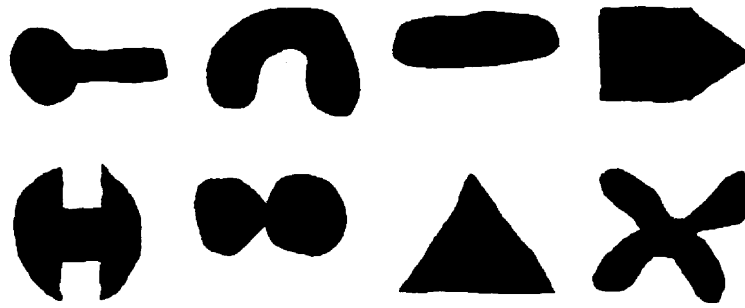


Fig. 6.2. Shapes used for the testing procedure.


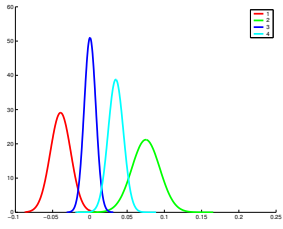

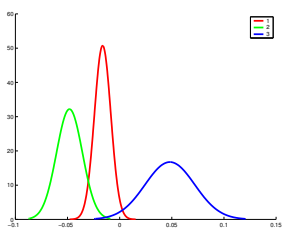
Shape	Emission Probability	Transition Probability																
		<table border="1"> <tr> <td>0.94</td> <td>0.00</td> <td>0.06</td> <td>0.00</td> </tr> <tr> <td>0.00</td> <td>0.96</td> <td>0.00</td> <td>0.04</td> </tr> <tr> <td>0.02</td> <td>0.00</td> <td>0.96</td> <td>0.02</td> </tr> <tr> <td>0.00</td> <td>0.02</td> <td>0.02</td> <td>0.96</td> </tr> </table>	0.94	0.00	0.06	0.00	0.00	0.96	0.00	0.04	0.02	0.00	0.96	0.02	0.00	0.02	0.02	0.96
0.94	0.00	0.06	0.00															
0.00	0.96	0.00	0.04															
0.02	0.00	0.96	0.02															
0.00	0.02	0.02	0.96															
		<table border="1"> <tr> <td>0.98</td> <td>0.01</td> <td>0.01</td> </tr> <tr> <td>0.03</td> <td>0.97</td> <td>0.00</td> </tr> <tr> <td>0.02</td> <td>0.00</td> <td>0.98</td> </tr> </table>	0.98	0.01	0.01	0.03	0.97	0.00	0.02	0.00	0.98							
0.98	0.01	0.01																
0.03	0.97	0.00																
0.02	0.00	0.98																

Fig. 6.3. Models of the first two objects.

6.3.1 Rotation

First of all, let us consider the effect of rotation on the signature of an object. We recall that the curvature is rotationally invariant at each boundary point, and that the curvature signal is computed starting from the rightmost point lying on the horizontal line passing through the object centroid; therefore, the rotation of an object involves in general a change in the starting point. From these two considerations we can infer that the object rotation causes only a *shift* in the curvature signal.

To test the invariance of our method, each object was rotated 10 times by an angle randomly chosen from 0 to 2π . The resulting classification accuracy was 100%, *i.e.*, the HMM was able to exactly recognize rotated objects¹.

6.3.2 Occlusion

Object occlusion is rendered by considering a fragment of the object boundary, starting from a point randomly chosen. It should be noted that the random choice of the initial point is important for assessing the invariance of the strategy to the specific part occluded.

Given an open contour, *i.e.*, a fragment of the original one, the curvature is calculated as explained in Section 6.2. Because of the local curvature properties, the resulting string is a substring of the original one. As pointed out in the following, an HMM trained on a sequence \mathbf{O} was able to effectively recognize a substring of \mathbf{O} . The trial was performed by occluding each object 10 times, starting from a randomly chosen initial point: occlusion varied from 5% to 50% (only an half of the whole object was visible), and results are given in Table 6.1(a). The obtained

Table 6.1. Classification accuracies obtained in: (a) occlusion experiments, for different occlusion levels; (b) noise experiments, for different noise levels.

Occlusion Level	Classification Accuracy	σ^2	Accuracy
5%	100%	0.05	100.00%
10%	100%	0.15	100.00 %
15%	100%	0.25	100.00 %
20%	100%	0.35	100.00%
25%	100%	0.45	97.50%
30%	100%	0.55	91.25%
35%	100%	0.65	83.75%
40%	97.5%	0.75	80.00%
45%	96.25%	0.85	71.25%
50%	95%		

(a)
(b)

accuracies were considerably high: also when 35% of each object was occluded, our technique was able to correctly classify all the fragments. These results are particularly valuable, considering that occlusion is one of the most severe problems of many object recognition methods.

6.3.3 Noise

We tried to investigate the robustness of our approach in noisy situations. To this end, two synthetic noising schemes are proposed. First, a Gaussian noise, with zero

¹ From this fact, one might deduce that HMMs are able to recognize shifted signals, *i.e.*, that they are shift-invariant, but this property is not proved for all HMMs.

mean and variance σ^2 ranging from 1 to 5, is added to the (X, Y) coordinates of an object. Shapes are not much affected by this kind of noise, and the resulting accuracy is 100%, thanks to the Gaussian filter applied before calculating the curvature: this filter is able to remove completely the effects of this kind of noise. An example of object affected by noise with variance $\sigma^2 = 2$ is shown in Fig. 6.4(a). The second type of noising scheme is adopted to degrade the object shapes more heavily. It is obtained by adding Gaussian noise to the *differential* signal, which results from computing, for each boundary point, the difference between the coordinates of each point and those of the following one. Subsequently, a zero-mean Gaussian noise is added to this difference-code; finally, the coordinates' values are re-computed from the pre-stored initial point. Examples of degradation of the first object are presented in Fig. 6.4(b) and (c) for two values of the variance σ^2 , and show that the degradation is significant and larger than that derived by the first type of noise.

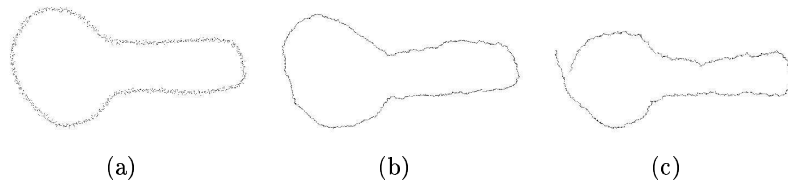


Fig. 6.4. Noising of the the first object, by varying the noising scheme: (a) first proposed scheme, with variance $\sigma^2 = 2$; (b-c) second proposed scheme, with variance (b) $\sigma^2 = 0.65$ and (c) $\sigma^2 = 0.85$.

The test set was obtained by adding noise to each object ten times, obtaining globally 80 noisy objects; the resulting accuracy values are presented in Table 6.1(b), taking the noise level (variance) as the varying parameter. As one can notice, the results are quite good, showing that HMMs can reduce the intrinsic curvature sensitivity to noise.

6.3.4 Combined transformations

After assessing the robustness of the proposed method to single-object degradations, experiments were carried out to evaluate the algorithm performances with respect to combined transformations, *i.e.*, 1) rotation and occlusion, and 2) rotation, occlusion, and added noise. Occluded and rotated objects were obtained by rotating the objects by a random angle (between 0 and 2π) and considering fragments of their contours. From the results presented in Table 6.2(a), it can be noticed that the accuracies are very good also in this case, even though lower than those obtained for unrotated objects, as expected. A more difficult situation occurred when objects were first rotated by a random angle, then occluded, and finally degraded by the second type of noise described in Section 6.3.3. Results are given in Table 6.2(b). Also in this case, the accuracy values are satisfactory but rapidly degrading with increasing noise level.

Table 6.2. Classification accuracies for combined transformations: (a) occluded and rotated objects, for different occlusion levels; (b) occluded, rotated and noisy objects, for different occlusion and noise levels.

Occlusion Level	Classification accuracy
5%	100%
10%	100%
15%	100%
20%	100%
25%	98.75%
30%	98.75%
35%	95%
40%	94%
45%	85%
50%	87.5%

(a)

Occlusion Level	Accuracy		
	$\sigma^2 = 0.1$	$\sigma^2 = 0.3$	$\sigma^2 = 0.5$
10%	100.00%	97.50%	87.50%
20%	98.75%	93.75%	80.00%
30%	98.75%	90.00%	80.00%
40%	93.75%	87.50%	77.50%
50%	86.25%	83.75%	75.00%

(b)

6.3.5 Shearing

Finally, the robustness of our approach to shearing transformations was assessed. This experiment was characterized by a higher degree of complexity than those of the previous tests, in that it consisted in an actual strong deformation of objects. The shearing transformation was obtained by considering the shape as a plane in a 3D space, and varying its *tilt* and *slant* angles. The tilt τ of a planar surface is defined as the angle formed by surface normal projected in the image plane and the reference x axis, while the slant ϕ is the angle between the surface normal and the line of the sight [198]. The resulting transformed surface was then orthonormally projected on the original (X,Y) plane to get the usual fronto-parallel view.

The first test was carried out by rotating each object by fixed growing tilt and slant angles, in steps of 10 degrees, and then applying the method. The related results are presented in Table 6.3, for different tilt and slant values. Accuracies are presented as percentage values, in order to standardize the layout of all results in the paper. In this case, nevertheless, there are only 8 test items, for fixed tilt and slant angles; therefore, each classification error decreases the accuracy value by a step of 12.5%.

From these results, we can notice that sensitivity of our approach to tilt and slant changes is very different. In fact, the variation of slant angle results in a severe distortion of the object appearance, while the variation of tilt angle could

Table 6.3. Classification accuracies obtained in shearing experiments, for different tilt and slant values.

		slant ϕ								
		0°	10°	20°	30°	40°	50°	60°	70°	80°
tilt τ	0°	100%	100%	100%	100%	100%	100%	62.5%	37.5%	25.0%
	10°	100%	100%	100%	100%	100%	100%	50.0%	37.5%	25.0%
	20°	100%	100%	100%	100%	100%	100%	50.0%	37.5%	37.5%
	30°	100%	100%	100%	100%	100%	87.5%	50.0%	50.0%	37.5%
	40°	100%	100%	100%	100%	100%	75.0%	62.5%	50.0%	37.5%
	50°	100%	100%	100%	100%	100%	75.0%	62.5%	37.5%	37.5%
	60°	100%	100%	100%	100%	100%	75.0%	50.0%	37.5%	37.5%
	70°	100%	100%	100%	100%	100%	62.5%	50.0%	50.0%	37.5%
	80°	100%	100%	100%	100%	100%	62.5%	50.0%	50.0%	25.0%
	90°	100%	100%	100%	100%	100%	50.0%	50.0%	37.5%	12.5%
	100°	100%	100%	100%	100%	87.5%	50.0%	50.0%	37.5%	25%
	110°	100%	100%	100%	87.5%	87.5%	50.0%	50.0%	37.5%	25%
	120°	100%	100%	100%	87.5%	87.5%	50.0%	37.5%	37.5%	25%
	130°	100%	100%	100%	87.5%	87.5%	62.5%	37.5%	37.5%	25%
	140°	100%	100%	100%	100%	87.5%	75.0%	50.0%	37.5%	25%
	150°	100%	100%	100%	100%	87.5%	75.0%	50.0%	37.5%	25%
	160°	100%	100%	100%	100%	100%	100%	50.0%	37.5%	25%
170°	100%	100%	100%	100%	100%	100%	62.5%	37.5%	25%	

be roughly considered as a kind of rotation of the slant derived transformation. Our approach is very robust to shape rotations, so the performance level is mostly driven by slant variations. Results proposed in Table 6.3 demonstrates that our approach is truly robust against shearing: only for large slant values, corresponding to severe distortions of the objects, the classification accuracies decrease, but, however, still remaining more than two or three times the random classification level.

The second experiment was performed by adding synthetic noise to the sheared shapes, using the second type of noise described in Section 6.3.3. The applied noise level was $\sigma^2 = 0.35$, a medium noise level, and each object was randomly affected by noise 10 times. The averaged results are given in Table 6.4, showing that the accuracies are very satisfactory also in this case.

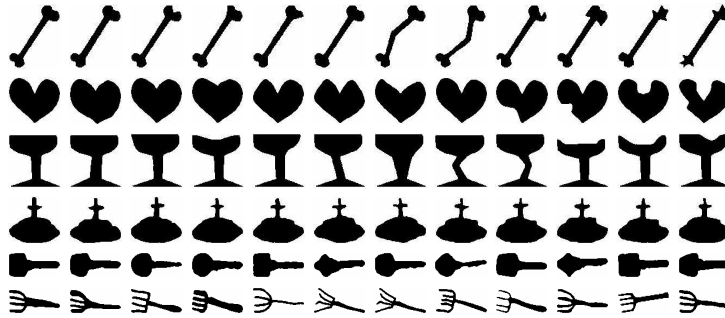
In general, the HMMs proved to be very powerful in classifying shapes and a large class of deformations, especially occlusions, which constitute one of the most severe problems in object recognition. Such results are particularly interesting if one recalls that the HMMs have been trained using only one shape model, which did not capture all (also strong) deformations applied to the testing objects.

6.3.6 Results on the second data set

To increase the statistical significance of the results, the method was also tested on another set of shapes, utilized in [197]. This set was composed of 6 classes, each containing 12 object instances (see Fig. 6.5). Unlike the previous data set, this model database is characterized by several deformed object instances for each

Table 6.4. Classification accuracies for noisy and sheared objects, at different tilt and slant angles.

		slant ϕ								
		0°	10°	20°	30°	40°	50°	60°	70°	80°
tilt τ	0°	98.8%	95.0%	100%	97.5%	97.5%	95.0%	65.0%	32.5%	27.5%
	10°	95.0%	98.8%	100%	100%	100%	93.8%	52.5%	37.5%	30.0%
	20°	100%	100%	100%	100%	96.3%	96.3%	51.3%	37.5%	37.5%
	30°	100%	100%	98.8%	100%	98.8%	88.8%	51.3%	48.8%	37.5%
	40°	97.5%	98.8%	100%	100%	100%	77.5%	52.5%	50.0%	37.5%
	50°	98.8%	98.8%	98.8%	100%	100%	75.0%	48.8%	37.5%	33.8%
	60°	97.5%	100%	98.8%	98.8%	98.8%	71.3%	46.3%	37.5%	31.3%
	70°	98.8%	98.8%	98.8%	98.8%	92.5%	70.0%	48.8%	40.0%	22.5%
	80°	96.3%	100%	98.8%	98.8%	82.5%	56.3%	43.8%	40.0%	17.5%
	90°	98.8%	98.8%	97.5%	97.5%	75.0%	53.8%	37.5%	38.8%	22.5%
	100°	96.3%	98.8%	97.5%	92.5%	71.3%	48.8%	33.8%	40.0%	22.5%
	110°	97.5%	96.3%	100%	91.3%	78.8%	43.8%	35.0%	37.5%	23.8%
	120°	100%	98.8%	100%	87.5%	80.0%	50.0%	36.3%	37.5%	26.3%
	130°	98.8%	98.8%	97.5%	92.5%	82.5%	63.8%	38.8%	37.5%	25.0%
	140°	100%	97.5%	98.8%	93.8%	83.8%	67.5%	50.0%	33.8%	25.0%
	150°	98.8%	96.3%	97.5%	96.3%	88.8%	72.5%	50.0%	35.0%	25.0%
	160°	98.8%	98.8%	100%	100%	97.5%	86.3%	52.5%	32.5%	25.0%
170°	100%	97.5%	100%	100%	98.8%	90.0%	65.0%	36.3%	25.0%	

**Fig. 6.5.** The second object set used for testing.

class. In this case, instead of training one HMM for each shape class, one HMM was trained for each instance: this resulted in 72 HMMs. Accuracy was computed by using the Leave One Out error scheme [210] and assigning an unclassified object to the class of the object whose model showed the maximum likelihood. The results were equal to 100%, confirming that the proposed approach was still robust and accurate, also for this set. Moreover, we evaluated the performances in the presence of occlusions, using the same procedure as described in Section 6.3.2; the results are presented in Table 6.5. Also these results are very satisfactory, even though not so good as for the Kundu database. Nevertheless, this model database contains instances of the same class that are the same object only semantically, but the

Table 6.5. Classification accuracies, in the presence of occlusions for the second object set employed for testing, at different occlusion levels.

Occlusion Level	Classification accuracy
10%	99.03%
20%	97.36%
30%	94.02%
40%	88.05%
50%	82.22%

related shapes are very different (e.g., the key class); yet the method’s performances did not degrade too much.

6.4 Significance of the classification scheme

In Section 4.2, two measures for assessing the reliability of the ML classification scheme adopted were introduced. As explained in Section 4.1, standard Maximum Likelihood scheme assigns an unknown item to the class whose model shows the highest likelihood. The idea below the proposed measures is that the difference between the likelihood of the first and the second choice of the scheme could give a measure of how sure is the system of its choice: in the case of correct classification, this could be interpreted as a measure of the robustness of the system. In the case of misclassified pattern, instead, the difference between the choice of the algorithm and the correct choice gives a measure of how wrong is the system decision.

In the 2D shape classification problem, both measures have been computed, in the presence of occlusions and in the presence of noise, separately. For the occlusion experiments, the two measures are plotted, using the same scale, in Fig. 6.6. In Fig. 6.6(a), the RCC factor decreases for increasing occlusion levels, still keeping a good margin. In Fig. 6.6(b), only REC values for occlusion levels higher than 35% are plotted, as no errors are made for lower occlusion levels. From these values, it can be observed that the misclassification reliability in the case of error is very low, and lower than the margin estimated in the case of correct classification (Fig. 6.6(a)).

The same behavior can be noticed looking at the reliability analysis for the experiments in presence of noise (Fig. 6.6(c) and (d)): the margin between the two factors in the two cases is narrowed, but it is still possible to discriminate between correct and wrong classification.

Two conclusions can be drawn from this analysis. First, the large values in the left column of Fig. 6.6, corresponding to the analysis of the correctly classified patterns, confirm that our approach is robust, but, as expected, robustness decreases with increasing task difficulty. Second, it seems relatively simple to obtain a rejection rule by merely thresholding the likelihood difference between the first two choices of the algorithm: if a classification is not sufficiently reliable, it can be rejected.

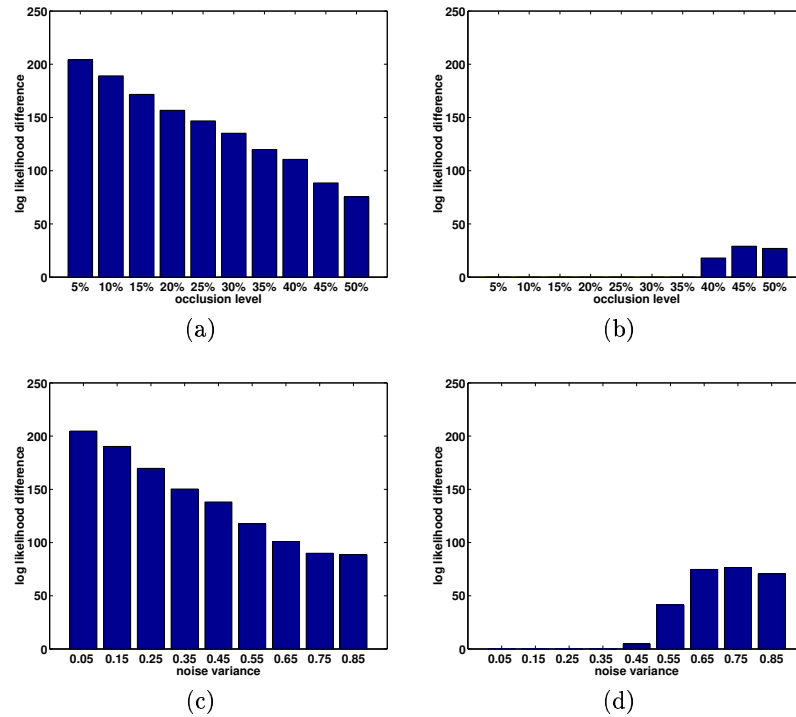


Fig. 6.6. Analysis of the reliabilities of the classification rule in occlusion experiments (a and b), at different occlusion levels, and in noisy experiments (c and d), at different noise levels. All plots are on the same scale. Left) RCC factors. Right) REC factors.

6.5 Conclusions

In this chapter, an HMM-based approach to the classification of planar shapes has been proposed. Given a model database, one HMM has been trained for each object model represented by curvature coefficients, paying particular attention to the HMM initialization and model selection issues during the learning process. Experimental tests on two different data sets have shown that the proposed system is able to recognize objects that are modified instances of the original shapes after rotation, occlusion, shearing and degradation by noise. Based on the evaluation performed on standard databases, the system has proved particularly robust to all these kinds of deformations and noise, despite the intrinsic curvature sensitivity. This demonstrates that the modelling method (*i.e.* Hidden Markov models), the curvature representation adopted for shapes, and the particular training phase performed, succeed in making the proposed approach very robust to several shape degradations and noise. An analysis of the classification scheme has also been proposed, in order to evaluate and quantify the “reliability” of the recognition process, proving that good reliability levels have been obtained.

All these features, together with the good performances achieved, make the proposed method a powerful and general computational approach to shape classification.

Face recognition

7.1 Introduction

Face recognition is undoubtedly an interesting research area, whose importance has increased in recent years, due to its applicability as a biometric system in commercial and security applications. These systems could be used to prevent unauthorized access or fraudulent use of ATMs, cellular phones, smart cards, desktop PCs, workstations, and computer networks. The appealing characteristic of a face recognition system is that, differently from fingerprint or iris biometric systems, it is not an invasive control tool.

A large literature is available on this topic (for a review see [46]): the first approaches, in the 70's, were based on geometric features [120]. One of the best known face recognition method is the so-called *Eigenface* method [200,214,156,18,229], which uses the *Principal Component Analysis* [113] to project faces into a low-dimensional space, where every face can be expressed as a linear combination of the eigenfaces. This method is not robust against variations of the face orientation and one solution was given by the view-based eigenspace method introduced in [176]. Another important approach is the *Elastic Matching* [229,132,209,128], introduced to obtain invariance against expression changes. The idea is to build a lattice on image faces (rigid matching stage), and to calculate at each point of the lattice a bank of Gabor filters. In case of variations of expression, this lattice can warp to adapt itself to the face (elastic matching stage).

Many other methods have been proposed in the last decade, using different techniques, as Neural Networks [51,147,135], Support Vector Machines [88,100] or Hidden Markov Models [194,1,125,160,70], each characterized by different features, like computational requirements, robustness to light changes or to different poses, and others.

To the best of our knowledge, the best results obtained on standard database, as ORL (Olivetti Research Ltd.) database, were proposed in [70] and [125]. These methods are based on DCT (Discrete Cosine Transform) and HMMs, achieving an almost perfect classification. In [70] a pseudo 2D HMM [3] was used for classifying faces, encoded with the DCT coefficients of a set of partially overlapped sub-images. One of the most interesting feature of this method is its direct applicability to JPEG (Joint Photographic Experts Group) images, without any need

of decompressing them. This method reaches a perfect classification rate on the ORL database. The other technique, proposed in [125], makes use of standard one-dimensional HMMs trained on sequences of DCT coefficients extracted from the image. Since this method is used to test most of the methods proposed in the previous chapters of this thesis, it is extensively described in Section 7.2.

A reasonable question could arise with the advent of the new standard JPEG, the so called JPEG2000 [104], which makes use of wavelet coding [55, 223]: is it possible to extend this strategy in order to accomplish with this new standard? To this aim, a comparison between DCT coding and Wavelet coding is presented in Section 7.4, showing that HMM is really effective in recognizing faces also using wavelet coefficients: we obtain a perfect classification accuracy on the ORL database.

The rest of the chapter is organized as follows: in Section 7.2 the approach proposed in [125] is detailed, while Section 7.3 contains a brief introduction to the wavelet approach for image compression. In Section 7.4 the comparison between DCT and Wavelet approach is discussed, while in Section 7.5, conclusions are finally drawn.

7.2 The DCT approach

In this section the method proposed in [125] is detailed. In that approach, the classification of faces was addressed by using HMMs: one model is trained for each class, using standard Baum Welch algorithm; the subsequent classification is performed using standard Maximum Likelihood classification rule. Here, differently than in [125], where the model selection issue was disregarded, the model size was carefully estimated, using the technique proposed in Section 3.6.

The strategy used to obtain the data sequence from a face image consists of two steps. In the first step, a sequence of sub-images of fixed dimension is obtained by sliding over the face image a square fixed size window, in a raster scan fashion, with a predefined overlap (the procedure for scanning the image is visualized in Fig. 7.1). The second step consists in applying the 2D DCT to each gathered sub image. The obtained coefficients are scanned in a zig-zag fashion, analogous to the method used for the JPEG coding. Only few of these coefficients are retained, determining the dimensionality of the observation. By applying this step to all the sub-images of the sequence, we finally obtain the sequence observation. Its dimensionality will be $D \times T$, where D is the number of the DCT coefficients retained, and T is the number of sub-images gathered in the sample scanning operation.

7.3 The Wavelet coding

The wavelet transform [55] has emerged in the last years as a cutting edge technology, within the field of image compression. Wavelet-based coding provides substantial improvements in picture quality at higher compression rates, with respect to standard DCT transform. Over the past few years, a variety of powerful and sophisticated wavelet-based schemes for image compression have been developed

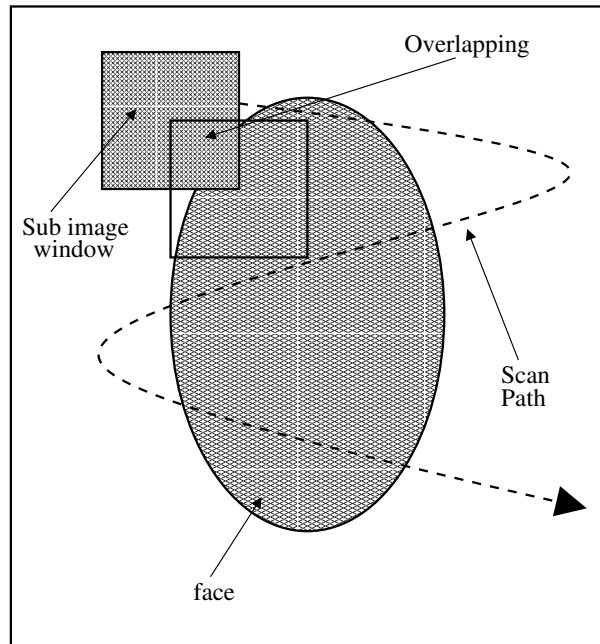


Fig. 7.1. Sampling scheme to generate the sequence of sub-images.

and implemented [191]. Because of these many advantages, the compression technologies used in the upcoming JPEG-2000 standard [104] are all based on the wavelet transform.

Wavelets could be defined as a mathematical tool for hierarchically decomposing functions. The wavelet transform is aimed at describing a function in terms of a coarse overall shape, together with details that range from broad to narrow. Formally, wavelets are functions defined over a finite interval and having a zero-value average. Their basic idea is to represent any arbitrary function $f(t)$ as a superposition of a set of such wavelets or basis functions. These basis functions or *baby wavelets* are obtained from a single prototype wavelet called the *mother wavelet*, by dilations or contractions (scaling) and translations (shifts). For a simple and excellent introduction to wavelets, see [43].

Here we propose to modify the sequence extraction approach presented in the previous section, by substituting the DCT coding with the wavelet coding. In this case we used the Haar wavelets [157, 223], representing the simplest wavelet basis. We employed the non standard decomposition, that alternates between row and column processing, allowing a more efficient coefficients computation. The proposed algorithm computes the coefficients representing the image with a normalized two-dimensional Haar basis, sorting these coefficients in order of decreasing magnitude; then the first M coefficients are retained, performing a lossy image compression. For a more complete treatment of wavelet image compression, see the De Vore's paper [223]. As in the DCT case, the number of retained coeffi-

cients determines the dimensionality of the observation vector, while its length is determined by the number of sub images gathered.

7.4 Comparison between DCT and Wavelet coding

In this section, wavelet and DCT approaches are compared, in order to assess the HMM suitability in wavelet coefficients modelling. The experiments have been conducted on the ORL database¹, which consists of 40 subjects with 10 faces each. 10 examples of subjects from the ORL database are presented in Fig. 7.2, with different 10 images: it is worth noting that this database is characterized by illumination, pose and expression changes between images of the same subject.



Fig. 7.2. 10 example subjects from the ORL face database.

¹ Downloaded from <http://www.uk.research.att.com/facedatabase.html>.

One HMM is built for each subject, using 5 images, while the remaining 5 are used for testing. Training is performed using standard Baum Welch technique, stopping the procedure after likelihood convergence. The adopted classification scheme is the usual one, *i.e.* the Maximum Likelihood (ML) scheme. Experiments are repeated 20 times, in order to increase the statistical significance of the results: this permits to obtain results independent from the training process. Sub image size is fixed during all experiments to 16x16, while the number of the retained coefficients (4, 8 and 12) and the overlapping ratio (50% and 75%) vary. Results are proposed in Table 7.1(a) and (b). From these tables it is evident that Wavelet

Table 7.1. Comparison of accuracies obtained in the ORL database by DCT and Wavelet approaches, for different number of retained coefficients and for different overlap ratios: (a) overlap ratio = 50% and (b) overlap ratio = 75%.

Num. coeff.	DCT accuracy	Wavelet accuracy
4	98.6%	97.4%
8	99.4%	100%
12	100%	100%

(a)

Num. coeff.	DCT accuracy	Wavelet accuracy
4	97.9%	95.4%
8	99.2%	99.5%
12	99.6%	98.8%

(b)

and DCT approaches perform equally well on this database: the Wavelet transform seems to be less effective when using few coefficients. With regards to the performances obtained with the DCT coefficients, it is worthwhile to note that the use of the model selection permits to reach a perfect classification (100%), not obtained in [125] (99.5%).

It is important to note that this approach is very effective in resolving the face recognition problem, outperforming, on the ORL database, all other methods proposed in the literature: this could be observed in Table 7.2, which presents a comparison between published results obtained by the most important face recognition algorithms on the ORL database. The five best performances are displayed in bold fonts in the table: the first three are all based on Hidden Markov Models, the fourth on a n-tuple classifier and the fifth on Support Vector Machines.

It is important to note that the ORL database is a somewhat ideal dataset, with limited variations of the environmental parameters (lightness, face scale and expressions). Nevertheless it is widely employed in the face recognition context, and several results on it are present in the literature: hence it represents a standard database.

Table 7.2. Comparative results on ORL database. “Wavelet + HMM” represents the proposed method. The five best results are displayed in bold font.

Method	Error	Ref.	Year
Top-down HMM + gray tone features	13%	[195]	1994
Eigenface	9.5%	[214]	1994
Pseudo 2D HMM + gray tone features	5.5%	[194]	1994
Elastic matching	20.0%	[229]	1997
PDNN	4.0%	[147]	1997
Continuous n-tuple classifier	2.7%	[149]	1997
Top-down HMM + DCT coef.	16%	[160]	1998
Point-matching and correlation	16%	[133]	1998
Ergodic HMM + DCT coef.	0.5%	[125]	1998
Pseudo 2D HMM + DCT coef.	0%	[70]	1999
SVM + PCA coef.	3%	[88]	2001
Independent Component Analysis	15%	[228]	2002
Gabor filters + rank correlation	8.5%	[8]	2002
SVM + Multilevel B-splines	2.75%	[100]	2002
Wavelet + HMM	0%		2002

7.5 Conclusions

In this chapter the use of Hidden Markov Models for face recognition was investigated. The method proposed in [125], used through the thesis to test algorithms, was detailed. A novel method was proposed, based on the wavelet coding; this method, compared to the DCT approach, proposed equivalent results, assessing the HMM suitability for dealing with the new JPEG2000 image compression standard. Obtained results outperform all results presented in the literature on the ORL database, reaching a perfect classification accuracy.

Spatio-temporal segmentation of video sequences

8.1 Introduction

Image segmentation is an important and challenging problem in image analysis, aimed at discovering and characterizing the different semantic objects of an image. When considering time, such description must evolve, resulting in a more difficult and computationally expensive task, typically called spatio-temporal segmentation. This is usually defined as the partition of the images sequence into spatial regions of motion homogeneity. Several approaches have been proposed in this field, as motion-based segmentation [44], spatial segmentation and motion tracking [59], moving objects extraction [89], and region growing using spatio-temporal similarity [47, 58]. Quantitative evaluation methods have also been suggested [32].

Generally, spatio-temporal segmentation has been successfully applied in several heterogeneous applications. The most important are surely two: the first one is the video surveillance, where the spatio-temporal segmentation is used to discriminate the background from the foreground. The second one is the video indexing and retrieval, where the spatio-temporal segmentation (in this context also called video-segmentation) provides a compact visual representation, eliminating the redundancy in contiguous frames.

Most of the proposed methods present a limitative characteristic: in the case of video-surveillance, the basic model is typically pixel-wise, without any use of region-level information; in the video retrieval context, associations are performed frame-by-frame, without considering the whole single pixel evolution process.

Here we propose a new method for spatio-temporal segmentation, that considers pixel information, and is aimed at partitioning the images sequence (gathered from a static camera) into static regions of homogeneous color and similar temporal evolution. In this case, the resulting segmentation is a spatial segmentation, obtained by using all available information: chromatic (different regions have different gray level values), spatial (each region is connected in the space) and temporal (each region varies color homogeneously in time). By the way of this, spatial knowledge, typically used to obtain a spatial segmentation, is augmented with temporal information, allowing a more detailed and informative partitioning. Even if this definition may appear in some way different from conventional one, this kind of segmentation could be considered a spatio-temporal segmentation of video

sequences as well, as the obtained regions are characterized by spatio-temporal homogeneity. Roughly speaking, we obtain a spatial segmentation of the background by using spatial, chromatic and temporal information. A similar definition was proposed also in the JSEG algorithm [58], where a spatial segmentation was obtained at each time step, using spatial and temporal information: in this case, a first segmentation is obtained from the first frame, and is iteratively updated with subsequent frames.

The basic idea under the proposed approach is the following: first, to consider the chromatic evolution of each pixel (or of a small area): these constitute a set of one-dimensional independent sequences. Subsequently, these sequences are grouped or clustered into similar regions by associating together sequences that are near in space and exhibit similar chromatic evolution.

Sequences are modelled using Hidden Markov Models, which are used for computing the similarity between sequences, with a method similar to those proposed in Chapter 5. Here, a new sequence distance is introduced, able to capture chromatic-temporal differences between sequences. This measure is able to remove non stationary components of the scene evolution, represented not only by noise, but also by foreground. Once given a similarity measure between sequences, a simple and standard region-growing approach is used to obtain the desired segmentation. At the end of the process, a meaningful segmented image is obtained, representing the time evolution of the static part of the scene. It is worthwhile to note that our method does not require to remove the moving objects to analyze the image sequence, since this task is naturally accomplished by the clustering technique together with the proposed similarity measure.

Examples of segmentation are proposed in the experimental session, using both synthetic and real sequences. It will be shown that our method is able to discover homogeneous spatio-temporal components of the sequences, resulting in a quite correct segmentation. In a real case, our method has been also compared with JSEG [58], presenting qualitative better results.

Several applications could take advantage from this compact representation of a video sequence, as video retrieval and video surveillance. In the former context, the compact representation of the sequence, obtained by the segmentation process, allows to reduce the video retrieval problem (that has to deal with the whole video sequence) to a simpler *image retrieval* system. For the latter applications, it will be shown in Section 8.4 that a spatial segmentation of the background can sensibly improve the background modelling, allowing an integration between region and pixel information in standard Time-Adaptive, Per Pixel, Mixture of Gaussians (TAPPMOG) [203,93] techniques. This integration permits to recover from situations where sudden and not global changes of illumination occur.

8.2 The proposed approach

The background of a video sequence can be defined as the part of the sequence that remains spatially static in time, *i.e.*, we assume intuitively that the semantic objects of the scene do not move their position in the sequence: otherwise, we define as foreground objects the spatially moving objects. With this assumption, we know

that, fixed a pixel location P_i , the only temporal variation is due to the evolution of the gray-level, denoted as $I(P_i)_t$. The proposed chromatic-temporal segmentation of the sequence aims at grouping near pixels P_i in regions R_k , where the gray-level intensity $I(R_k)_t$ is 1) *homogeneous* in the region and 2) varies homogeneously during time.

A measure able to capture the chromatic-temporal similarity between adjacent pixels is then needed. Given a pair of neighboring pixels P_1 and P_2 , and the corresponding gray-level temporal evolution $I(P_1)_t$ and $I(P_2)_t$, we need a model able to capture three characteristics: 1) the most stable gray-level components measured in the whole sequence; 2) the chromatic-temporal variation of that components; 3) the sequentiality in which the components vary. In such a case, an adequate model is the Hidden Markov Model with continuous Gaussian emission probability. Using this model all requirements are in fact accomplished: the most important gray level components are modelled by the means μ_i of the Gaussian of the states, the variability of those components is encoded in the covariance matrices Σ_i , and the sequentiality is encoded into the transition matrix \mathbf{A} .

Once given all models λ_i , each one modelling the temporal evolution of a pixel P_i (or of a small neighborhood of pixels), it is necessary to define a similarity measure, in order to decide when a pair of neighboring pixels must be labelled as belonging to the same region. The similarity measure needs to exhibit some precise characteristics: two sequences have to be considered similar if they share a comparable main chromatic and temporal character, independently from the values assumed by the less important components. A possible solution is to use the measures proposed in Chapter 5, here briefly summarized¹:

$$D(i, j) = 1/2(L_{ij} + L_{ji}) \quad (8.1)$$

where $L_{ij} = P(\mathbf{O}_j | \lambda_i)$, $\mathbf{O}_j = I(P_j)_t$ and λ_i is the HMM trained on sequence $\mathbf{O}_i = I(P_i)_t$. Another possibility is

$$D(i, j) = \frac{1}{2} \left\{ \frac{L_{ij} - L_{jj}}{L_{jj}} + \frac{L_{ji} - L_{ii}}{L_{ii}} \right\} \quad (8.2)$$

The problem with these measures is that the Gaussian of each state contributes in the same way to the computation of the probability, because of the forward backward procedure. For our target, nevertheless, it is necessary that the Gaussian of each state contributes differently to the probability computation, depending to the “importance” of the corresponding state. The idea is therefore to “flatten” the Gaussians of those states that are not really important by increasing their variance such that their contribution to the computation of the probability is reduced. In order to obtain a quantitative measure of the “importance” of the state, we used a concept typical in the Markov theory, the so-called *stationary probability* \mathbf{p}_∞ . This probability represents the probability of being in a precise state after an infinite number of transitions: in other words, \mathbf{p}_∞ represents the “average” occupation of each state, after the Markov Chain has achieved the stationary state. We assume that the “importance” of each state is measured by this stationary probability,

¹ Note that typically they are not a proper metric, as not satisfying all the distance's properties.

computed as the left eigenvector of the transition matrix \mathbf{A} associated with the unit eigenvalue (for further details on these concepts see Section 3.6).

The operation of “flattening” is traduced in the transformation of each model λ in a new model λ' , where all components remain unchanged, except the variance σ_j of the Gaussian $\mathcal{N}(\mu_j, \sigma_j)$ of each state S_j , that becomes

$$\sigma'_j = \frac{\sigma_j}{\mathbf{p}_\infty(j)} \quad (8.3)$$

The new distance, called $D_{\text{ES}}(i, j)$ (*Enhanced Stationary*), is then computed using the eq.(8.2) on these modified HMMs λ'_i . The increase of the variance σ_j , corresponding to the flattening of the Gaussian $\mathcal{N}(\mu_j, \sigma_j^2)$, has two beneficial effects: 1) the possibility of matching between Gaussians of important states of different models is increased; 2) Gaussians of non-important states are very flattened, reducing their contribution to the probability computation. It is worthwhile to note that such a metric is able to remove moving objects from the video sequence, as they are considered as not-stationary components of the background model.

Assumed this kind of similarity measure between sequences, the segmentation process can be developed as an ordinary segmentation process of static images. We adopt a simple region growing algorithm: starting the process from some seed-points, we use a threshold S to estimate when two adjacent sequences $I(P_i)$ and $I(P_j)$ are similar using the distance $D_{\text{ES}}(i, j)$. Obviously the value of this threshold could affect the performance of the algorithm: in the experimental session proposed in this thesis, this threshold has been determined by the use of heuristics.

We will see in the experimental session that the modification of the metric (8.2), with the integration of the chromatic-temporal information of the video-sequence, permits a visible improvement of the segmentation process in synthetic experiments and in real sequences.

8.3 Experimental session

The proposed approach was tested using both synthetic and real cases. In the former case, the synthetic sequence contains blocks flickering with the same palette but with different frequency, to which a 0.001 variance Gaussian noise has been added. Some frames of the sequence are shown in Fig. 8.1 (the central region is fixed).

In Fig. 8.2(a) the obtained segmentation is presented, showing that all 9 regions are correctly identified by our algorithm. In order to explicitly assess the advantage owned by the use of temporal information of our algorithm, we present also results derived from a simpler non temporal segmentation, obtained by segmenting the averaged image. In this case, after obtaining the mean image by averaging the gray level values of all the frames of the sequence, we applied a region growing algorithm, similar to that used in our algorithm. Results are shown in Fig. 8.2(b): it is evident that this method is not able to capture temporal diversity between the pixels of the regions, resulting in only five regions. To assess the robustness of our approach to noisy sequences we add two kinds of synthetic noise to the sequence: a Salt & Pepper noise, of intensity 0.05 and 0.25, and a white Gaussian noise, of

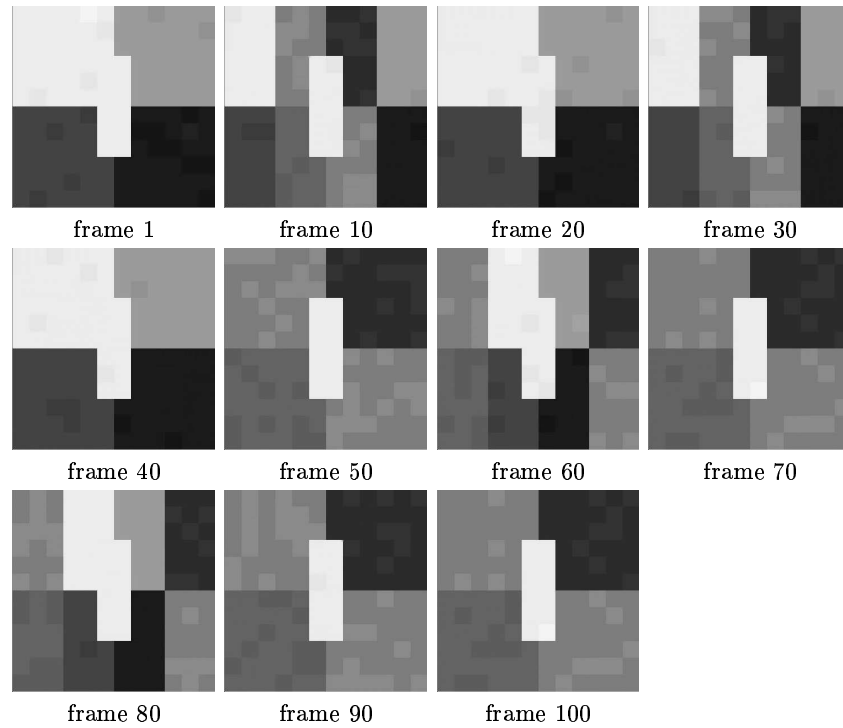


Fig. 8.1. Some frames of the synthetic sequence used for testing our algorithm.

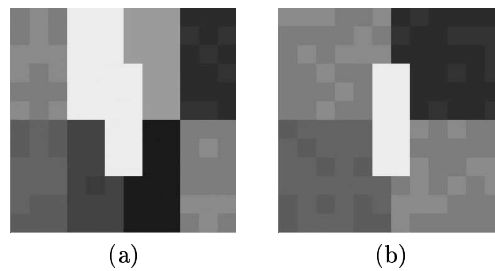


Fig. 8.2. Segmented sequence obtained by (a) the proposed approach (b) a region growing method onto the averaged image.

variance 0.01. An example of such a noisy frame and the corresponding sequence segmentation are presented in Fig. 8.3, for all noise situations. It is clear that our approach is quite robust to recover from that noise; actually, even if the frame sequence is quite corrupted, the different semantic regions are identified quite well.

The proposed approach is also tested on some real cases, with indoor and outdoor surveillance sequences. The first two, obtained from [204], regard the monitoring of an indoor environments with one moving object. The sequences are formed by 160 and 106 frames respectively, acquired at 20 frame/sec. Some of the

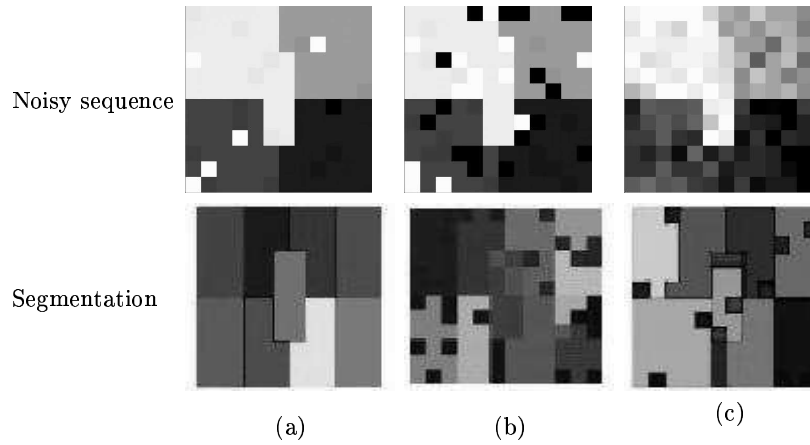


Fig. 8.3. Synthetic experiment with noise: in the left column a frame from noisy sequence, in the right column the resulting segmentation, for different noise type and level: (a) Salt & Pepper 0.05, (b) Salt & Pepper 0.25, and (c) Gaussian with 0.01 variance.

frames of the two sequences are presented in Fig. 8.4 and Fig. 8.6, showing a sudden not uniformly distributed change of the illumination. The non uniform change of

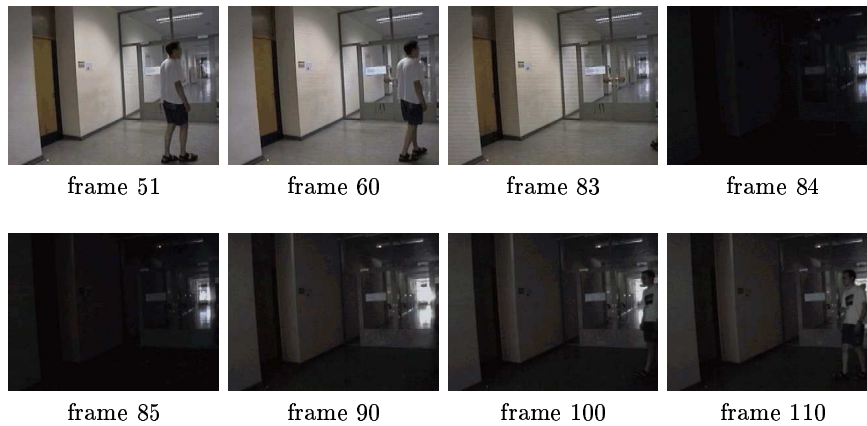


Fig. 8.4. Frames of the indoor sequence 1.

the luminosity could drastically affect the comprehension of the sequence, and only a method using spatio-temporal information, like the one proposed here, is able to correctly identify the semantic separated regions of the scene. To maintain a reasonable computational effort, we partitioned the field of view in a grid with circular Gaussian filters of 5×5 pixels: at each time step each filter provides one single weighted value. The results of the segmentations, obtained after the HMM training using these values, are reported in Fig. 8.5 and Fig. 8.7, showing

the goodness of the segmentations. The proposed approach is tested on another

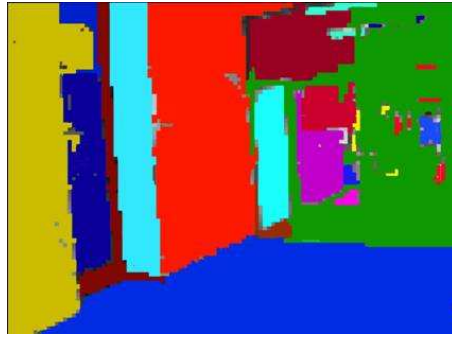


Fig. 8.5. Segmentation of the indoor sequence 1 with the proposed approach.



Fig. 8.6. Frames of the indoor sequence 2.

sequence, consisting in two moving objects in an outdoor scene. Few frames of the sequence are presented in Fig. 8.8. The resulting segmentation is proposed in Fig. 8.9(a): the segmentation is clear, expressive, and quite accurate. It is worthwhile to note that this segmentation is obtained by processing the whole sequence, without any need to remove the moving objects, which are in fact naturally removed by the procedure used to compute the distance. These results were compared to that obtained by the JSEG algorithm² [58], and is presented in Fig. 8.9(b). The result

² Result obtained with the no more available demo at the JSEG web site <http://vision.ece.ucsb.edu/segmentation/JSEG/index.html>.



Fig. 8.7. Segmentation of the indoor sequence 2 with the proposed approach.



Fig. 8.8. Few frames from the outdoor sequence.

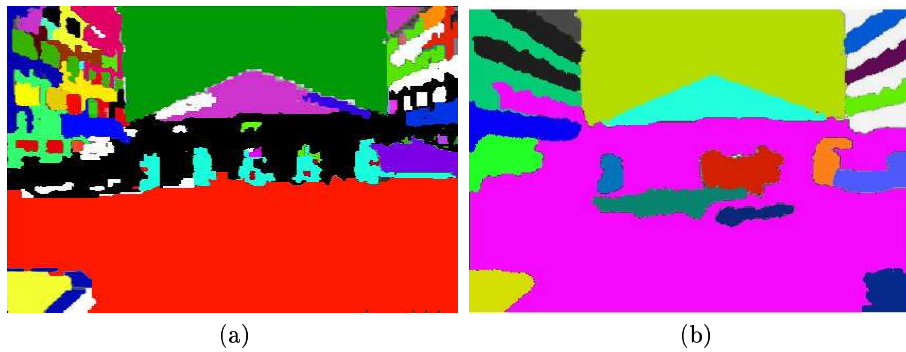


Fig. 8.9. Segmentation of the outdoor sequence:(a) proposed approach, (b) Jseg method [60]

of the proposed approach seems to be more accurate, in particular with regards to the ground in front of the scooters and the segmentation of the windows of the two lateral buildings.

8.4 Application to background modelling

In this section a direct application of this spatio-temporal segmentation is presented. In particular, it will be shown that such a compact representation of the background could drastically improve the robustness of standard background modelling methods with respect to sudden changes, as, for example, no-global illumination variations.

8.4.1 The background modelling problem

A videosurveillance system typically contemplates the monitoring of a site for long periods, using a static camera: the goal is to detect and classify moving objects (*foreground*) from the static scene (*background*). A fundamental issue to be solved is therefore the modelling of the background. Methods employing Time-Adaptive, Per-Pixel, Mixture Of Gaussian (TAPPMOG) have recently become a popular choice for modelling the background [203,93]. By way of these methods, the time evolution of each pixel is considered as a spatial independent process, modelled using a mixture of Gaussians. Each mixture is updated as new observations arrive, while the importance of older observations decays. A subset of Gaussians are considered as background at each time step and for each pixel, and current observations that do not match this distribution are labelled as foreground. The attractive properties of TAPPMOG method are various: first of all, it is able to slowly adapt itself to persistent scene appearance modifications, like the relocation of a background object; secondly, it is quite effective in modelling relatively simple but largely repetitive scene appearance changes associated with dynamic objects, like moving foliage; thirdly, it is suitable for real-time implementation.

Nonetheless, this technique presents some drawbacks. For example, the assignment of a pixel to the background or to the foreground is based on a threshold on the Gaussian mixture, that has to be fixed *a priori*. Another problem is that the said technique considers each pixel as an independent process without any use of spatial information or, more generally, higher-level information. This problem has been recently addressed by Harville in [93], where positive and negative feedbacks from higher level models have been used to guide low level pixel processes. Moreover, the choice of the learning rate, that determines the “speed” of the self adaptation of the TAPPMOG method to variations of the background, is critical. A high learning rate allows to adapt rapidly to illumination changes, but does not permit the detection of slowly moving objects, or accentuates the foreground aperture phenomena (*i.e.*, when an uniformly colored object moves, internal pixels could not be detected as foreground [212]). On the other hand, a low learning rate permits only slow adaptation, hence in case of a sudden change of the background the model finds numerous false foreground points for several frames until adaptation is completed.

The sensitivity to sudden changes of the illumination of the scene is another delicate issue. This is one of the most severe problems to be solved by a background modelling system, especially if changes are local and not uniformly distributed over the scene. Actually, a variation on the illumination of the whole scene could be detected and recovered with a standard histogram normalization technique,

whereas local variations could not be detected with such a global analysis. This situation can be very frequent in indoor situations, for example when the door of a lit room is opened in a monitored dark corridor. A substantial contribution in this sense was provided by Stenger *et al.* in [204], where a topology free Hidden Markov Model was used in order to model illumination changes of the scene. Even if results are promising, this method does not work on-line, and illumination changes have to be pre-classified off-line. Another interesting approach was proposed by Ohta in [164], where the possible changes in illumination are coded explicitly in a mathematical model. Nevertheless, the effectiveness of the method depends on the number of background prototypes estimated, failing in case of unexpected illumination changes.

A novel approach is here proposed, which is able to deal with sudden variations of illumination in the scene, also restricted to partial areas of it. We start from a generic TAPPMOG method like that proposed by Stauffer and Grimson [203]. The basic idea of our approach is that this process can be improved if we consider also a sort of region-based modelling, *i.e.*, considering the spatial information as provided by a background segmentation. In this case, the segmentation is obtained using the spatio-temporal method previously proposed, able to provide a really effective segmentation. With high probability, a change in illumination, both global and restricted to a particular area, results in a variation of the gray-level values of most of the pixels of the regions in that zone. In other words, if all pixels of a region significantly vary simultaneously, a typical system will tend to identify them as foreground, but, if the region is large enough, this situation can very likely be due to an illumination change rather than actual foreground.

Our approach uses spatial information resulting from the spatio-temporal segmentation of the background as prior in order to modulate the response of a TAPPMOG system. In particular, a variation of the learning parameter of the system is devised in order to efficiently cope with sudden changes in the background appearance.

Subsequently, this approach is naturally integrated in a probabilistic Bayesian framework, the particle filtering [102, 61] paradigm for tracking. This Monte Carlo technique [61], that has recently received growing attention, is based on sequential importance sampling/resampling. It provides a sound statistical framework for propagating sample-based approximations of posterior distributions, with almost no restriction on the ingredients of the model. We will show how a TAPPMOG model can be naturally inserted in this framework, eliminating the mixture threshold problem discussed above. We will also show, on real sequences available in the literature [204], that the use of spatial information is able to correctly manage sudden changes of illumination, even if restricted to local scene areas.

8.4.2 The TAPPMOG background modelling

In this subsection, a standard time adaptive per-pixel mixture of Gaussians background modelling scheme is presented, following [203]. By the way of this approach, a mixture of Gaussians is associated to each pixel, modelling the evolution of its gray level during time. The probability to observe the value $z_{uv}^{(t)}$, *i.e.* the intensity gray level of the pixel (u, v) of the image at time t , is given by:

$$P(z_{uv}^{(t)}) = \sum_{j=1}^K w_{j,uv}^{(t)} \mathcal{N}\left(z_{uv}^{(t)} | \mu_{j,uv}^{(t)}, \sigma_{j,uv}^{(t)}\right) \quad (8.4)$$

where $w_{j,uv}^{(t)}$, $\mu_{j,uv}^{(t)}$ and $\sigma_{j,uv}^{(t)}$ are the mixing coefficients, the mean and the standard deviation of the j -th Gaussian of the mixture of the pixel (u, v) at time t , respectively. The background modelling algorithm proceeds as follows. Suppose that, at each time instant, the Gaussians in a mixture are ranked in descending order by the value of w/σ . Every new pixel value is checked against the existing K Gaussian functions until a match is found, where a success match is defined as a pixel value within 2.5σ of any mode of the distribution. If none of the K Gaussian functions matches the pixel value, the least probable function is replaced with a new one, whose mean is equal to the current value, high variance, and low mixing coefficient. If j_{hit} is the Gaussian component matched, a pixel $z_{uv}^{(t)}$ is labelled as foreground if

$$\sum_{j=1}^{j_{hit}} w_{j,uv}^{(t)} > T \quad (8.5)$$

where T is a threshold (to be defined *a priori*) that indicates the minimum portion of the data that should be accounted for by the background.

Each mixture evolves during time, as new evidence arrives. The adaption is driven by the following rules. For the mixing coefficients:

$$w_{j,uv}^{(t)} = (1 - \alpha)w_{j,uv}^{(t-1)} + \alpha M_{uv}^{(t)}, 0 \leq j \leq K \quad (8.6)$$

where $M_{uv}^{(t)}$ is 1 for the matched Gaussian and 0 for the others, and α is the learning rate. Low α values imply a slow adaption, and vice versa. Parameters μ and σ remain the same for unmatched Gaussians, but, for the matched Gaussian function j_{hit} , we have (omitting indexes for clarity):

$$\mu^{(t)} = (1 - \rho)\mu^{(t-1)} + \rho z^{(t)} \quad (8.7)$$

$$\begin{aligned} \sigma^2(t) &= (1 - \rho)\sigma^2(t-1) \\ &+ \rho \left(z^{(t)} - \mu^{(t)}\right)^T \left(z^{(t)} - \mu^{(t)}\right) \end{aligned} \quad (8.8)$$

where $\rho = \alpha \mathcal{N}\left(z^{(t)} | \mu^{(t)}, \sigma^{(t)}\right)$.

8.4.3 The particle filtering tracker

A comprehensive description of this approach is out of the scope of this work, so it is not presented here: only the general ideas are introduced, mainly to setup the notation. Interested readers may refer to [102, 61, 103].

The particle filtering is a Bayesian approach: let Z^t be the image of the sequence at time t , and X^t be the model of the moving objects (foreground) in the scene at time t . This approach assumes that all information obtainable from the image Z^t about the model X^t is encoded in the posterior distribution $P(X^t | Z^t)$. This probability is approximated using a set of samples $\{s_{(\ell)}^t, p_{(\ell)}^t\}$, where each sample

represents an instance of the model X^t with a probability $p_{(\ell)}^t$ (also called *weight*). The algorithm, in its general formulation, follows a set of rules for propagating this set of samples over time. Basically, at each time instant t , the following steps are performed:

- *Sampling from prior (the posterior of step $t - 1$)*. L samples are chosen from $\{s_{(\ell)}^{t-1}\}$ with probability $\{p_{(\ell)}^{t-1}\}$, obtaining $\{\hat{s}_{(\ell)}^{t-1}\}$: the higher the weight $p_{(\ell)}^{t-1}$ at time $t-1$, the larger the probability of $s_{(\ell)}^{t-1}$ to “survive”.
- *Prediction*. Samples $\{s_{(\ell)}^t\}$ for time t are then obtained by applying a dynamic step to $\{\hat{s}_{(\ell)}^{t-1}\}$, predicting the new configurations. This prediction is based on previous values and on some *a priori* knowledge about the possible movements of the objects; typically, this dynamics also contains a stochastic component. This step is highly application-dependent.
- *Weighting*. Samples obtained by previous step are then weighted using the evidence $P(Z^t|X^t)$ (also called *likelihood*) from the image Z^t ; at each sample $s_{(\ell)}^t$ is then assigned the weight $p_{(\ell)}^t$, computed as $p_{(\ell)}^t = P(Z^t|X^t = s_{(\ell)}^t)$.

At each time step t , the estimated model X^t could be obtained with a Maximum A posteriori Probability (MAP) approach, *i.e.* by choosing the most probable sample.

In our approach, we used, as elementary image entity, the response of a circular Gaussian filter of mean 0 and variance 1, instead of the pixel. These filters are spaced in the image every 5 pixels, and are partially overlapped. The set of the responses of each filter at time t yields the “image” $Z^t = \{z_n^{(t)}\}$.

The definition of the sample $s_{(\ell)}^t$ follows the idea of multiple blob tracker proposed in [103]: $s_{(\ell)}^t$ is a configuration $(m_{\ell}^t, x_{\ell,1}^t, x_{\ell,2}^t, \dots, x_{\ell,m_{\ell}^t}^t)$, where m_{ℓ}^t is the number of objects, and $\{x_{\ell,i}^t\}$ are the positions of the objects in the scene. Each object is simply described with a vertically oriented ellipse, centered on $x_{\ell,p}^t$, denoted as $\mathcal{E}(x_{\ell,p}^t)$. The dynamics (second step of the algorithm) operates on the samples by predicting not only the objects’ positions, but also the number of objects. In this way, the system is also able to track more than one object, managing also entities which are entering in or exiting from the scene. Finally, the likelihood of a configuration $s_{(\ell)}^t$ is computed starting from the background response $L(z_n^{(t)}) = P(z_n^{(t)} \in FG)$, that represents the probability that the filter n is foreground at time t . The likelihood is zero for the configurations in which not all ellipses are covered by a sufficient foreground. By way of this, all configurations that predict an object in a position where no objects are actually present will be automatically discarded. For the others, the likelihood is computed as:

$$P(Z^t|X^t = s_{(\ell)}^t) = \frac{1}{k} \left(\left(\sum_p \sum_{n \in \mathcal{E}(x_{\ell,p}^t)} L(z_n^{(t)}) \right) - \sum_{n \notin \mathcal{E}(x_{\ell,p}^t)} L(z_n^{(t)}) \right) \quad (8.9)$$

where k is a normalization constant. In other words, a positive contribution to the likelihood of the sample $s_{(\ell)}^t$ derives from filters “covered” by the objects of $s_{(\ell)}^t$, whereas the others filters contribute negatively. By way of this, the configurations

that correctly predict both the positions and the number of objects in the scene have higher likelihood than configurations that correctly predict the positions of a minor number of objects only.

8.4.4 The integrated region- and pixel-based approach

In this section, the proposed approach is detailed: first, we describe how a TAPPMOG-based system is extended in order to naturally incorporate spatial information and to be encapsulated in the particle filtering framework; secondly, we explain the strategy that uses region-based information to modulate the pixel-based response, in order to obtain the background response L needed by the tracking algorithm.

The starting point is a spatial segmentation of the background scene, obtained with the method proposed in the previous section. The spatio-temporal segmentation is determined by using only the first fragment of the sequence: it is important to note that the method does not need any preprocessing step, as it is able to remove automatically any possible moving object.

The segmented image is defined as $R = \{R_i\}$, $1 \leq i \leq M$, and $R_i = \{R_i^1 \dots R_i^{|R_i|}\}$, where $|R_i|$ is the size of region R_i and R_i^n is the n -th filter of the region R_i . We denote as $z_n^{(i,t)}$ the observation of the n -th filter of the i -th region at time t .

The unmodulated pixel-level background response $L(z_n^{(i,t)})$ is naturally obtained by computing

$$L(z_n^{(i,t)}) = P\left(z_n^{(i,t)} \in \text{FG}\right) = \sum_{j=1}^{j_{hit}} w_{j,n}^{(i,t)} \quad (8.10)$$

representing the probability that $z_n^{(i,t)}$ is foreground, which is assigned by the TAPPMOG model, *i.e.* before high-level modulation. The weights $w_{j,n}^{(i,t)}$ are mixing coefficients related to the j -th Gaussian of the mixture corresponding to the n -th filter of the i -th region, at time t . It is worth noting that by means of this the threshold T , present in the Eq. (8.5), is not required anymore. The tracking algorithm uses all information embedded in Eq. (8.10), without any loss derived from the thresholding approximation.

Subsequently, the spatial information derived from segmentation is used to modulate the low-level response, varying the learning parameter α in order to allow the system to rapidly evolve in case of sudden change of the background. The idea is to “accelerate”, when needed, the process of adaptiveness of the low level model. With a sudden change in illumination, for example, most part of the pixels of the interested region changes suddenly, thus obtaining a wrong high probability to be foreground. Monitoring these sudden changes, we can adapt learning parameters in order to recover from these situations. To do that, we define for each region R_i the *approximate filling coefficient* $\gamma_i^{(t)}$, that represents the probability, assigned by the low level model, that a region R_i is foreground:

$$\gamma_i^{(t)} = \frac{\sum_{n=1}^{|R_i|} L(z_n^{(i,t)})}{|R_i|} \quad (8.11)$$

We define also the *modulated filling coefficient* $\hat{\gamma}_i^{(t)}$ in the same manner, using only $\hat{L}(z_n^{(i,t)})$ instead of $L(z_n^{(i,t)})$. $\hat{L}(z_n^{(i,t)})$ represents the final estimate, after modulation, of the probability of being foreground of the n -th filter of the region R_i . The computation of this quantity is described later in this section.

Instead of having a fixed learning parameter α , at each time step t we define a set of *time-varying* learning parameters $\alpha_i^{(t)}$, one for each region R_i . These coefficients are computed with the following formula:

$$\alpha_i^{(t)} = \max \left(\alpha, \left| \gamma_i^{(t)} - \hat{\gamma}_i^{(t-1)} \right| \right) \quad (8.12)$$

where α is the TAPPMOG learning parameter of formulas (8.6) and (8.7): this was fixed to 0.7, value that permits to detect also relatively slowly moving objects.

The quantity $\left| \gamma_i^{(t)} - \hat{\gamma}_i^{(t-1)} \right|$ represents a measure of how much part of the region R_i is changed from step $t-1$ to step t . If this quantity is low, the low-level model does not need any rectifications or adjustments. On the contrary, when this quantity is high, a large part of the region R_i has changed rapidly, and, if the regions are sufficiently larger than the foreground, this rapid change can likely be due to an illumination variation. If this is the case, the background model must adapt very fast to this new situation, hence the learning parameter should be increased. Moreover, this upsurge of the speed of adaptiveness is not *a priori* fixed, but depends on the rapidity and the globality of the background change.

The increasing of the adaptiveness speed means that, in the update of the parameters, most of the importance is given to the last observation (the one of the illumination change), forcing it to become rapidly one of the background Gaussians. This is correct if the whole region is background, but, if foreground is present during the change, this update is indeed wrong. In this case, the algorithm sets as background what is actually foreground, losing the foreground in the scene. This is solved by using the value $\hat{z}^{(i,t)}$ instead of $z_n^{(i,t)}$ in the updating parameter equations (Eq. (8.6) and (8.7)). This value is the weighted average of the observations $z_n^{(i,t)}$ of the filters of the region R_i , each weighted by its probability to be background at time step $t-1$, *i.e.*,

$$\hat{z}^{(i,t)} = \frac{1}{k} \sum_{n=1}^{|R_i|} (1 - \hat{L}(z_n^{(i,t-1)})) z_n^{(i,t)} \quad (8.13)$$

where k is a normalization constant. By way of this, the system is able to detect the foreground also after the re-parameterization of the background model. The use of this averaged region-based value to update the model, instead of using pixel-based (or filter-based) value, is actually reasonable in that the segmentation used as prior knowledge determines regions of gray-level similarity. Consequently, the substitution of each value in the region with the averaged region value, results in an approximation, indeed sufficient to recover from illumination change situations. This region-based approximation is then refined in a couple of frames by the usual time-adaptation of the TAPPMOG pixel-based process.

If the learning parameter $\alpha_i^{(t)}$ has changed, the mixture parameters of the whole region are adjusted accordingly. From the update, we obtain new mixture parameters' estimates, $\hat{w}_{j,n}^{(i,t)}$, $\hat{\mu}_{j,n}^{(i,t)}$, $\hat{\sigma}_{j,n}^{(i,t)}$, and we re-compute the likelihood \hat{L} , allowing

an immediate correction and recovery from illumination changes. The final modulated background response \hat{L} , used by the tracker (Eq. (8.9)), is finally computed as:

$$\hat{L}(z_n^{(i,t)}) = \begin{cases} L(z_n^{(i,t)}) & \text{if } \alpha_i^{(t)} = \alpha \\ \sum_{j=1}^{j_{hit}} \hat{w}_{j,n}^{(i,t)} & \text{otherwise} \end{cases} \quad (8.14)$$

8.4.5 Results

This approach was tested on the same two indoor sequences presented in Section 8.3: their main characteristic is a sudden non uniform illumination change, occurring in the middle of the sequence.

Some frames of the first sequence are presented in Fig. 8.4, showing the illumination change, occurring at frames 83-84. The initial spatial segmentation used in this experiment is shown in Fig. 8.5 of the previous section. In Fig. 8.10(a), a comparison between standard TAPPMOG method (as in [203]) and the proposed approach is presented (white pixels represent the foreground). We can notice that, in correspondence of the sudden change of illumination (frames 83-84), the TAPPMOG method identifies almost all pixels in the scene as foreground. This is obvious, as the per pixel process recognizes only the pixel gray level variation. With our approach, the use of the spatial high-level information permits the detection of the globality of the change, recovering in real time the correct background. We can also notice that when the foreground object actually comes in again in the scene at frame 100, our approach succeeds in distinguishing it, whereas the TAPPMOG method succeeds in discriminating it after a certain latency, only at frame 112. More precisely, the TAPPMOG approach needs 28 frames to adapt to the change of illumination, whereas in the proposed approach the recover is immediate. This is confirmed by results obtained applying tracking procedure, proposed in Fig. 8.10(b). We could notice that, before the illumination change, the object (identified by the ellipse) is correctly tracked by both methods. After the change, the background response given by the TAPPMOG model is not-informative, and the tracker identifies several false foreground objects. With our approach, the response of the background model is instead correct, and the object is correctly tracked.

The same conclusions could be drawn for the second sequence analysis: some frames of the examined sequences are presented in Fig. 8.6, while the segmentation used by the approach is shown in Fig. 8.7, both in the previous section. Also in this case there is a sudden illumination change in the middle of the sequence, that drastically affects the sequence understanding. In Fig. 8.11(a) and (b) results obtained with the standard TAPPMOG background modelling and the proposed approach are presented, together with the corresponding tracking results. Also in this case one can notice the improvement obtained by the use of the proposed approach, that allows the tracking system not to lose the object after the illumination change. In the standard approach, on the contrary, several false objects are detected.

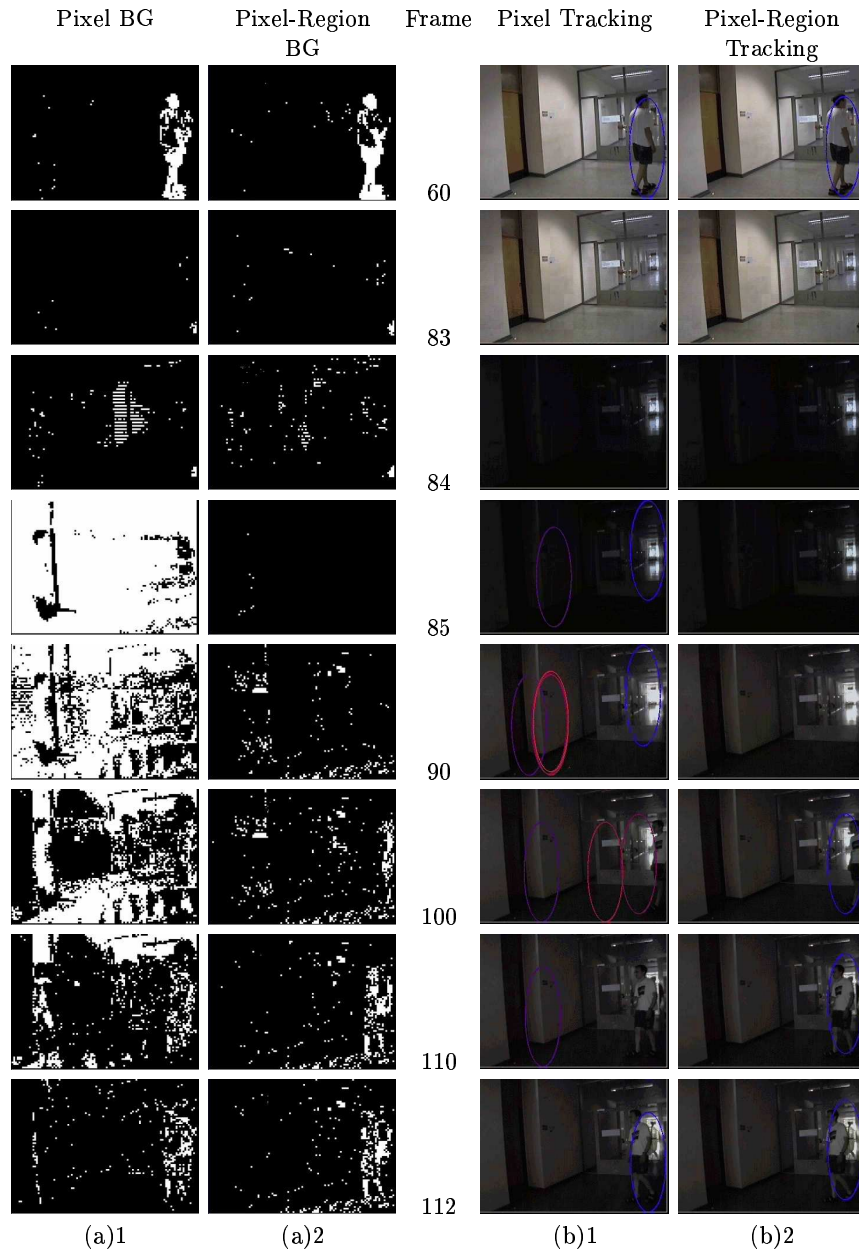


Fig. 8.10. Response of the background model on sequence 1, with the corresponding tracking: (a) Background modelling: (a1) standard TAPPMOG model; (a2) the proposed approach; (b) the corresponding tracking.

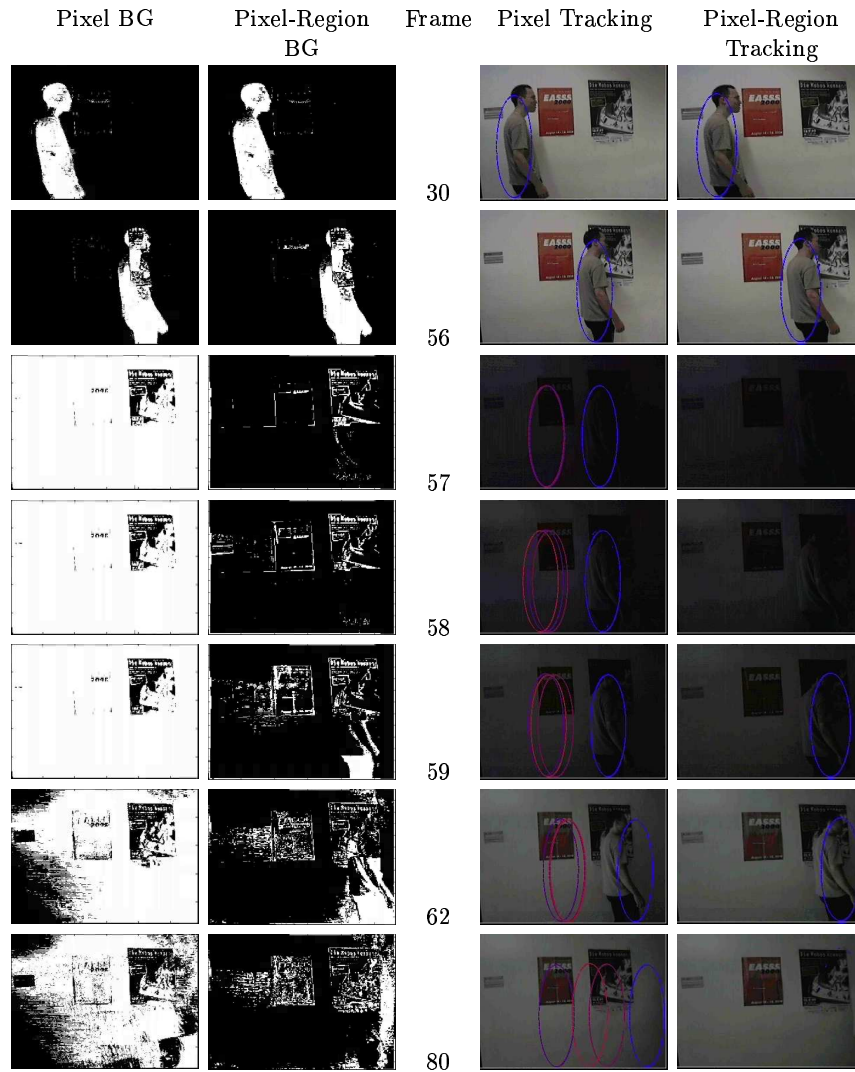


Fig. 8.11. Response of the background model on sequence 1, with the corresponding tracking: (a) Background modelling: (a1) standard TAPPMOG model; (a2) the proposed approach; (b) the corresponding tracking.

8.5 Conclusions

In this chapter a novel method for spatio-temporal segmentation is proposed, able to discover regions of spatial, chromatic and temporal homogeneity. This method is based on the clustering of a forest of Hidden Markov Models, each one modelling the temporal evolution of each scene location. A novel similarity measure between HMMs has been introduced, able to characterize stationary components of the sequence, assigning less importance to not determinant components, as

moving objects. The proposed approach has been tested with synthetic and real sequences, showing promising performances. Moreover it has been shown that such a segmentation could be effectively integrated in a background modelling context. The idea is to use this kind of higher level region-based information for modulating the pixel-based information given by a standard TAPPMOG approach. This modulation results in a variation of the adaptiveness speed of the background modelling system driven by region-based reasoning. Experimental results have shown that this approach seems able to effectively recover from sudden changes in the illumination of the scene.

Conclusions

This thesis could be collocated in the statistical Pattern Recognition context, and it concerns the analysis and the investigation of the Hidden Markov Model approach. This approach represents a widely employed statistical tool for probabilistic modelling of sequential data, which has assumed a great importance in the last decade. In this thesis, Hidden Markov Models have been addressed in a twofold way: from a methodological perspective, by investigating and analyzing some open issues related to the methodology itself, and from an application perspective, by applying this approach to some Computer Vision and Pattern Recognition problems.

Several contributions have been proposed in this thesis, with respect to both methodological issues and applications. In the former case, the main contributes lie in the contexts of model selection, classification and clustering with HMM. Regarding the model selection issue, a formal proof of the equivalence between continuous Gaussian HMMs has been proposed, able to reduce, in the continuous case, the space in which the best model has to be searched for. Three original methods have been also proposed, each characterized by different features, but all capable of determining the more appropriate model structure from the training data. The first one is linked to the initialization issue, and results in a very fast technique. The second one makes use of a syntactic equivalence relation, the probabilistic bisimulation [10], and is aimed at reducing an oversized model to a more compact representation. The more interesting is the third one, able to directly address both the model selection and the initialization issue. The key idea is to use a decreasing learning strategy, starting each training session from an informative situation derived from the previous training phase. More specifically, the proposed procedure consists of starting the model training using a large number of states, run the estimation algorithm, and, after convergence, evaluate a pre-chosen model selection criterion for that model. Then, the “least probable” state is pruned, and this configuration is taken as initial situation from which to start again the training procedure. In this way, each training session is started from a “nearly good” estimate, reducing the impact of the initialization problem. Moreover, the “good” initialization drastically reduces the number of iterations required by the learning algorithm, resulting in a less computational demanding procedure.

In the classification context, some considerations on the reliability of the standard classification scheme have been presented. An alternative classification

scheme has been then introduced, inspired by the similarity-based classification paradigm. This scheme is able to use all the information available from the evaluation process, building a new representation space, which has been shown to be really discriminant: the classification in that space results in a sensible improvement of the classification accuracy, obviously relatively to the data set investigated. Further investigations will regard the choice of the prototypes used for build the feature space, representing a crucial aspect that determines the dimensionality of that space.

Finally, with respect to the clustering problem, some contributions have been proposed in the context of the standard method, aimed at obtaining a more effective clustering. Experimental evaluations on a EEG clustering problem have shown promising results. Subsequently an alternative scheme has been presented, founded on the similarity-based representation introduced in the classification context. The method is able to notably enhance the clustering results on both synthetic and real experiments. The problem of clustering sequential data is a challenging problem in Pattern Recognition, due to its intrinsic higher difficulty if compared with supervised classification. This context has also grown in practical importance in last years, due to its applicability in emergent application domains, as bioinformatics (modelling of DNA strings) or data mining.

All of the proposed methodological approaches have been evaluated by using synthetic and real experiments, regarding 2D shape classification, face recognition, DNA modelling, EEG segmentation, and video analysis. With respect to the investigated problems, the proposed strategies have shown promising results. Obviously, it is not possible to assess the definitive superiority of the proposed approaches with respect to the state of the art, since only few problems have been addressed, and Pattern Recognition is an ill-posed problem. This means that, except for special cases, definitive conclusions could never be drawn, even if a lot of experiments have been carried out [66,96,67]. For some of the applications investigated, the use of HMM-based approaches has produced a surplus also in the application context. In particular, in the 2D shape classification problem, a very robust system has been introduced, that correctly manages object translations, rotations, occlusions, affine projections and noise. It is interesting to note that each HMM is trained using *only* one aspect of the object, without including in the training phase any object variation, so that the proposed encouraging results are obtained by training a single shape.

In the face recognition context, a really effective technique has been developed, able to outperform all other methods present in the literature on the same standard database. This problem is not a classical “sequence” classification problem, since the sequence has been forcedly extracted from the face; nevertheless, HMMs are very useful and effective also in this case.

Finally, for the video sequence analysis, a HMM-based approach has been introduced, able to subdivide the background of a sequence into regions of chromatic, temporal and spatial homogeneity. A novel measure of similarity between sequences has been developed, able to remove non stationary elements from the sequence. The obtained spatial representation has been profitably integrated in a background modelling system, that effectively recovers from sudden illumination changes in the scene.

At the end of this thesis, some general considerations about the Hidden Markov Model methodology could be done. In my opinion, this technique is a very versatile and accurate probabilistic tool for sequential data modelling, able to effectively manage noisy situations and missing data. It has been shown that an HMM trained on a sequence \mathbf{O} is able to correctly identify it also in presence of shifting, subsampling, oversampling or fragmenting. Moreover, this technique is useful not only for sequential data modelling, but also in some not sequential situations. Some applications, as face recognition, gain several advantages when using Hidden Markov Models, even if the problem is not “sequential”, and the sequence has to be forcedly determined from the data. Experimental evaluations have shown that model selection and initialization are crucial issues, fundamental for obtaining a correct and effective modelling, and that great advantages could be obtained in practical applications if these issues are taken into consideration.

On the other hand, I think that HMMs are not very suited for data generation. It has been experimentally found that data generated by a Hidden Markov Model seldom resemble to the original data used to train the model. This is likely due to the not-stationarity of the transition matrix of the employed model, that contains information about changes between states, but not about the time instant when these changes occur.

With regards to the future perspectives, I think that the model selection issue is not a completely solved problem, and some other research has to be developed in this context. In my opinion, the next methodological issue to be raised is the integration of this technique with other models, to determine more complex structures. Some efforts in this direction have been produced by Brand *et al.* [34], who propose Coupled Hidden Markov Models, by Fine *et al.* [76], who introduce Hierarchical Hidden Markov Models, and by Bengio [20], who introduces hybrids models involving neural networks.

In conclusion, the optimal results obtained by HMM approaches proposed in this thesis, together with the methodological innovations introduced, confirm the effectiveness and the wide applicability of the Hidden Markov Model approach to complex real application problems. Since the goal of the Pattern Recognition is to resolve problems of great practical relevance, the importance of HMM in this context is large, and justifies the efforts done in this thesis.

Appendix

Linear dimensionality reduction techniques

A central problem in Pattern Recognition research area is finding a suitable representation of multivariate data. The goal is to reduce the dimensionality of the original data, in order to prevent or reduce the impact of the *curse of dimensionality* problem [30]. This situation occurs when the dimensionality of the problem space increases too much: if a limited quantity of data is available, as in practical applications, the increasing of the dimensionality of the space rapidly leads to situations in which the data are very sparse, and the space is almost empty. It is therefore important to find techniques able to reduce the dimensionality of the space, maintaining almost all “relevant” information. For computational and conceptual simplicity, this reduction is typically addressed by linear transformations of the original space, *i.e.*

$$\mathbf{y} = \mathbf{A}\mathbf{x} \tag{A.1}$$

where \mathbf{x} are the original data, \mathbf{y} are the reduced data, and \mathbf{A} is a matrix to be determined.

Several methods have been developed to find a suitable linear transformation; usually, these methods define a principle indicating which transform is optimal, and determine the matrix \mathbf{A} optimizing this criterion. In this section, three linear reduction techniques are reviewed: the Principal Component Analysis (PCA) [113], the Independent Component Analysis (ICA) [99], and the Fisher Discriminant Analysis (FDA) [81]. All these methods are aimed at reducing the dimensionality of a space while preserving almost all the “relevant information” contained in a data set. The concept of “relevant information” is different in these techniques. In PCA, the information to be preserved is the variance of the data, while ICA tries to find the most statistically independent directions. Both these techniques are unsupervised methods, that do not take into consideration eventual information regarding pattern labels. When labels are available, more appropriate techniques could be used, able to exploit the information derived from this supervised context. FDA represents a well-known example of such a technique, which looks for a low-dimensional projection that best preserves the class separability of the data.

A.1 Principal Component Analysis (PCA)

The Principal Component Analysis, also called, in some contexts, the Karhunen-Loève transform, is a linear reduction technique widely used in several application areas. Its main objective is to reduce the space dimensionality by maintaining the maximum adherence to the original data, in the mean-square sense: the optimal directions are those explaining the maximum amount of variance of the data. The derivation of the PCA transform is briefly presented.

First of all, please note that a vector in a generic d -dimensional space could be represented as a linear combination of d orthonormal vector \mathbf{u}_i :

$$\mathbf{x} = \sum_{i=1}^d z_i \mathbf{u}_i \quad (\text{A.2})$$

z_i could be obtained as

$$z_i = \mathbf{u}_i^T \mathbf{x} \quad (\text{A.3})$$

that implies a rotation of the coordinate system. Now suppose that, in order to reduce the dimensionality, only a subset M of the d vectors \mathbf{u}_i is retained. The approximated $\tilde{\mathbf{x}}$ is then

$$\tilde{\mathbf{x}} = \sum_{i=1}^M z_i \mathbf{u}_i + \sum_{i=M+1}^d b_i \mathbf{u}_i \quad (\text{A.4})$$

The error introduced in the approximation is

$$\mathbf{x} - \tilde{\mathbf{x}} = \sum_{i=M+1}^d (z_i - b_i) \mathbf{u}_i \quad (\text{A.5})$$

The best approximation is then the one that minimizes the sum of the squares of the errors on the whole data set. The goal is therefore to find the matrix \mathbf{A} that minimizes

$$E_M = \frac{1}{2} \sum_{n=1}^N \|\mathbf{x}^n - \tilde{\mathbf{x}}^n\|^2 = \frac{1}{2} \sum_{n=1}^N \sum_{i=M+1}^d (z_i^n - b_i)^2 \quad (\text{A.6})$$

It could be shown [30] that the minimum of E_M with respect to \mathbf{u}_i occurs when

$$\boldsymbol{\Sigma} \mathbf{u}_i = \lambda_i \mathbf{u}_i \quad (\text{A.7})$$

i.e., when vectors \mathbf{u}_i represent the eigenvectors (λ_i are the corresponding eigenvalues) of the covariance matrix $\boldsymbol{\Sigma}$ of the data. After some algebras, the error E_M could be written as

$$E_M = \frac{1}{2} \sum_{i=M+1}^d \lambda_i \quad (\text{A.8})$$

The minimum is then obtained by discarding the eigenvectors corresponding to the $d - M$ smallest eigenvalues.

In practice, the algorithm proceeds by the following steps: first the data are centered, *i.e.* the mean is subtracted from each element; then the covariance matrix of the data is computed, and its eigenvectors and eigenvalues are found. The eigenvectors corresponding to the M *largest* eigenvalues are retained; the input vectors are then projected to the retained eigenvectors, to obtain the coordinates z_i^n in the M -dimensional space.

A.2 Independent Component Analysis (ICA)

The Independent Component Analysis is a linear reduction technique whose importance has rapidly increased in recent years; it has typically been used in problems of blind source separation or feature extraction. As the name implies, the basic goal is to find a transformation in which the components are statistically as much independent as possible. In this thesis, the ICA transform has been used for feature extraction. This use is motivated by results in neuroscience, where it has been shown that the brain, in the early processing of sensory data, applies a similar principle of redundancy reduction.

In the literature, at least two different basic definition for linear ICA can be found [50, 118]: one for the *Noisy ICA model*, and one for the *Noise-free ICA model*. In this thesis we use the Noise-free ICA model, which is defined as [99]:

Definition A.1. *Independent Component Analysis of a random vector \mathbf{x} consists of estimating the following generative model for the data:*

$$\mathbf{x} = \mathbf{A}\mathbf{s} \quad (\text{A.9})$$

where the latent variables (components) s_i in the vector $\mathbf{s} = (s_1, \dots, s_n)^T$ are assumed independent, and the matrix \mathbf{A} is a constant “mixing” matrix.

The independent component analysis is usually performed in two steps: first, an objective function should be defined, able to quantify the independency of the components. Secondly, an efficient and effective algorithm to minimize or maximize this criterion should be derived.

With regards to the first step, several objective criterions have been proposed in the recent years: in this thesis a Maximum Likelihood approach [177] is used. This approach, after defining the likelihood of a noise-free ICA model, estimates the model by maximizing the likelihood. Denoting by $\mathbf{W} = (\mathbf{w}_1, \dots, \mathbf{w}_m)^T$ the matrix \mathbf{A}^{-1} , the log-likelihood takes the form [177]:

$$L = \sum_{n=1}^N \sum_{i=1}^m \log f_i(\mathbf{w}_i^T \mathbf{x}^n) + N \ln(\det(\mathbf{W})) \quad (\text{A.10})$$

where f_i are the density functions of the s_i (here assumed to be known), and $\{\mathbf{x}^n\}$ is the data set. This approach, under some conditions, is equivalent to the method based on the maximization of network entropy (Infomax) [19].

With regards to the optimization step, one can use any of the classical methods for optimizing the objective function, like (stochastic) gradient methods, Newton-like methods, etc.

In this thesis, the ICA model was estimated by using the toolbox obtained from [126], that uses the Maximum Likelihood approach [177] and, in the optimization step, the method proposed in [161].

A.3 Fisher Discriminant Analysis (FDA)

Principal Component Analysis and Independent Component Analysis are unsupervised techniques, since they produce a linear transformation without any use of the labels eventually present. If these labels are actually present, a supervised technique could instead be used. Fisher Discriminant Analysis represents an example of such a technique: the goal of this technique is to project data onto a space of lower dimensionality, trying to maintain the maximum class separability between items. In FDA, several criteria can be adopted to quantify the concept of “class separability” [81]. In this thesis we adopt the classical one proposed by Fisher [78]: given a problem in C classes, each one with N_k elements $\{\mathbf{x}_k^i\}$ ($\sum_k N_k = N$), the criterion to be maximized is defined as

$$J(\mathbf{A}) = \text{tr}\{(\mathbf{A}^T S_W \mathbf{A})^{-1} (\mathbf{A} S_B \mathbf{A}^T)\} \quad (\text{A.11})$$

where:

- S_W is the *within-class* covariance, defined as

$$S_W = \sum_{k=1}^C \left(\sum_{i=1}^{N_k} (\mathbf{x}_k^i - \mathbf{m}_k)(\mathbf{x}_k^i - \mathbf{m}_k)^T \right) \quad (\text{A.12})$$

where

$$\mathbf{m}_k = \frac{1}{N_k} \sum_{i=1}^{N_k} \mathbf{x}_k^i \quad (\text{A.13})$$

is the mean of the cluster k ;

- S_B is the *between-class* covariance, defined as

$$S_B = \sum_{k=1}^C N_k (\mathbf{m}_k - \mathbf{m})(\mathbf{m}_k - \mathbf{m})^T \quad (\text{A.14})$$

where

$$\mathbf{m} = \frac{1}{N} \sum_{k=1}^C N_k \mathbf{m}_k \quad (\text{A.15})$$

is the mean of the whole data set.

This criterion is properly defined, as it assumes large values when the covariance between class is large (*i.e.*, the clusters are well separated) and the covariance within class is low (*i.e.*, the clusters are compact).

It has been shown in [81] that the matrix \mathbf{A} that maximizes the criterion (A.11) is obtained by computing the eigenvectors and the eigenvalues of the matrix $S_W^{-1} S_B$. As in the Principal Component Analysis case, if the goal is to determine a space of dimensionality M , only the eigenvectors corresponding to the M largest eigenvalues should be retained, in order to maximize the class separability in the projected space.

References

1. B. Achermann and H. Bunke. Combination of face classifiers for person identification. In *Int. Conf. Pattern Recognition*, pages C416–C420, 1996.
2. P. Aczel. Non-well-founded sets. *Lecture Notes, Center for the Study of Language and Information*, 14, 1988.
3. O.E. Agazzi and S.-S. Kuo. Pseudo two-dimensional hidden Markov models for document recognition. *AT & T Technical Journal*, 72(5):60–72, 1993.
4. R. Agrawal, C. Faloutsos, and A. Swami. Efficient similarity search in sequence databases. In D. B. Lomet, editor, *Proc. Int. Conf. on Foundations of Data Organization and Algorithms*, pages 69–84. Springer-Verlag, 1993.
5. H. Akaike. A new look at the statistical model identification. *IEEE Trans. on Automatic Control*, AC-19:716–723, 1974.
6. C.W. Anderson, E.A. Stolz, and S. Shamsunder. Multivariate autoregressive models for classification of spontaneous electroencephalogram during mental tasks. *IEEE Transactions on Biomedical Engineering*, 45(3):277–286, 1998.
7. N. Arica and F.T. Yarman-Vural. A shape descriptor based on circular Hidden Markov Model. In *IEEE Proc. Int. Conf. Pattern Recognition*, volume 1, pages 924–927, 2000.
8. O. Ayinde and Y.H. Yang. Face recognition approach based on rank correlation of gabor-filtered images. *Pattern Recognition*, 35(6):1275–1289, 2002.
9. C. Bahlmann and H. Burkhardt. Measuring hmm similarity with the bayes probability of error and its application to online handwriting recognition. In *Proc. Int. Conf. Document Analysis and Recognition*, pages 406–411, 2001.
10. C. Baier, B. Engelen, and M. Majster-Cederbaum. Deciding bisimilarity and similarity for probabilistic processes. *Journal of Computer and System Sciences*, 60:187–231, 2000.
11. R. Bakis. Continuous speech word recognition via centisecond acoustic states. In *Proc. ASA Meeting*, Washington, DC, 1976.
12. G.H. Ball and D.J. Hall. A clustering technique for summarizing multivariate multivariate data. *Behavioral Science*, 12:153–155, 1967.
13. L.E. Baum. An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. *Inequality*, 3:1–8, 1970.
14. L.E. Baum and J.A. Egon. An inequality with applications to statistical estimation for probabilistic functions of a Markov process and to a model for ecology. *Bull. Amer. Meteorology Soc.*, 73:360–363, 1967.
15. L.E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state Markov chains. *Annals of Math. Statistics*, 37:1,554–1,563, 1966.

16. L.E. Baum, T.E. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Math. Statistics*, 41(1):164–171, 1970.
17. L.E. Baum and G.R. Sell. Growth functions for transformations on manifolds. *Pacific J. Math.*, 27(2):211–227, 1968.
18. P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, July 1997.
19. A. Bell and T.J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159, 1995.
20. Y. Bengio. Markovian models for sequential data. *Neural Computing Surveys*, 2:129–162, 1999. available at <http://www.icsi.berkeley.edu/~jagota/NCS>.
21. J. Van Benthem. *Modal Correspondence Theory*. PhD thesis, Universiteit van Amsterdam, Instituut voor Logica en Grondslagenonderzoek van Exacte Wetenschappen, 1978.
22. J. Berger. *Statistical decision theory and Bayesian analysis*. Springer-Verlag, New York, 1980.
23. J. Bernardo and A. Smith. *Bayesian Theory*. John Wiley & Sons, Chichester (UK), 1994.
24. M. Bicego, A. Dovier, and V. Murino. Designing the minimal structure of Hidden Markov Models by bisimulation. In M.A.T. Figueiredo, J. Zerubia, and A.K. Jain, editors, *Energy Minimization Methods in Computer Vision and Pattern Recognition*, LNCS 2134, pages 75–90. Springer, 2001.
25. M. Bicego and V. Murino. 2D shape recognition by Hidden Markov Models. In *IEEE Proc. of Int. Conf. on Image Analysis and Processing*, pages 20–24, 2001.
26. M. Bicego and V. Murino. Investigating Hidden Markov Models' capabilities in 2D shape classification. Submitted for publication to *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2002.
27. M. Bicego, V. Murino, and M.A.T. Figueiredo. A sequential pruning strategy for the selection of the number of states in Hidden Markov Models. *Pattern Recognition Letters*, in press, 2002.
28. M. Bicego, V. Murino, and M.A.T. Figueiredo. Similarity-based classification of sequences using hidden Markov models, 2002. submitted to *Pattern Recognition*.
29. A. Biem, Jin-Young Ha, and J. Subrahmonia. A bayesian model selection criterion for HMM topology optimization. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, volume 1, pages 989–992, 2002.
30. C.M. Bishop. *Neural Network for Pattern Recognition*. Clarendon Press - Oxford, 1995.
31. C.M. Bishop. Variational principal components. In *Proc. of IEEE Int. Conf. on Artificial Neural Networks*, volume 1, pages 509–514, 1999.
32. M. Borsotti, P. Campadelli, and R. Schettini. Quantitative evaluation of color image segmentation results. *Pattern Recognition letters*, 19(8):741–48, 1998.
33. M. Brand. An entropic estimator for structure discovery. In M.S. Kearns, S.A. Solla, and D.A. Cohn, editors, *Advances in Neural Information Processing Systems*, volume 11. MIT Press, 1999.
34. M. Brand, N. Oliver, and S. Pentland. Coupled hidden markov models for complex action recognition. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 1997.
35. M.E. Brand. Structure learning in conditional probability models via an entropic prior and parameter extinction. *Neural Computation Journal*, 11(5):1155–1182, 1999.
36. P. Brémaud. *Markov Chains*. Springer-Verlag, 1999.

37. I. Cadez, S. Gaffney, and P. Smyth. A general probabilistic framework for clustering individuals. In *Proc. of ACM SIGKDD 2000*, 2000.
38. J. Cai and Z.-Q. Liu. Hidden Markov Models with spectral features for 2D shape recognition. *IEEE Trans. Pattern Analysis Machine Intelligence*, 23(12):1454–1458, 2001.
39. A. Camproux, F. Saunier, and G. Thomas. A hidden Markov model approach to neuron firing patterns. *Biophysical journal*, 71(5):2404–2412, 1996.
40. J.F. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Analysis Machine Intelligence*, 8(6):679–698, 1986.
41. Olivier Cappé. Ten years of HMMs, 2001. Available at <http://www.tsi.enst.fr/cappe/docs/hmbib.html>.
42. K.R. Castleman. *Digital Image Processing*. Prentice Hall, 1996.
43. Y.T. Chan. *Wavelet Basics*. Kluwer Academic Publishers, Norwell, MA, 1995.
44. M.M. Chang, A.M. Tekalp, and M.I. Sezan. Simultaneous motion estimation and segmentation. *IEEE Trans. on Image Processing*, 6(9):1326–1333, 1997.
45. P. Cheeseman and J. Stutz. Bayesian classification (autoclass): Theory and results. In *Advances in Knowledge discovery and data mining*, pages 153–180, 1996.
46. R. Chellappa, C.L. Wilson, and S.A. Sirohey. Human and machine recognition of faces: a survey. *Proceedings of IEEE*, 83(5):705–740, 1995.
47. J.G. Choi, S.W. Lee, and S.D. Kim. Spatio-temporal video segmentation using a joint similarity measure. *IEEE Trans. on Circuits and Systems for Video Technology*, 7(2):279–86, 1997.
48. C.K. Chow. An optimum character recognition system using decision functions. *IRE Trans. on Electronic Computers*, 6:247–254, 1957.
49. C.K. Chow. On optimum error and reject trade-off. *IEEE Trans. on Information Theory*, 16:41–46, 1970.
50. P. Common. Independent component analysis - a new concept? *Signal Processing*, 36:287–314, 1994.
51. G. W. Cottrell and M. Fleming. Face recognition using unsupervised feature extraction. In *Proc. Int. Neural Network Conf.*, volume 1, pages 322–325, Paris, France, July 1990.
52. M. Cristani, M. Bicego, and V. Murino. Hidden markov models clustering for spatio-temporal segmentation of video sequences, 2002. submitted to CVPR03.
53. M. Cristani, M. Bicego, and V. Murino. Integrated region- and pixel-based approach to background modelling. In *Proc. of IEEE Workshop on Motion and Video Computing*, pages 3–8, 2002.
54. M.S. Crouse, R.D. Nowak, and R.G. Baraniuk. Wavelet-based statistical signal processing using hidden Markov models. *IEEE Trans. on Signal Processing*, 46(4):886–902, 1998.
55. I. Daubechies. *Ten Lectures on Wavelets*. SIAM, Philadelphia, PA, 1992.
56. D.L. Davies and D.W. Bouldin. A cluster separation measure. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1(2):224–227, 1979.
57. A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc. B*, 39:1–38, 1977.
58. Y. Deng and B.S. Manjunath. Unsupervised segmentation of color-texture regions in images and video. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(8):800–810, 2001.
59. Y. Deng, D. Mukherjee, and B.S. Manjunath. Netra-v: Toward an object-based video representation. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 202–215, 1998.
60. Yining Deng and B. S. Manjunath. Unsupervised segmentation of color-texture regions in images and video. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(8):800–810, 2001.

61. A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.
62. A. Dovier, C. Piazza, and A. Policriti. An efficient algorithm for computing bisimulation equivalence. Submitted for publication, 2002.
63. A. Dovier, C. Piazza, and A. Policriti. A fast bisimulation algorithm. In *Proc of 13th Conf. on Computer Aided Verification*, Paris, France, 2001.
64. S. Dubnov, R. El-Yaniv, Y. Gdalyahu, E. Schneidman, N. Tishby, and G. Yona. A new nonparametric pairwise clustering algorithm based on iterative estimation of distance profiles. *Machine Learning*, 47(1):35–61, 2002.
65. R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley and Sons, second edition, 2001.
66. R.P.W. Duin. A note on comparing classifiers. *Pattern Recognition Letters*, 17(5):529–536, 1996.
67. R.P.W. Duin. Compactness and complexity of pattern recognition problems. In C. Perneel, editor, *Proc. Int. Symposium on Pattern Recognition "In Memoriam Pierre Devijver"*, pages 124–128. Royal Military Academy, 1999.
68. R. Durbin, S. Eddy, A. Krogh, and G.J. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.
69. S. Eickeler, A. Kosmala, and G. Rigoll. Hidden Markov Model based continuous online gesture recognition. In *IEEE Proc. Int. Conf. Pattern Recognition*, volume 2, pages 1206–1208, 1998.
70. S. Eickeler, S. Mller, and G. Rigoll. Recognition of jpeg compressed face images based on statistical methods. *Image and Vision Computing*, 18:279–287, March 2000.
71. M. Falkhausen, H. Reininger, and D. Wolf. Calculation of distance measures between hidden markov models. In *Proc. of Eurospeech*, pages 1487–1490, 1995.
72. O. Faugeras. *Three-dimensional Computer Vision: a Geometric Viewpoint*. MIT Press, 1993.
73. M.A.T. Figueiredo and A.K. Jain. Unsupervised learning of finite mixture models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(3):381–396, 2002.
74. M.A.T. Figueiredo, J.M.N. Leitao, and A.K. Jain. On fitting mixture models. In E. Hancock and M. Pellilo, editors, *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 54–69. Springer Verlag, 1999.
75. M.A.T. Figueiredo, J.M.N. Leitao, and A.K. Jain. *Introduction to Bayesian Theory and Markov Random Fields for Image Analysis*. Springer, 2002. Under preparation.
76. S. Fine, Y. Singer, and N. Tishby. The hierarchical hidden markov model: Analysis and applications. *Machine Learning*, 32:41–62, 1998.
77. R. Fisher. On the mathematical foundations of theoretical statistics. *Philosophical Trans. of the Royal Society of London (A)*, 222:309–368, 1922.
78. R.A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936. Republished in *Contributions to Mathematical Statistics*, John Wiley, 1950.
79. G.D. Forney. The Viterbi algorithm. *Proc. of IEEE*, 61:268–278, 1973.
80. A.L.N. Fred, J.S. Marques, and P.M. Jorge. Hidden Markov Models vs. syntactic modeling in object recognition. In *IEEE Proc. Int Conf. Image Processing*, volume 1, pages 893–896, 1997.
81. K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, second edition, 1990.
82. G. Fumera, F. Roli, and G. Giacinto. Reject option with multiple thresholds. *Pattern Recognition*, 33(12), 2000.

83. Z. Ghahramani. An introduction to Hidden Markov Models and Bayesian Networks. *Int. Journal of Pattern Recognition and Artificial intelligence*, 15(1):9–42, 2001. Special Issue in Hidden Markov Models in Vision.
84. Z. Ghahramani and M.J. Beal. Graphical models and variational methods. In *Advanced mean field methods: theory and practice*. MIT Press, 2000.
85. G. Giacinto, F. Roli, and L. Bruzzone. "combination of neural and statistical algorithms for supervised classification of remote-sensing images". *Pattern Recognition Letters*, 21(5):385–397, 2000.
86. W.R. Gilks, S. Richardson, and D.J. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. London: Chapman & Hall, 1996.
87. T. Graepel, R. Herbrich, P. Bollmann-Sdorra, and K. Obermayer. Classification on pairwise proximity data. In D. Cohn M. Kearns, S. Solla, editor, *Advances in Neural Information Processing*, volume 11. MIT Press, 1999.
88. G. Guo, S. Z. Li, , and C. Kapluk. Face recognition by support vector machines. *Image and Vision Computing*, 19(9-10):631–638, 2001.
89. G. Halevi and D. Weinshall. Motion of disturbances: Detection and tracking of multi-body non-rigid motion, 1997.
90. Y.K. Ham and R.-H. Park. 3D object recognition in range images using hidden Markov models and neural networks. *Pattern Recognition*, 32(5):729–742, 1999.
91. J. Hamaker, A. Ganapathiraju, and J. Picone. Information theoretic approaches to model selection. In *Proc. of Int. Conf. on Spoken Language Processing*, Sydney, Australia, 1998.
92. B. Hannaford and P. Lee. Hidden Markov model analysis of force/torque information in telemanipulation. *International Journal of Robotics Research*, 10(5):528–539, 1991.
93. M. Harville. A framework for high-level feedback to adaptive, per-pixel, mixture-of-gaussian background models. In *European Conf. Computer Vision*, volume 3, pages 543–560, 2002.
94. Y. He and A. Kundu. 2-D shape classification using Hidden Markov Model. *IEEE Trans. Pattern Analysis Machine Intelligence*, 13(11):1172–1184, 1991.
95. D. Heckerman. A tutorial on learning with bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research, 1995. Revised November, 1996 - downloadable from <ftp://ftp.research.microsoft.com/pub/tr/tr-95-06.pdf>.
96. T.K. Ho and M. Basu. Complexity measures of supervised classification problems. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(3):289–300, 2002.
97. J. Hu, M.K. Brown, and W. Turin. HMM based online handwriting recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(10):1039–1045, 1996.
98. Y. Huhtala, J. Kärkkäinen, and H. Toivonen. Mining for similarities in aligned time series using wavelets. In *SPIE Proc. of Data Mining and Knowledge Discovery: Theory, Tools, and Technology*, pages 150–160, 1999.
99. A. Hyvärinen. Survey on independent component analysis. *Neural Computing Survey*, 2:94–128, 1999.
100. G. Iacono, M. Bicego, and V. Murino. Face recognition using multilevel b-splines and support vector machines, 2002. Submitted for publication to Pattern Recognition.
101. S. Ikeda. Construction of phoneme models - model search of hidden Markov models-. In *Proc. Int. Workshop on Intelligent Signal Processing and Communication Systems*, pages 82–87, 1993.
102. M. Isard and A. Blake. CONDENSATION: Conditional density propagation for visual tracking. *Int. J. of Computer Vision*, 29(1):5–28, 1998.
103. M. Isard and J. MacCormick. BraMBLe: a bayesian multiple-blob tracker. In *Int. Conf Computer Vision*, volume 2, pages 34–41, 2001.

104. ISO/IEC/JTC1/SC29/WG1 and N390R. Jpeg 2000 image coding system, March 1997. <http://www.jpeg.org/public/wg1n505.pdf>.
105. T. Jaakkola. Tutorial on variational approximation methods. In *Advanced mean field methods: theory and practice*. MIT Press, 2000.
106. T. Jaakkola and M. Jordan. Bayesian parameter estimation via variational methods. *Statistics and Computing*, 10:25–37, 2000.
107. D.W. Jacobs and D. Weinshall. Classification with nonmetric distances: Image retrieval and class representation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(6):583–600, 2000.
108. A.K. Jain and R.C. Dubes. *Algorithms for clustering data*. Prentice Hall, 1988.
109. A.K. Jain, M. N. Murty, and P.J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999.
110. A.K. Jain and D. Zongker. Representation and recognition of handwritten digits using deformable templates. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(12):1386–1391, 1997.
111. R. Jain, R. Kasturi, and B.G. Schunck. *Machine Vision*. McGraw-Hill, 1995.
112. T. Jebara and A. Pentland. Action reaction learning: Automatic visual analysis and synthesis of interactive behavior. In *Proc. Int Conf. Computer Vision Systems*, 1999.
113. I.T. Jolliffe. *Principal component analysis*. Springer Verlag, New York, 1986.
114. B. Juang and L. Rabiner. A probabilistic distance measure for hidden markov models. *AT&T Tech. Journal*, 64(2):391–408, 1985.
115. B.H. Juang. A simple complex in artificial intelligence and machine learning. *Int. Journal of Pattern Recognition and Artificial intelligence*, 15(1):5–7, 2001. Special Issue in Hidden Markov Models in Vision.
116. B.H. Juang, S.E. Levinson, and M.M. Sondhi. Maximum likelihood estimation for multivariate mixture observations of Markov Chain. *IEEE Trans. Information Theory*, 32(2):307–309, 1986.
117. B.H. Juang and L.R. Rabiner. Mixture autoregressive hidden Markov models for speech signals. *IEEE Trans. Acoustic Speech and Signal Processing*, 33(6):1404–1413, 1985.
118. C. Jutten and Herault J. Blind separation of source, part I: an adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24:1–10, 1991.
119. R.E. Kalman. A new approach to linear filtering and prediction problems. *Transaction of the ASME - Journal of Basic Engineering*, pages 35–45, 1960.
120. Takeo Kanade. Picture processing system by computer complex and recognition of human faces, November 1973. doctoral dissertation, Kyoto University.
121. P.C. Kanellakis and S.A. Smolka. Ccs expressions, finite state processes, and three problems of equivalence. *Information and Computation*, 86(1):43–68, 1990.
122. L. Kaufman and P. Rousseuw. *Findings groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, New York, 1990.
123. Z. Keirn. Alternative modes of communication between man and machine. Master's thesis, Purdue University, 1988.
124. B. King. Step-wise clustering procedure. *J. Am. Sta. Assoc.*, 69:86–101, 1967.
125. V. V. Kohir and U. B. Desai. Face recognition using DCT-HMM approach. In *Workshop on Advances in Facial Image Analysis and Recognition Technology (AFIART)*, Freiburg, Germany, June 1998.
126. T. Kolenda, S. Sigurdsson, O. Winther, L.K. Hansen, and J. Larsen. Dtu:toolbox. Internet, 2002. <http://mole.imm.dtu.dk/toolbox/>.
127. T. Kosaka, S. Matsunaga, and M. Kuraoka. Speaker-independent phone modeling based on speaker-dependent hmm's composition and clustering. In *Int. Proc. on Acoustics, Speech, and Signal Processing*, volume 1, pages 441–444, 1995.

128. C. Koutropoulos, A. Tefas, and I. Pitas. Morphological elastic graph matching applied to frontal face authentication under well-controlled and real conditions. In *International Conference on Multimedia Computing and Systems*, volume 2, pages 934–938, 1999.
129. V. Krishnamurthy, S. Dey, and J. LeBlanc. Blind equalization of iir channels using hidden Markov models and extended least squares. *IEEE Trans on Signal Processing*, 43(12):2994–3006, 1995.
130. S. Kullback. *Information theory and Statistics*. Dover publications, New York, 1959.
131. S. Kullback and R.A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.
132. M. Lades, J. Vorbruggen, J. Buhmann, J. Lange, C. Malsburg, and R. Wurtz. Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers*, 42(3):300–311, 1993.
133. K. M. Lam and H. Yan. An analytic-to-holistic approach for face recognition on a single frontal view. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(7):673–686, 1998.
134. M.H. Law and J.T. Kwok. Rival penalized competitive learning for model-based sequence. In *Proc. Int. Conf. Pattern Recognition*, volume 2, pages 195–198, 2000.
135. S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back. Face recognition: a convolutional neural network approach. *IEEE Transactions on Neural Networks*, 8(1):98–113, November 1997.
136. J.J. Lee, J. Kim, and J.H. Kim. Data-driven design of HMM topology for online handwriting recognition. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 15(1):107–121, 2001.
137. K.F. Lee. Context-dependent phonetic hidden Markov models for speaker-independent continuous speech recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38(4):599–609, 1990.
138. S.E. Levinson, L.R. Rabiner, and M.M Sondhi. An introduction to the application of the theory of probabilistic function of a Markov process to automatic speech recognition. *Bell Syst. Tech. J.*, 62(4):1035–1074, 1983.
139. C. Li. *A Bayesian Approach to Temporal Data Clustering using Hidden Markov Model Methodology*. PhD thesis, Vanderbilt University, 2000.
140. C. Li and G. Biswas. Clustering sequence data using hidden Markov model representation. In *Proc. of SPIE'99 Conf. on Data Mining and Knowledge Discovery: Theory, Tools, and Technology*, pages 14–21, 1999.
141. C. Li and G. Biswas. A bayesian approach to temporal data clustering using hidden Markov models. In *Proc. Int. Conf. on Machine Learning*, pages 543–550, 2000.
142. C. Li and G. Biswas. Applying the Hidden Markov Model methodology for unsupervised learning of temporal data. *Int. Journal of Knowledge-based Intelligent Engineering Systems*, 6(3):152–160, 2002.
143. C. Li, G. Biswas, M. Dale, and P. Dale. Matryoshka: A HMM based temporal data clustering methodology for modeling system dynamics. *Intelligent Data Analysis Journal*, in press, 2002.
144. D. Li, A. Biem, and J. Subrahmonia. HMM topology optimization for handwriting recognition. In *Proc. of IEEE Int. Conf Acoustics, Speech, and Signal Processing*, volume 3, pages 1521–1524, 2001.
145. J. Li, A. Najmi, and R.M. Gray. Image classification by a two-dimensional hidden Markov model. *IEEE Trans on Signal Processing*, 48(2):517–533, 2000.
146. S.Z. Li. *Markov Random Field Modeling in Computer Vision*. Springer, 1995.
147. S. Lin, S. Kung, and L. Lin. Face recognition/detection by probabilistic decision-based neural network. *IEEE Trans. Neural Networks*, 8(1):114–131, January 1997.

148. A. Lisitsa and V. Sazanov. Bounded Hyperset Theory and Web-like Data Bases. In *5th Kurt Gödel Colloquium*, volume 1289 of *LNCS*, pages 172–185, 1997.
149. S. M. Lucas. Face recognition with the continuous n-tuple classifier. In *Proceedings of British Machine Vision Conference*, September 1997.
150. R.B. Lyngsø, C.N.S. Pedersen, and H. Nielsen. Metrics and similarity measures for hidden markov models. In *Proc. Int. Conf. Intelligent system for molecular Biology*, pages 178–186, 1999.
151. D. MacKay. Ensemble learning for Hidden Markov Models, 1997. Unpublished. Department of Physics, University of Cambridge. Available at <http://wol.ra.phy.cam.ac.uk/mackay>.
152. D. Mackay. Introduction to Monte Carlo methods. In M. Jordan, editor, *Learning in Graphical Models*. MIT Press, 1999. available as <ftp://wol.ra.phy.cam.ac.uk/pub/mackay/erice.ps.gz>.
153. S. Mallat and Z. Zhang. Matching pursuit with time-frequency dictionaries. *IEEE Trans. on Signal Processing*, 41(12):3397–3415, 1993.
154. G. McLachlan and D. Peel. *Finite Mixture Models*. John Wiley & Sons, New York, 2000.
155. R. Milner. Operational and Algebraic Semantics of Concurrent Processes. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, 1990.
156. B. Moghaddam, W. Wahid, and A. Pentland. Beyond eigenfaces: Probabilistic matching for face recognition. In *The 3rd IEEE International Conference on Automatic Face and Gesture Recognition*, Nara, Japan, April 1998.
157. C. Mulcahay. Image compression using the haar wavelet transform. *Spelman College Science & Mathematics Journal*, 1(1):22–31, 1997.
158. F. Murtagh. A survey of recent advances in hierarchical clustering algorithms which use cluster centres. *Comput. J.*, 26:354–359, 1984.
159. R.M. Neal and G.E. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 355–368. Kluwer Academic Publishers, 1998.
160. A. V. Nefian and M. H. Hayes. Hidden Markov models for face recognition. In *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2721–2724, Seattle, May 1998.
161. H.B. Nielsen. Ucmf - an algorithm for unconstrained, nonlinear optimization. Technical report, IMM, Technical University of Denmark, 2001.
162. P.L. Nunez. *Neocortical Dynamics and Human EEG Rhythms*. Oxford University Press, 1995.
163. T. Oates, L. Firoiu, and P.R. Cohen. Clustering time series with hidden markov models and dynamic time warping. In *Proc. of the IJCAI workshop on robot action planning*, 1999.
164. N. Ohta. A statistical approach to background subtraction for surveillance systems. In *Int. Conf. Computer Vision*, volume 2, pages 481–486, 2001.
165. J. Oliver, R. Baxter, and C. Wallace. Unsupervised Learning using MML. In *Machine Learning: Proceedings of the Thirteenth International Conference (ICML 96)*, pages 364–372. Morgan Kaufmann Publishers, 1996.
166. R. Paige and R.E. Tarjan. Three partition refinement algorithms. *SIAM Journal on Computing*, 16(6):973–989, 1987.
167. A. Panuccio, M. Bicego, and V. Murino. A Hidden Markov Model-based approach to sequential data clustering. In T. Caelli, A. Amin, R.P.W. Duin, M. Kamel, and D. de Ridder, editors, *Structural, Syntactic and Statistical Pattern Recognition*, LNCS 2396, pages 734–742. Springer, 2002.
168. H.-S. Park and S.-W. Lee. A truly 2D hidden Markov model for off-line handwritten character recognition. *Pattern Recognition*, 31(12):1849–1864, 1998.

169. E. Pekalska and R.P.W. Duin. Automatic pattern recognition by similarity representations. *Electronics Letters*, 37(3):159–160, 2001.
170. E. Pekalska and R.P.W. Duin. Dissimilarity representations allow for building good classifiers. *Pattern Recognition Letters*, 23(8):943–956, 2002.
171. E. Pekalska and R.P.W. Duin. Prototype selection for finding efficient representations of dissimilarity data. In R. Kasturi, D. Laurendeau, and C. Suen, editors, *16th Int. Conf. on Pattern Recognition*, volume 3, pages 37–40, 2002.
172. E. Pekalska and R.P.W. Duin. Spatial representation of dissimilarity data via lower-complexity linear and non linear mappings. In T. Caelli, A. Amin, R.P.W. Duin, M. Kamel, and D. de Ridder, editors, *Structural, Syntactic and Statistical Pattern Recognition*, LNCS 2396, pages 488–497. Springer, 2002.
173. E. Pekalska, P. Paclik, and R.P.W. Duin. A generalized kernel approach to dissimilarity-based classification. *Journal of Machine Learning Research*, 2(2):175–211, 2002.
174. W.D. Penny and S.J. Roberts. Dynamic models for nonstationary signal segmentation. *Computers and Biomedical Research*, 32(6):483–502, 1998.
175. W.D. Penny, S.J. Roberts, E. Curran, and M. Stokes. EEG-based communication: a pattern recognition approach. *IEEE Trans. Rehabilitation Engineering*, 8(2):214–215, 2000.
176. A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. *Computer Vision and Pattern Recognition*, pages 84–91, 1994.
177. D.-T. Pham, P. Garrat, and C. Jutten. Separation of a mixture of independent sources through a maximum likelihood approach. In *Proc. of EUSIPCO*, pages 771–774, 1992.
178. B. Povlow and S. Dunn. Texture classification using noncausal hidden Markov models. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 17(10):1010–1014, 1995.
179. L. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition*. Prentice-Hall, Englewood Cliffs, 1993.
180. L.R. Rabiner. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proc. of IEEE*, 77(2):257–286, 1989.
181. L.R. Rabiner, C.H. Lee, B.H. Juang, and J.G. Wilpon. HMM clustering for connected word recognition. In *Proc. of IEEE ICASSP*, pages 405–408, 1989.
182. G. Radons, J.D. Becker, B. Dülfer, and J. Krüger. Analysis, classification, and coding of multielectrode spike trains with hidden Markov models. *Biol. Cybern.*, 71:359–373, 1994.
183. A.E. Raftery. Bayesian model selection in social research. *Sociological Methodology*, pages 111–196, 1995.
184. C. Raphael. Automatic segmentation of acoustic musical signals using hidden Markov models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(4):360–370, 1999.
185. J. Rissanen. Stochastic complexity and modeling. *The Annals of Statistics*, 14, 1986.
186. E.S. Ristad and P.N. Yianilos. Learning string edit distance. In J.D.H. Fisher, editor, *Proc. of Int. Conf. on Machine Learning*. Morgan Kaufmann Publishers, Inc., 1997.
187. C. Robert. *The Bayesian choice: a decision theoretic motivation*. Springer-Verlag, New York, 1994.
188. C.P. Robert, T. Rydén, and D.M. Titterton. Bayesian inference in hidden Markov models through jump Markov chain Monte Carlo. *J. Royal Statistic Society B*, 62(1):57–75, 2000.

189. S. Roberts, D. Husmeier, I. Rezek, and W. Penny. Bayesian approach to gaussian mixture modelling. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(11):1133–1142, 1998.
190. T. Rydén, T. Teräsvirta, and S. Åsbrink. Stylized facts of daily return series and the hidden Markov model of absolute returns. *Journal of Applied Econometrics*, 13:217–244, 1998.
191. S. Saha. Image compression - from DCT to wavelets: A review. *ACM Crossroads Magazine*, Spring 2000.
192. S.L Salzberg. Gene discovery in dna sequences. *IEEE Intelligent Systems*, 14(6):44–48, 1999.
193. S.L. Salzberg, D. Searls, and S. Kasif. *Computational methods in Molecular Biology*. Elsevier Science, 1998.
194. F. Samaria. *Face recognition using Hidden Markov Models*. PhD thesis, Engineering Department, Cambridge University, October 1994.
195. F. Samaria and A. Harter. Parameterisation of a stochastic model for human face identification. In *Proceedings of IEEE Workshop on Applications of Computer Vision*, Sarasota, Florida, December 1994.
196. G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.
197. T.B. Sebastian, P.N. Klein, and B.B. Kimia. Recognition of shapes by editing Shock Graphs. In *Proc. Int Conf. Computer Vision*, pages 755–762, 2001.
198. L.G. Shapiro and G.C. Stockman. *Computer Vision*. Prentice Hall, 2001.
199. H. Singer and M. Ostendorf. Maximum likelihood successive state splitting. In *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, volume 2, pages 601–604, 1996.
200. L. Sirovich and M. Kirby. Low dimensional procedure for the characterization of human face. *J. Opt. Soc. Am. A*, 4(3):519–524, March 1987.
201. P. Smyth. Clustering sequences with hidden Markov models. In M. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing*, volume 9. MIT Press, 1997.
202. P.H.A. Sneath and R.R. Sokal. *Numerical taxonomy*. Freeman, London, UK, 1973.
203. C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *Int. Conf. Computer Vision and Pattern Recognition*, volume 2, 1999.
204. B. Stenger, V. Ramesh nad N. Paragios, F.Coetzee, and J. M. Buhmann. Topology free hidden Markov models: Application to background modeling. In *Int. Conf. Computer Vision*, volume 1, pages 294–301, 2001.
205. A. Stolcke and S. Omohundro. Hidden Markov Model induction by Bayesian model merging. In S.J. Hanson, J.D. Cowan, and C.L Giles, editors, *Advances in Neural Information Processing Systems*, volume 5, pages 11–18. Morgan Kaufmann, San Mateo, CA, 1993.
206. A. Stolcke and S.M. Omohundro. Best-first model merging for Hidden Markov Model induction. Technical Report TR-94-003, ICSI, 1994.
207. M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society B*, 36:111–147, 1974.
208. J. Takami and S. Sagayama. A successive state splitting algorithm for efficient allophone modeling. In *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, volume 1, pages 573–576, 1992.
209. A. Tefas, C. Kotropoulos, and I. Pitas. Using support vector machines to enhance the performance of elastic graph matching for frontal face authentication. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(7), July 2001.
210. S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, 1999.

211. L. Tierney and J.B. Kadane. Accurate approximations for posteriors moments and marginal densities. *Journal of Statistical Association*, 81:82–86, 1986.
212. K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: Principles and practice of background maintenance. In *Int. Conf. Computer Vision*, pages 255–261, 1999.
213. H. Van Trees. *Detection, Estimation and Modulation Theory*, volume 1. John Wiley, New York, 1968.
214. M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.
215. V. Vapnik. *Statistical Learning Theory*. John Wiley, New York, 1998.
216. R. Vasko, A. El-Jaroudi, and R. Boston. An algorithm to determine hidden Markov model topology. In *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pages 3578–3581, 1996.
217. S.R. Veltman and R. Prasad. Hidden Markov models applied to on-line handwritten isolated character recognition. *IEEE Trans. on Image Processing*, 3(3):314–318, 1994.
218. L. Venkataramanan, R. Kuc, and F.J. Sigworth. Identification of hidden Markov models for ion channel currents. Part II. State-dependent excess noise. *IEEE Trans. on Signal Processing*, 46(7):1916–1929, 1998.
219. L. Venkataramanan, R. Kuc, and F.J. Sigworth. Identification of hidden Markov models for ion channel currents. Part III: Bandlimited sampled data. *IEEE Trans. on Signal Processing*, 48(2):376–385, 2000.
220. L. Venkataramanan, J.L. Walsh, R. Kuc, and F.J. Sigworth. Identification of hidden Markov models for ion channel currents. Part I. Colored background noise. *IEEE Trans. on Signal Processing*, 46(7):1901–1915, 1998.
221. P. Vincent and Y. Bengio. Kernel matching pursuit. *Machine Learning*, 48(1):165–187, 2002.
222. A.J. Viterbi. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Trans. on Information Theory*, IT-13:260–269, 1967.
223. R.A. De Vore, B. Jawerth, and B.J. Lucier. Image compression through wavelet transform coding. *IEEE Trans. on Information Theory*, 38(2), 1992.
224. J.H. Ward. Hierarchical grouping to optimize an objective function. *J. Am. Sta. Assoc.*, 58:236–244, 1963.
225. A.D. Wilson and A.F. Bobick. Parametric hidden Markov models for gesture recognition. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 21(9):884–900, 1999.
226. C.F.J. Wu. On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11(1):95–103, 1983.
227. L. Xu, A. Krzyzak, and E. Oja. Rival penalized competitive learning for clustering analysis, RBF nets, and curve detection. *IEEE Trans. on Neural Networks*, 4(4):636–648, 1993.
228. P. C. Yuen and J. H. Lai. Face representation using independent component analysis. *Pattern Recognition*, 35(6):1247–1257, 2002.
229. J. Zhang, Y. Yan, and M. Lades. Face recognition: eigenface, elastic matching and neural nets. *Proceedings of the IEEE*, 85(9), September 1997.
230. M. Zimmermann and H. Bunke. Hidden Markov model length optimization for handwriting recognition systems, 2001. TR IAM-01-003 University of Bern.

Acknowledgements

The first thank is surely for my advisor Professor Vittorio Murino, to whom I am debtor for a twofold reason: first, because three years ago he gave me the great opportunity of working with him, without knowing me, but basing only on his first impression. Second, because during these three years he followed my free evolution in the Pattern Recognition research area, without imposing me any constraint deriving from application or project reasons. I want to thank him also for the uncountable number of “evening” discussions, that gave me the possibility of learning several concepts and comparing my impressions on several questions with him and his experience.

The second person I want to thank is the Professor M̀ario Figueiredo, that gave me the chance of staying 4 months in his laboratory in Lisboa, patiently teaching me a lot of interesting and fundamental things, as the Bayes Theory. Many “obrigado” also for all the time he spent after my stay for collaborating with me.

Then I want to thank all other people I worked with: Agostino Dovier, which introduces me in the bisimulation equivalence area, Antonello Panuccio, my colleague in the first part of the HMM journey, and Marco Cristani, that enters with me in several “tunnels”, scientific and not, teaching me that the fantasy is fundamental in all situations. Many thanks to Gianluca Sandrini, who carefully read the draft of this thesis, revealing me all the secrets of the English language.

I want also to thank Umberto Castellani e Andrea Fusiello, even if belonging to a different “religion” (Computer Vision rather than Pattern Recognition), with who I had several interesting and stimulating discussions, aimed at finding contact points. And finally, many thanks to the whole VIPS lab, that helps me in enjoying my stay in this university.

Many thanks to my family, that supported me in all my decisions, even if suffered, and gives me a lot of love and support.

But the largest thank is for my “girlfriend” Giovanna (that is now my “wife”, what an emotion!) for all the love she transmits to me, and for all the support: ideological (interminable discussions on the sense of the life), practical (she is a fantastic cooker) and scientific (she helps me in reviewing the thesis and some important papers). This thesis is dedicated to her.

Sommario

Questa tesi, intitolata “*Modelli di Markov a stati nascosti per la Pattern Recognition e la Visione Artificiale: aspetti metodologici e applicativi*”, si colloca nel contesto degli approcci probabilistici alla *Pattern Recognition*: tali approcci assumono che l’informazione sul problema, le dipendenze tra i vari fattori e i risultati prodotti siano tutti espressi in termini di probabilità. In particolare il presente lavoro è incentrato sulla tecnica denominata Modello di Markov a stati nascosti (*Hidden Markov Model* - HMM), classificatore statistico ampiamente utilizzato per analisi di sequenze. Questa tecnica può essere intesa come estensione del modello di Markov dal quale differisce per la non osservabilità dei suoi stati. Tale modello trova svariate applicazioni, sia nell’ambito della *Pattern Recognition* che nell’ambito della Visione Artificiale; negli ultimi anni, in particolare, questa tecnica è stata applicata con successo a tutte quelle problematiche che prevedono un’analisi di dati sequenziali (temporali o non).

Ciononostante, alcune questioni risultano ancora aperte, legate sia alla metodologia stessa che all’applicazione in nuovi emergenti contesti. Questa tesi nasce sulla base di queste considerazioni, ponendosi come obiettivo il raggiungimento di un duplice scopo: da un lato si vogliono individuare e analizzare i problemi metodologici ancora irrisolti degli HMM, dall’altro si vogliono investigare nuovi utilizzi di questa metodologia in problematiche di *Pattern Recognition* e Visione Artificiale.

Da un punto di vista metodologico, questa tesi propone svariati contributi in diversi contesti, quali il problema della selezione del modello, il problema della classificazione e il problema della classificazione non supervisionata (detta anche *clustering*). Nel primo contesto, l’obiettivo è quello di determinare automaticamente dai dati la migliore struttura del modello, intesa come numero di stati e connettività tra di essi. Come prima cosa è stata prodotta una dimostrazione formale di equivalenza tra modelli Gaussiani continui, che riduce notevolmente lo spazio di ricerca del miglior modello. Successivamente sono stati introdotti tre metodi, originali e innovativi, in grado di determinare automaticamente la miglior struttura dai dati. Per quanto concerne la classificazione con HMM, è stato analizzato lo schema classico di classificazione, proponendo alcune considerazioni sull’affidabilità di una decisione presa da tale schema. Successivamente è stato introdotto uno schema di riconoscimento alternativo, basato sulla classificazione per similarità: l’utilizzo di questo schema ha portato a notevoli miglioramenti nelle prestazioni del sistema,

sia in casi sintetici che reali. Per quanto riguarda il *clustering* con HMM, contesto scarsamente investigato nella letteratura dell'HMM, i contributi introdotti riguardano principalmente la definizione di misure di distanza tra sequenze, la definizione di più efficienti algoritmi di *clustering*, e l'introduzione di uno schema alternativo, basato sulla rappresentazione per similarità introdotta nel contesto della classificazione. Questo schema, che produce risultati interessanti, permette di ricondurre un difficoltoso problema di modellazione di sequenze ad un più collaudato problema di modellazione di dati non sequenziali (vettori di *features*).

Tutte le tecniche proposte sono state attentamente valutate e validate attraverso esperimenti con problemi sintetici e reali, quali la classificazione di forme planari, il riconoscimento di volti, la modellazione di sequenze di DNA, la segmentazione di segnali elettro-encefalografici e l'analisi di sequenze video. Gli ottimi risultati ottenuti hanno dimostrato la validità delle metodologie proposte.

Da un punto di vista più strettamente applicativo, l'utilizzo di tecniche basate su HMM ha prodotto in alcuni ambiti un contributo importante anche nel contesto dell'applicazione stessa. Più in particolare, nella classificazione di forme 2D è stato introdotto un sistema di riconoscimento basato su HMM, in grado di affrontare con successo perturbazioni delle forme quali la traslazione, la rotazione, l'occlusione, le proiezioni affini e il rumore. Nel riconoscimento di volti è stato introdotto uno schema basato su HMM e *wavelet*; i risultati prodotti risultano essere migliori di tutti quelli proposti nella letteratura sullo stesso database standard. Infine, è stato proposto un metodo per l'analisi di sequenze video, basato sul *clustering* di HMM. Il metodo proposto si è dimostrato essere in grado di suddividere la scena statica in regioni di omogeneità spaziale, cromatica e temporale.

Gli ottimi risultati ottenuti dagli algoritmi proposti in queste applicazioni dimostrano, se ancora necessario, l'efficacia e la vasta applicabilità dei modelli di Markov a stati nascosti.