Pattern Recognition 42 (2009) 27 – 32

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Soft clustering using weighted one-class support vector machines

Manuele Bicego[a,*], Mario A.T. Figueiredo[b]

[a] DEIR, University of Sassari, via Torre Tonda, Sassari, Italy
[b] Instituto de Telecomunicações, Instituto Superior Técnico, Lisboa, Portugal

ARTICLE INFO

Article history:
Received 29 August 2007
Received in revised form 6 May 2008
Accepted 12 July 2008

Keywords:
Soft clustering
One-class support vector machines
EM-like algorithms
Kernel methods
Deterministic annealing

ABSTRACT

This paper describes a new soft clustering algorithm in which each cluster is modelled by a one-class support vector machine (OC-SVM). The proposed algorithm extends a previously proposed hard clustering algorithm, also based on OC-SVM representation of clusters. The key building block of our method is the weighted OC-SVM (WOC-SVM), a novel tool introduced in this paper, based on which an expectation–maximization-type soft clustering algorithm is defined. A deterministic annealing version of the algorithm is also introduced, and shown to improve the robustness with respect to initialization. Experimental results show that the proposed soft clustering algorithm outperforms its hard clustering counterpart, namely in terms of robustness with respect to initialization, as well as several other state-of-the-art methods.

1. Introduction

Kernel-based methods [1–4] have emerged in the last decade as a flexible and effective approach for addressing pattern recognition problems. These methods have been largely employed in supervised learning contexts (e.g., support vector classification and regression [1–4]) but their application in unsupervised learning represents a more recent and less explored trend [5–9].

Recently Camastra and Verri [8] have presented a hard clustering scheme which uses one-class support vector machines (OC-SVMs) to represent the clusters. An OC-SVM is a binary classification machine which is trained using only "positive examples" (i.e., examples from one class). In the version of Tax and Duin [10], training the OC-SVM consists in determining the smallest hyper-sphere containing the training data. In the version of Schölkopf et al. [11,12], the OC-SVM training algorithm works by finding the maximum margin separation between the training points and the origin. Both approaches involve only inner products between pairs of data points, thus being easily kernelizable via the famous "kernel-trick" [3,4]. After training, an OC-SVM is able to provide an answer to the question: "Was this new data point generated by the same distribution that generated the training data, or is it an outlier?" In fact, using its soft output, an OC-SVM is also able to provide a soft (or fuzzy) answer to this question, in the form of a real number which expresses a degree of confidence with which a new data point belongs to the same class (i.e., density) as the training data. The OC-SVM was proposed mainly to address (in a non-parametric way) problems of outlier (or novelty) detection.

The clustering scheme of Camastra and Verri is an iterative procedure, similar to the well-known K-means algorithm [13], which learns a set of K OC-SVMs. In each iteration, every data point is assigned to the "nearest" cluster, and then the clusters are updated using the assigned points. Specifically, the distance measure based on which the "nearest" cluster is found is provided by the OC-SVMs. For example, with the Tax and Duin OC-SVM, the "distance" between a point and a given cluster is simply the distance between that point and the center of the hyper-sphere of the corresponding OC-SVM. After all the points have been assigned to the clusters, each OC-SVM is retrained using only the corresponding points and the procedure is repeated until some convergence criterion is met. This corresponds to a hard clustering (K-means-type) scheme since, in each iteration, each pattern is assigned to a single cluster. It is known that iterative hard clustering schemes, namely K-means, are very sensitive to initialization. In Ref. [8], the authors skip this problem by manually initializing the method using a subset of points; the resulting method is thus not a fully unsupervised learning procedure.

Higher robustness to initialization is typically exhibited by soft clustering methods, such as finite mixture fitting using the expectation–maximization (EM) algorithm [14,15], which tend to be more effective than K-means in avoiding local minima of the underlying criteria. The robustness to initialization can be further improved by resorting to deterministic annealing strategies [17,18], in which the smoothness of the assignments is initially very high and then slowly decreased until some desired value.

* Corresponding author. Tel.: +39 79 2017321; fax: +39 79 2017312.
E-mail addresses: bicego@uniss.it (M. Bicego), mario.figueiredo@lx.it.pt (M.A.T. Figueiredo).

0031-3203/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2008.07.004

In this paper we present two contributions to OC-SVM-based clustering:

- We extend the concept of OC-SVM by defining a novel variant, which we call weighted OC-SVM (WOC-SVM). The WOC-SVM can be trained using a set of points and associated weights, where each weight indicates the importance to be given to the corresponding point. We define weighted versions of both the Tax and Duin and the Schölkopf et al. OC-SVM formulations. To the best of our knowledge, no method had been proposed for incorporating weights into OC-SVM training, although this problem had been already addressed in standard two-class (binary) SVMs; see, e.g., Ref. [16].
- We introduce a soft clustering method based on the WOC-SVM, which can be considered as a soft version of the method of Camastra and Verri [8]. The proposed method is similar to an EM algorithm for fitting a finite mixture, in which the density of each component is a function of the soft output of the corresponding WOC-SVM. The proposed approach depends on a scale parameter, which can be used to control the softness of the cluster assignments. This parameter opens the door to a deterministic annealing version of the basic scheme, which is shown to be more robust to initialization.

Experimental results on different UCI ML-Repository datasets demonstrate that the proposed soft clustering algorithm compares favorably with its hard clustering counterpart (an implementation of the method of Camastra and Verri [8] with similar characteristics), particularly in terms of robustness with respect to initialization. Moreover, when compared with other state-of-the-art methods, its performance is very competitive.

The remaining sections of the paper are organized as follows. In Section 2, we introduce the weighted versions of the OC-SVM formulations of Tax and Duin and of Schölkopf et al. Section 3 contains the description of the new soft clustering algorithm based on the WOC-SVM as well as the description of its deterministic annealing variant; Section 4 reports experiments, and, finally, Section 5 concludes the paper.

2. Weighted One-Class Support Vector Machines

In this section, after briefly reviewing the OC-SVM formulations of Tax and Duin [10] and of Schölkopf et al. [11,12], namely to establish the necessary notation, we introduce the proposed weighted versions.

2.1. The Tax and Duin OC-SVM

In the OC-SVM formulation of Tax and Duin [10], the goal is to find the smallest sphere containing the data points $\{\mathbf{x}_1, \ldots, \mathbf{x}_\ell\}$, with some relaxation given by the so-called slack variables. This goal is formulated as a constrained convex optimization problem:

$$\begin{aligned} \min_{R,\mathbf{a},\xi_1,\ldots,\xi_\ell} \quad & R^2 + C\sum_{i=1}^{\ell} \xi_i \\ \text{s.t.} \quad & \|\mathbf{x}_i - \mathbf{a}\|^2 \leqslant R^2 + \xi_i, \quad i = 1,\ldots,\ell, \\ & \xi_i \geqslant 0, \quad i = 1,\ldots,\ell, \end{aligned} \tag{1}$$

where R and $\mathbf{a}$ are, respectively, the radius and center of the sphere, $\|\cdot\|$ denotes the Euclidean norm, the ξ_i are the slack variables, and C is a trade-off parameter controlling how much the slack variables are penalized. The Wolfe dual of this problem is

$$\begin{aligned} \min_{\alpha_1,\ldots,\alpha_\ell} \quad & \sum_{i=1}^{\ell} \alpha_i \langle \mathbf{x}_i, \mathbf{x}_i \rangle - \sum_{i=1}^{\ell}\sum_{j=1}^{\ell} \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\ \text{s.t.} \quad & 0 \leqslant \alpha_i \leqslant C, \quad i = 1,\ldots,\ell, \\ & \sum_{i=1}^{\ell} \alpha_i = 1, \end{aligned} \tag{2}$$

where $\langle \cdot, \cdot \rangle$ is the inner product, and $\{\alpha_1, \ldots, \alpha_\ell\}$ are Lagrange multipliers. Denoting the solution to problem (2) as $\alpha_1^*, \ldots, \alpha_\ell^*$, the sphere center is given by $\mathbf{a} = \sum_i \alpha_i^* \mathbf{x}_i$, thus the squared distance between a given test point $\mathbf{x}$ and $\mathbf{a}$ is

$$\|\mathbf{x} - \mathbf{a}\|^2 = \langle \mathbf{x}, \mathbf{x} \rangle - 2\sum_{i=1}^{\ell} \alpha_i^* \langle \mathbf{x}_i, \mathbf{x} \rangle + \sum_{i=1}^{\ell}\sum_{j=1}^{\ell} \alpha_i^* \alpha_j^* \langle \mathbf{x}_i, \mathbf{x}_j \rangle, \tag{3}$$

i.e., it only involves inner products. Typically, the decision of whether $\mathbf{x}$ belongs to the same class as the training data or not is obtained by comparing $\|\mathbf{x} - \mathbf{a}\|^2$ with some threshold. Clearly $\|\mathbf{x} - \mathbf{a}\|^2$ can be seen as a squared distance measure between $\mathbf{x}$ and the class defined by the training data.

The fact that Eqs. (2) and (3) only depend on the data via inner products allows using the *kernel-trick* to obtain a kernelized version of the OC-SVM [10]: simply replace all the inner products $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ in Eqs. (2) and (3) by the kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$. In the kernel version, the hyper-sphere lives in a high (maybe infinite) dimensional space induced by the kernel [3].

2.2. The Schölkopf et al. OC-SVM

In the OC-SVM formulation of Schölkopf et al. [11,12], the idea is to find a hyper-plane $\langle \mathbf{w}, \mathbf{x} \rangle + \rho = 0$ that separates the data from the origin with maximal margin. This goal is also formulated as a convex problem:

$$\begin{aligned} \min_{\mathbf{w},\xi_1,\ldots,\xi_\ell,\rho} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{\nu\ell}\sum_{i=1}^{\ell} \xi_i - \rho \\ \text{s.t.} \quad & \langle \mathbf{w}, \mathbf{x}_i \rangle \geqslant \rho - \xi_i, \quad i = 1,\ldots,\ell, \\ & \xi_i \geqslant 0, \quad i = 1,\ldots,\ell, \end{aligned} \tag{4}$$

where the ξ_i are the slack variables and ν controls the amount of penalization incurred by these slack variables. The corresponding Wolfe dual is

$$\begin{aligned} \min_{\alpha_1,\ldots,\alpha_\ell} \quad & \sum_{i=1}^{\ell}\sum_{j=1}^{\ell} \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\ \text{s.t.} \quad & 0 \leqslant \alpha_i \leqslant 1/(\nu\ell), \quad i = 1,\ldots,\ell, \\ & \sum_{i=1}^{\ell} \alpha_i = 1, \end{aligned} \tag{5}$$

where $\{\alpha_1, \ldots, \alpha_\ell\}$ are Lagrange multipliers. Denoting the solution of Eq. (5) as $\alpha_1^*, \ldots, \alpha_\ell^*$, and since $\mathbf{w} = \sum_i \alpha_1^* \mathbf{x}_i$, the (directed) distance from a given point $\mathbf{x}$ to the separating hyper-plane is given by

$$d(\mathbf{x}) = \sum_{i=1}^{\ell} \alpha_i^* \langle \mathbf{x}_i, \mathbf{x} \rangle - \rho. \tag{6}$$

Parameter ρ can be obtained from the fact that, for any α_i^* such that $0 < \alpha_i^* < 1/(\nu\ell)$, the corresponding data point $\mathbf{x}_i$ satisfies

$$\rho = \sum_{j=1}^{\ell} \alpha_j^* \langle \mathbf{x}_i, \mathbf{x}_j \rangle. \tag{7}$$

The decision of whether some point $\mathbf{x}$ belongs to the same class as the training data is obtained by comparing $d(\mathbf{x})$ with zero. The function $d(\mathbf{x})$ can be seen as a measure of similarity between its argument and the learnt class. The fact that this formulation only involves inner products also allows for easy kernelization, simply by replacing each inner product $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ by the kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$.

2.3. Weighted OC-SVM

The weighted version of the OC-SVM is able to take into account a set of weights $\{w_1, \ldots, w_\ell\}$, where $w_i \in [0, 1]$, for $i = 1, \ldots, \ell$, indicating the importance assigned to each point of the training set $\{\mathbf{x}_1, \ldots, \mathbf{x}_\ell\}$.

Consider first the Tax and Duin formulation. The introduction of weights into the OC-SVM formulation is carried out by letting the penalty on slack variable ξ_i (which corresponds to the pattern $\mathbf{x}_i$) be proportional to the weight w_i. The rationale is straightforward: if a point $\mathbf{x}_i$ has a small weight, $w_i \ll 1$, the corresponding slack variable ξ_i has a small penalty, thus being able to have a large value, which will allow that point to be far from the center of the sphere, having a weak influence on its location and radius. With this modification, the optimization problem in Eq. (1) becomes

$$\begin{aligned} \min_{R, \mathbf{a}, \xi_1, \ldots, \xi_\ell} \quad & R^2 + C \sum_{i=1}^{\ell} w_i \xi_i \\ \text{s.t.} \quad & \|\mathbf{x}_i - \mathbf{a}\|^2 \leqslant R^2 + \xi_i, \quad i = 1, \ldots, \ell, \\ & \xi_i \geqslant 0, \quad i = 1, \ldots, \ell, \end{aligned} \tag{8}$$

where the variables $R, \mathbf{a}, \xi_1, \ldots, \xi_\ell$, and C have the exact same meaning as in Eq. (1). The Lagrangian for problem (8) is given by

$$\begin{aligned} & L(R, \mathbf{a}, \xi_1, \ldots, \xi_\ell, \alpha_1, \ldots, \alpha_\ell, \beta_1, \ldots, \beta_\ell)) \\ & \quad = R^2 - \sum_i (R^2 + \xi_i - \|\mathbf{x}_i - \mathbf{a}\|^2)\alpha_i - \sum_i \xi_i \beta_i + C \sum_i \xi_i w_i, \end{aligned} \tag{9}$$

where $\alpha_1, \ldots, \alpha_\ell$ and $\beta_1, \ldots, \beta_\ell$ are the Lagrange multipliers associated with the two sets of constraints in Eq. (8). Finally, the dual problem is obtained by minimizing L with respect to $R, \mathbf{a}, \xi_1, \ldots, \xi_\ell$, and $\beta_1, \ldots, \beta_\ell$, which leads to

$$\begin{aligned} \min_{\alpha_1, \ldots, \alpha_\ell} \quad & \sum_{i=1}^{\ell} \alpha_i \langle \mathbf{x}_i, \mathbf{x}_i \rangle - \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\ \text{s.t.} \quad & 0 \leqslant \alpha_i \leqslant w_i C, \quad i = 1, \ldots, \ell, \\ & \sum_{i=1}^{\ell} \alpha_i = 1. \end{aligned} \tag{10}$$

Notice that the objective function in Eq. (10) is the same as the one in Eq. (2); only the constrains are changed. In particular, each α_i is constrained to being less than $w_i C$ rather than C. This is in agreement with the desired behavior: a weight w_i close to zero forces the corresponding α_i to also be close to zero, thus contributing very weakly to the definition of the distance between any point $\mathbf{x}$ and the center of the sphere, which is still given by Eq. (3). This behavior is reenforced by the last constraint (the sum of the α_i has to be equal to one), since by limiting some of the α_i to small values forces the others to have larger values to keep the total sum equal to one. Finally, notice that to guarantee that the feasible set is not empty, the weights have to satisfy

$$\sum_{i=1}^{\ell} w_i \geqslant \frac{1}{C},$$

otherwise, even if each α_i equaled its allowed maximum $w_i C$, their sum would be less than one.

The exact same approach can be applied to the Schölkopf et al. formulation, leading to the dual problem

$$\begin{aligned} \min_{\alpha_1, \ldots, \alpha_\ell} \quad & \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\ \text{s.t.} \quad & 0 \leqslant \alpha_i \leqslant w_i/(\nu\ell), \quad i = 1, \ldots, \ell, \\ & \sum_{i=1}^{\ell} \alpha_i = 1. \end{aligned} \tag{11}$$

As in the previous case, the sole change is in the constraints and all the comments in the previous paragraph are still valid. Namely, to guarantee that the feasible set is not empty, the weights have to satisfy

$$\sum_{i=1}^{\ell} w_i \geqslant \nu\ell.$$

The distance function $d(\mathbf{x})$ and the expression for ρ are, of course, still given by Eqs. (6) and (7), where $\alpha_1^*, \ldots, \alpha_\ell^*$ are now the solution of Eq. (11).

Finally, as in the original (non-weighted) versions, kernelization is obtained by replacing each inner product $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ by the adopted kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$.

3. The soft clustering scheme

In this section, the proposed soft clustering algorithm is presented. The key ideas are that each cluster is represented by one WOC-SVM and that the algorithm has an EM-type structure. After initialization (more on this below), the two following steps are cyclically repeated until some convergence criterion is met (or a maximum number of iterations is reached): the E-type step computes a degree of similarity between every data point and each cluster; the M-type step uses these degrees of similarity as weights to retrain the WOC-SVM representing the clusters.

3.1. The similarities

In the standard EM algorithm for a finite mixture model, the class likelihood represents the "similarity" between some point $\mathbf{x}$ and the class, namely how well the point $\mathbf{x}$ is modelled by a specific component of the mixture. Likewise, in our algorithm we need a function measuring the closeness of a point to a cluster, given that the cluster is modelled with the WOC-SVM. Of course, unlike in a finite mixture model, the "mixture" of WOC-SVM does not correspond to a generative model of the data. Although the WOC-SVM is intrinsically a classifier (with a binary output), we can consider the "soft" output (namely the output before thresholding) as a measure of the degree with which a given point belongs to the corresponding cluster. The notion that the similarities should behave similarly to cluster likelihoods suggests that the soft outputs of the WOC-SVM can be used as follows to obtain these similarities/"likelihoods".

- In the Tax and Duin formulation, the WOC-SVM can provide the squared distance between a point $\mathbf{x}$ and the center of its sphere, via

Eq. (3) or its kernelized version. A natural choice for the similarity between point $\mathbf{x}$ and cluster k is thus

$$\mathscr{S}_{\mathrm{TD}}(\mathbf{x},k)=\exp\left\{-\frac{\|\mathbf{x}-\mathbf{a}_k\|^2}{\sigma}\right\}, \tag{12}$$

where $\mathbf{a}_k$ is the center of the sphere of the WOC-SVM representing cluster k, and the subscript TD in $\mathscr{S}_{\mathrm{TD}}$ stands for "Tax and Duin". Parameter σ controls how the similarity scales with the distance, much as the variance of a Gaussian density; its impact will be analyzed in Section 3.3.

- In the Schölkopf et al. formulation, the WOC-SVM provides, via Eq. (6) or its kernel version, the signed distance between some point $\mathbf{x}$ and the hyper-plane associated with cluster k. Since positive (respectively, negative) values correspond to high (respectively, low) similarities, a natural choice for the similarity between point $\mathbf{x}$ and cluster k is thus

$$\mathscr{S}_{\mathrm{S}}(\mathbf{x},k)=\exp\left\{\frac{d_k(\mathbf{x})}{\sigma}\right\}, \tag{13}$$

where d_k is the signed distance function (6) corresponding to the WOC-SCM of the k-th cluster and the subscript S in $\mathscr{S}_{\mathrm{S}}$ stands for "Schölkopf et al". Again σ controls the scale of the distance.

3.2. The algorithm

With the similarity functions defined above, we are now ready to describe the soft clustering algorithm in detail. Consider a set of ℓ points, $\{\mathbf{x}_1,\ldots,\mathbf{x}_\ell\}$, to be clustered in K groups. The proposed algorithm proceeds as follows:

- *Initialization*: For $k=1,\ldots,K$, initialize the k-th WOC-SVM; for example, train each WOC-SVM using all points as positive examples, each with a different random weight. Three different initialization methods are described in the experimental section. Similarly to the EM algorithm for finite mixtures, we need to employ in the algorithm the mixing coefficients γ_k, which we call "cluster weights"; such weights are initialized as $\gamma_k=1/K$, for $k=1,\ldots,K$.
- *E-type step*: Compute the cluster membership weights

$$z_i^k=\frac{\gamma_k\mathscr{S}(\mathbf{x}_i,k)}{\sum_{r=1}^K\gamma_r\mathscr{S}(\mathbf{x}_i,r)}\quad \text{for } i=1,\ldots,\ell,\ \ k=1,\ldots,K, \tag{14}$$

where $\mathscr{S}(\mathbf{x}_i,k)$ is given by Eq. (12) or (13), computed with the current WOC-SVM of each cluster, and γ_k is the current estimate of the weight of cluster k. Notice that, naturally, $\sum_k z_i^k=1$.
- *M-type step*: For $k=1,\ldots,K$, update the k-th WOC-SVM by training it using as positive examples all the points in the dataset, each weighted according to the set of weights $\{z_1^k,\ldots,z_\ell^k\}$, and update the cluster weight parameters according to

$$\gamma_k=\frac{1}{\ell}\sum_{i=1}^{\ell}z_i^k. \tag{15}$$

Notice that this definition guarantees, exactly as in EM for mixtures, that the γ_k parameters sum to one:

$$\sum_{k=1}^K\gamma_k=\frac{1}{\ell}\sum_{k=1}^K\sum_{i=1}^{\ell}z_i^k=\frac{1}{\ell}\sum_{i=1}^{\ell}\underbrace{\sum_{k=1}^K z_i^k}_{1}=1.$$

- *Stopping criterion*: If a maximum number of iterations has been reached, or some other convergence criterion is satisfied, stop; otherwise, go back to the step.

In our implementation, we used as stopping criterion the convergence of the following function f_{LL}, which takes inspiration from the log-likelihood computed in the mixture EM case:

$$f_{LL}=\sum_{i=1}^{\ell}\log\left(\sum_{k=1}^K\gamma_k\mathscr{S}(\mathbf{x}_i,k)\right). \tag{16}$$

If wanted, a final hard partition can be obtained by assigning each point $\mathbf{x}_i$ to the cluster

$$\arg\max_k\{z_i^k,k=1,\ldots,K\},$$

or simply to the closest cluster.

We would like to stress again that although we have used EM terminology and our algorithm has an EM-type structure, it is not an EM algorithm, namely due to the absence of an underlying probabilistic generative model. For this same reason, we cannot directly import the monotonicity and convergence properties of EM to our algorithm. In the experiments described in the next section, the algorithm almost always converged. In some rare cases, the function continued to oscillate between two values; this behavior is currently under study.

3.3. The scale parameter and deterministic annealing

This section investigates the effect on the proposed algorithm of the scale parameter σ of Eqs. (13) and (12), introducing a variant of the presented clustering scheme exploiting that effect.

First of all, it is important to notice that σ controls the "softness" of the clustering scheme. Let us clarify this statement: a soft clustering scheme, by definition, assigns to each pattern a membership vector, which describes its similarities to every cluster. On the other hand, a hard clustering algorithm assigns a pattern just to one cluster (i.e., uses membership vectors with just one non-zero entry). Consider the membership vector for pattern i, obtained by the E-type step, denoted as $\mathbf{z}_i(\sigma)=[z_i^1(\sigma),\ldots,z_i^K(\sigma)]$ (where we have used notation that explicitly indicates the dependency on σ). Now, noting that we can write

$$\exp\left\{-\frac{\|\mathbf{x}-\mathbf{a}\|^2}{\sigma}\right\}=(\exp\{-\|\mathbf{x}-\mathbf{a}\|^2\})^{1/\sigma},$$
$$\exp\left\{\frac{d_k(\mathbf{x})}{\sigma}\right\}=(\exp\{d_k(\mathbf{x})\})^{1/\sigma},$$

it is easy to show that

$$\lim_{\sigma\to\infty}[z_i^1(\sigma),\ldots,z_i^K(\sigma)]=[\gamma_1,\ldots,\gamma_K]$$

and

$$\lim_{\sigma\to 0}\ z_i^k(\sigma)=\begin{cases}1 & \text{if } k=\arg\max_j\mathscr{S}(\mathbf{x}_i,j),\\ 0 & \text{otherwise.}\end{cases}$$

These limits show that when σ is very large, the memberships are essentially controlled by the cluster weights γ_k, whereas when σ becomes very small the membership vector specifies a hard partition.

The above described effect suggests the use of a determinist annealing version of the algorithm, controlled by the scale parameter σ, which is potentially able to improve robustness with respect to initialization [17,18]. The basic idea is to start the algorithm with a large σ: in this case the algorithm is working in a regime where the clustering is very soft (since the cluster weights are initialized to $\gamma_k=1/K$), so there are no local minima. Subsequently, σ is gradually decreased, thus hardening the assignment vectors. A similar approach has been exploited in Ref. [17], in order to improve the robustness of the standard EM with respect to initialization, and in Ref. [18] for vector quantization (equivalently K-means) algorithms.

The application of this idea to our algorithm leads to the insertion of an outer loop (the annealing loop), as follows:

(1) Initialization: as in the EM-like algorithm described in Section 3.2.
(2) set $\sigma = \sigma_{\max}$;
(3) repeat E-type and M-type steps until convergence;
(4) decrease σ;
(5) if $\sigma > \sigma_{\min}$ go to step (3), otherwise stop.

4. Experiments

4.1. Basic algorithm: fixed σ

The experimental evaluation was based on five well-known real datasets: the Iris dataset, the Wisconsin breast cancer (referred to as WSC), the Wine data set—all from the USC Machine Learning Repository[1] —the Pima Indian diabetes dataset[2] (referred to simply as Pima), and the Biomed dataset.[3]

The analyzed methods were implemented in Matlab, except the OC-SVM and WOC-SVM training algorithms. For the OC-SVM, we have used the LIBSVM software [19], which implements the version of Schölkopf et al. [11,12]. The training algorithm for the WOC-SVM was obtained by modifying the OC-SVM code from LIBSVM. When computing the similarity (13), the scale parameter σ was set to 1.

In the experiments, we compare our soft clustering method (which we named *WOC-SVM soft clustering* —WOC-SVM SC) with its hard clustering counterpart—which we call *OC-SVM hard clustering* (OC-SVM HC). In particular, in order to have a precise and fair comparison with our algorithm, we slightly adapted the algorithm in Ref. [8] to our case. The main modifications were: first, in our implementation we did not use the parameter ρ, which was used in Ref. [8] to discard the elements that are too distant in feature space. Second, in Ref. [8] the initialization was obtained by manually assigning some points to each cluster, whereas in the OC-SVM HC two different *automatic* initializations were introduced: one initializes each OC-SVM using random points (called "Random Init" in the table), while in the other (called "*K*-means init" in the table) a preliminary *K*-means algorithm is run, and the points assigned to each cluster are used to initialize the corresponding OC-SVM. All the experiments use a standard Gaussian radial basis function (GRBF) kernel of width τ. Different values for parameters ν and τ have been tested, choosing those leading to the best results. In particular, for each dataset, the optimal pair (τ, ν) was the same for all the kernel methods: $(0.85, 0.97)$ for Iris; $(0.2, 0.99)$ for WBC; $(0.01, 0.98)$ for Wine; $(0.1, 0.69)$ for Pima and $(0.002, 0.95)$ for Biomed.

All the algorithms were run 20 times and we report in Tables 1 and 2 the average, the minimum, and the maximum accuracies obtained over these 20 runs, for the proposed technique and for the corresponding hard clustering version, respectively. Since true labels are known, clustering accuracies could be quantitatively assessed. In particular, given a specific group, an error is considered when a pattern does not belong to the most frequent class inside the group. For the proposed soft clustering method (results shown in Table 1), we have considered several initialization methods: "Rand Pnts Init", in which each WOC-SVM is initialized using as positive examples a randomly selected subset of points; "Rand Wghts Init", in which each WOC-SVM is initialized using as positive examples all points with different random weights; and "GMM Init", in which a standard Gaussian mixture model is used to find the clusters, and the likelihoods are used as initial weights. From the tables it can be concluded that for random initializations, the proposed approach outperforms its hard version in all dataset (except Pima where it performs in line). Further, for clever initializations (*K*-means or GMM), the proposed method performs in line (in three cases) or above the hard clustering scheme (in two cases). Moreover, the differences between the maximum and minimum accuracies show that the proposed soft clustering algorithm is more stable and robust with respect to initialization (except in the Pima case). With three of these datasets (Iris, WBC and Wine), it seems that all initializations methods work equally well. For the other two, clever initializations lead to more stable results. The sensitivity of the hard clustering scheme with respect to initialization is confirmed by comparing results on Iris and WBC in Table 2 with the results reported in Ref. [8], where the variations were lower than those we found; recall that, in their method, a manual initialization was used. As a general comment, we have to say that parameter settings for kernel methods was quite difficult, confirming the need of proper model selection techniques.

In order to further assess the performance of the proposed algorithm in comparison with other methods, we include in Table 3 the average accuracy values reported in Ref. [8], obtained for the Iris and WBC datasets by other recent state-of-the-art algorithms: self-organizing maps (SOM) [20], neural gas [21], the Ng–Jordan spectral clustering algorithm [22]. Results of the classical GMM-based clustering and *K*-means clustering are also reported. It can be observed that the proposed approach is very competitive on these two datasets, achieving the best performance for WBC and the second best for Iris.

Table 1
Percentage accuracies of the experimental evaluation for the proposed WOC-SVM SCscheme, for three different initialization: initialization using random points ("Rand Pnts Init"), using random weights ("Rand Wghts Init") and using GMM clustering ("GMM Init")

Dataset	Accuracy (max–min)		
	Rand Pnts Init	Rand Wghts Init	GMM Init
Iris	92.0% (93.3–66.7%)	93.1% (93.3–90.7%)	93.3% (93.3–93.3%)
WBC	97.1% (97.1–97.1%)	96.5% (97.1–93.9%)	97.1% (97.1–97.1%)
Wine	94.5% (96.1–67.4%)	96.1% (96.1–96.1%)	96.1% (96.1–96.1%)
Pima	68.3% (77.0–37.0%)	71.8% (76.0–58.5%)	75.0% (75.0–75.0%)
Biomed	84.6% (90.2–72.2%)	84.3% (90.7–50.5%)	88.2% (88.7–88.1%)

Table 2
Percentage accuracies of the experimental evaluation for the OC-SVM HC scheme, for two different initialization: random initialization ("Random Init") and *K*-means initialization ("*K*-means init")

Dataset	Accuracy (max–min)	
	Random Init	*K*-Means Init
Iris	87.1% (96.0–33.3%)	91.6% (96.0–42.7%)
WBC	94.0% (97.1–35.0%)	97.1% (97.1–97.1%)
Wine	77.4% (96.1–41.6%)	96.1% (96.1–96.1%)
Pima	68.5% (70.0–65.5%)	70.0% (70.0–70.0%)
Biomed	81.2% (83.0–65.5%)	83.0% (83.0–83.0%)

Table 3
Average accuracies for different methods on the Iris and WBCdatasets

Method	Iris (%)	WBC (%)
Self organizing maps[a]	81.0	96.7
Neural gas[a]	91.7	96.1
Spectral clustering[a]	84.3	95.5
K-means[a]	89.0	96.1
GMM	89.3	94.6
Camastra–Verri[a]	94.7	97.0
Proposed method	93.3	97.1

[a]The results are taken from Ref. [8].

[1] Available at http://archive.ics.uci.edu/ml/datasets.html
[2] Available at www.stats.ox.ac.uk/pub/PRNN/
[3] Available at http://lib.stat.cmu.edu/datasets/

Table 4
Percentage accuracies of the experimental evaluation for the deterministic annealing variant of the proposed WOC-SVM SC scheme, described in Section 3.3

Dataset	Accuracy (max–min)		
	Rand Pnts Init	Rand Wghts Init	GMM Init
Iris	93.3% (93.3–93.3%)	93.3% (93.3–93.3%)	93.3% (93.3–93.3%)
WBC	97.1% (97.1–97.1%)	97.1% (97.1–97.1%)	97.1% (97.1–97.1%)
Wine	96.1% (96.1–96.1%)	96.1% (96.1–96.1%)	96.1% (96.1–96.1%)
Pima	70.0% (71.0–69.0%)	70.0% (70.5–69.0%)	70.0% (70.0–70.0%)
Biomed	81.3% (85.6–70.1%)	80.7% (86.1–69.6%)	85.6% (85.6–85.6%)

All the experimental conditions are the same.

4.2. Deterministic annealing version

In order to assess the suitability of the presented deterministic annealing version of the algorithm, we repeated the experiments described in the previous subsection, using the same configuration; note that, in order to have a fair comparison, *exactly the same* initializations were used by both versions of the algorithm. We set $\sigma_{\max} = 5$, $\sigma_{\min} = 0.1$, and the update rule of step 4 as $\sigma \leftarrow \sigma * 0.95$. The results obtained are reported in Table 4.

Comparing Tables 4 and 1, we can observe that in general there is not a substantial improvement in the averaged accuracies. Nevertheless, note that the minimum accuracy is almost always higher when applying the deterministic annealing version, thus showing an increased robustness with respect to initialization.

5. Concluding remarks and ongoing work

In this paper we have introduced a soft clustering algorithm based on one-class SVM (OC-SVM) representations of the clusters. The proposed method is based on a weighted version of the OC-SVM (termed WOC-SVM) which we have also introduced in this paper, and is inspired by the OC-SVM-based hard clustering algorithm proposed in Ref. [8]. Due to the use of WOC-SVM, the algorithm is directly kernelizable, thus constituting a kernel-based soft clustering method. Experimental results reported have shown that the proposed soft clustering algorithm outperforms the hard clustering counterpart, namely in what concerns robustness with respect to initialization. The proposed algorithm performs competitively with several state-of-the-art methods, including the spectral clustering algorithm of Ng and Jordan [22]. A deterministic annealing version has been also proposed, and shown to be able to provide robustness with respect to initialization.

As with most kernel-based methods, the performance of our method depends critically on the choice (and tuning of parameters) of the kernel. We are currently investigating approaches to adjust the kernel to the data in a more automatic way. Another research front concerns the development of model selection criteria under which the algorithm can select the number of clusters in the data.

Acknowledgments

The authors would thank Annalisa Barla for initial discussions on WOC-SVM and Cheng Dong Seon for precious help in adapting the LISBSVM code to the weighted case. The authors would also thank the anonymous reviewers for their helpful comments.

M. Figueiredo was partially supported by *Fundação para a Ciência e Tecnologia* (FCT), Portuguese Ministry of Science and Higher Education, under Grant POSC/EEA-SRI/61924/2004.

References

[1] V. Vapnik, The Nature of Statistical Learning Theory, Springer, Berlin, 1995.
[2] V. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.
[3] B. Schölkopf, A. Smola, Learning with Kernels, MIT Press, Cambridge, MA, 2002.
[4] J. Shawe-Taylor, N. Cristianini, Kernel Methods for Pattern Analysis, Cambridge University Press, Cambridge, 2004.
[5] A. Ben-Hur, D. Horn, H. Siegelmann, V. Vapnik, Support vector clustering, J. Mach. Learn. Res. 2 (2001) 125–137.
[6] J. Yang, V. Estivill-Castro, S. Chalup, Support vector clustering through proximity graph modelling, Adv. Neural Inf. Process. Syst. (2002) 898–903.
[7] J. Lee, D. Lee, An improved cluster labeling method for support vector clustering, IEEE Trans. Pattern Anal. Mach. Intell. 27 (2005) 461–464.
[8] F. Camastra, A. Verri, A novel kernel method for clustering, IEEE Trans. Pattern Anal. Mach. Intell. 27 (2005) 801–805.
[9] J. Lee, D. Lee, Dynamic characterization of cluster structures for robust and inductive support vector clustering, IEEE Trans. Pattern Anal. Mach. Intell. 28 (2006) 1869–1874.
[10] D. Tax, R. Duin, Support vector domain description, Pattern Recognition Lett. 20 (1999) 1191–1199.
[11] B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylor, J. Platt, Support vector method for novelty detection, Adv. Neural Inf. Process. Syt. (1999) 526–532.
[12] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. Smola, R. Williamson, Estimating the support of a high-dimensional distribution, Neural Comput. 13 (2001) 1443–1447.
[13] A. Jain, R. Dubes, Algorithms for Clustering Data, Prentice-Hall, Englewood Cliffs, NJ, 1988.
[14] M. Figueiredo, A. Jain, Unsupervised learning of finite mixture models, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002) 381–396.
[15] A. Dempster, N. Laird, D. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. R. Statist. Soc. (B) 39 (1977) 1–38.
[16] X. Wu, R. Srihari, Incorporating prior knowledge with weighted margin support vector machines, in: Proceeedings of the ACM International Conference on Knowledge Discovery and Data Mining, 2004, pp. 326–333.
[17] N. Ueda, R. Nakano, Deterministic annealing EM algorithm, Neural Networks 11 (1998) 271–282.
[18] K. Rose, Deterministic annealing for clustering, compression, classification, regression, and related optimization problems, Proc. IEEE 80 (1998) 2210–2239.
[19] C.-C. Chang, C.-J. Lin, Libsvm: a library for support vector machines. Software, 2001 available at ⟨http://www.csie.ntu.edu.tw/~cjlin/libsvm⟩.
[20] T. Kohonen, Self-Organizing Maps, Springer, Berlin, 1997.
[21] T. Martinetz, K. Schulten, Neural-gas network for vector quantization and its application to time-series prediction, IEEE Trans. Neural Networks 4 (1993) 558–569.
[22] A. Ng, M. Jordan, Y. Weiss, On spectral clustering: analysis and an algorithm, Adv. Neural Inf. Process. Syst. (2001) 849–856.

About the Author—MANUELE BICEGO received his Laurea degree and PhD degree in Computer Science from University of Verona in 1999 and 2003, respectively. Now he is a researcher at the University of Sassari. His research interests include statistical pattern recognition, hidden Markov models, video analysis and biometrics. Since 2004 he is an associate editor of the ELCVIA international journal, and he was a guest editor of the special issue of Pattern Recognition on "Similarity Based Pattern Recognition" (2006). He has served as a member of the scientific committee of different international conferences, and he is a reviewer for several international conferences and journals.

About the Author—MÁRIO FIGUEIREDO holds a PhD (1994) from IST, Technical University of Lisbon, and is now a professor at IST and a researcher at Instituto de Telecomunicações. He is an associate editor of several journals, including IEEE-TPAMI and IEEE-TIP, and was a guest editor of several special issues. He co-chaired several workshops and has been in program committees of many conferences, including NIPS, ICML, CVPR, EECV, ICIP, ICPR.