
Laboratorio di Immagini

Esercitazione 6:

Segmentation & PR

Mauro Zucchelli
28/04/2016

Split & Merge

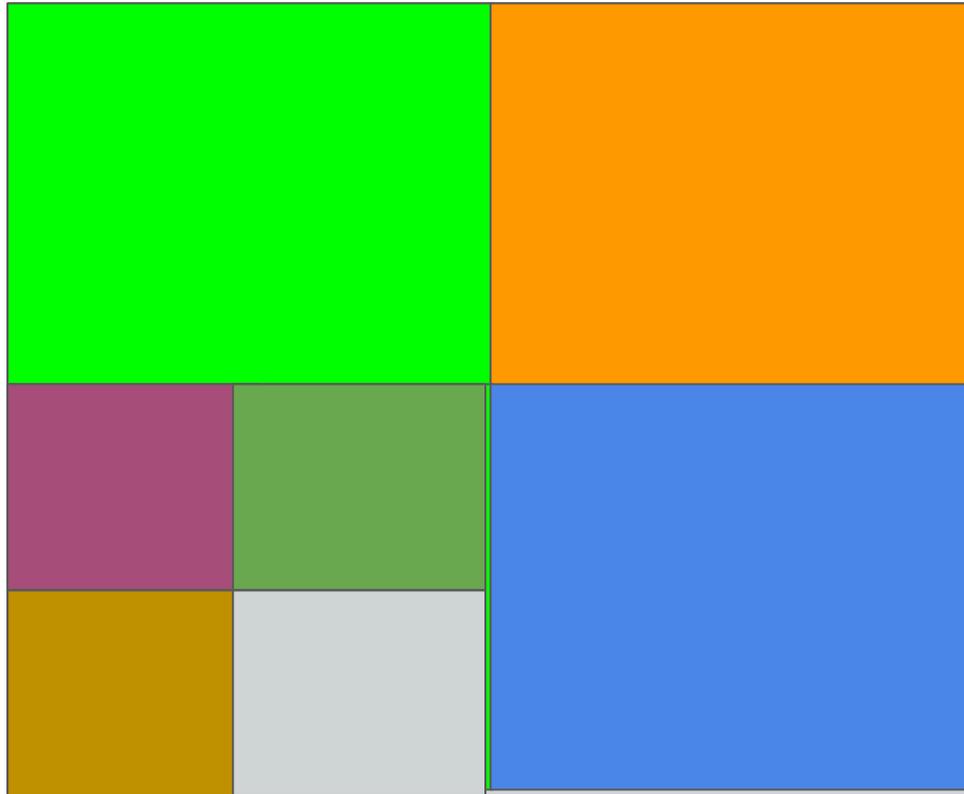
- Algoritmo di segmentazione
- Usa informazioni sia di livello di grigio sia di vicinanza
- Può essere definito in maniera ricorsiva



Fase 1: split

- 1) Considero l'immagine
 - 2) Se la **varianza** dell'immagine è bassa l'algoritmo inserisce l'immagine tra le regioni di interesse
 - 3) Altrimenti, se la varianza è sopra soglia divido l'immagine in **4** blocchi
 - 4) Per ogni blocco riprendo dal **passo 1**
 - 5) L'algoritmo termina quando tutti i blocchi sono uniformi
-

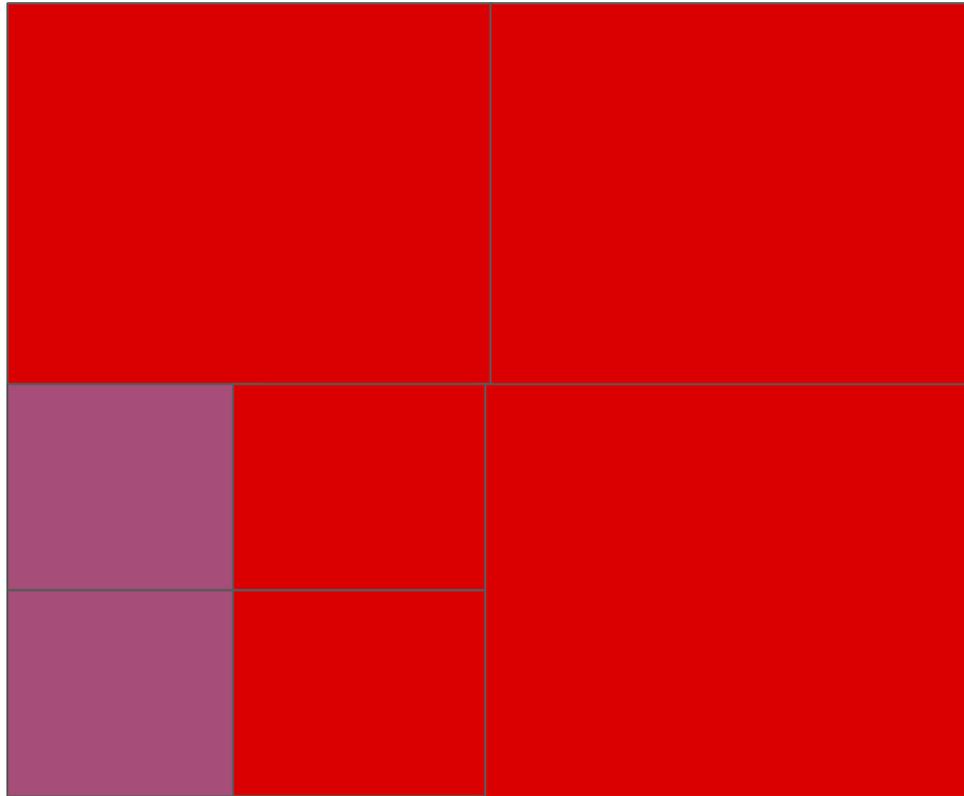
Fase 1: split



Fase 2: merge

- 1) Considero una regione di interesse
 - 2) La confronto con le altre regioni di interesse
 - 3) Se due regioni sono simili (per media e varianza) e sono **contigue** le unisco
 - 4) L'algoritmo termina quando non si possono più unire due regioni
-

Fase 2: merge



Region Growing

- Algoritmo di segmentazione
 - Usa informazioni sia di livello di grigio sia di vicinanza
 - Molto usato
-

Region growing

- 1) Si parte da un punto detto “**seed**” che fornisce il punto di partenza della nostra ROI
 - 2) Si guardano i punti vicini al bordo della ROI **spazialmente**
 - 3) Tutti i punti vicini che sono anche vicini come livello di grigio si **uniscono** alla ROI
 - 4) Gli altri si scartano e si riprende dal punto 2
 - 5) L’algoritmo termina quando nessun punto vicino alla ROI ha un livello di grigio compatibile
-

In matlab

```
img =imread('coronal.png');
```

```
figure
```

```
imagesc(img)
```

```
% input del seed con un click
```

```
[y,x] = ginput(1);
```

```
x = floor(x);
```

```
y = floor(y);
```

```
R = my_region_growing(img, x, y, 25);
```

```
figure
```

```
subplot(1,2,1)
```

```
imshow(img,[])
```

```
subplot(1,2,2)
```

```
imshow(R,[])
```

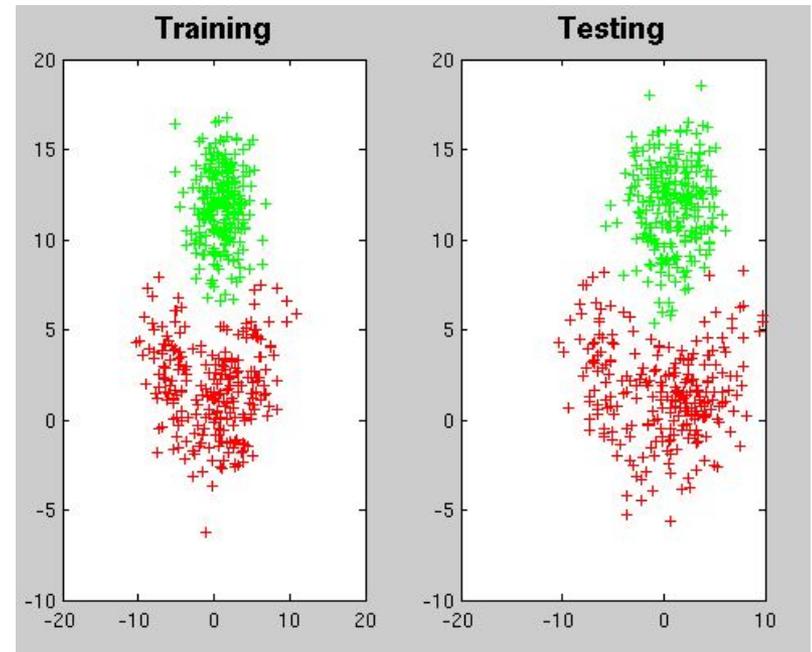


K-NN

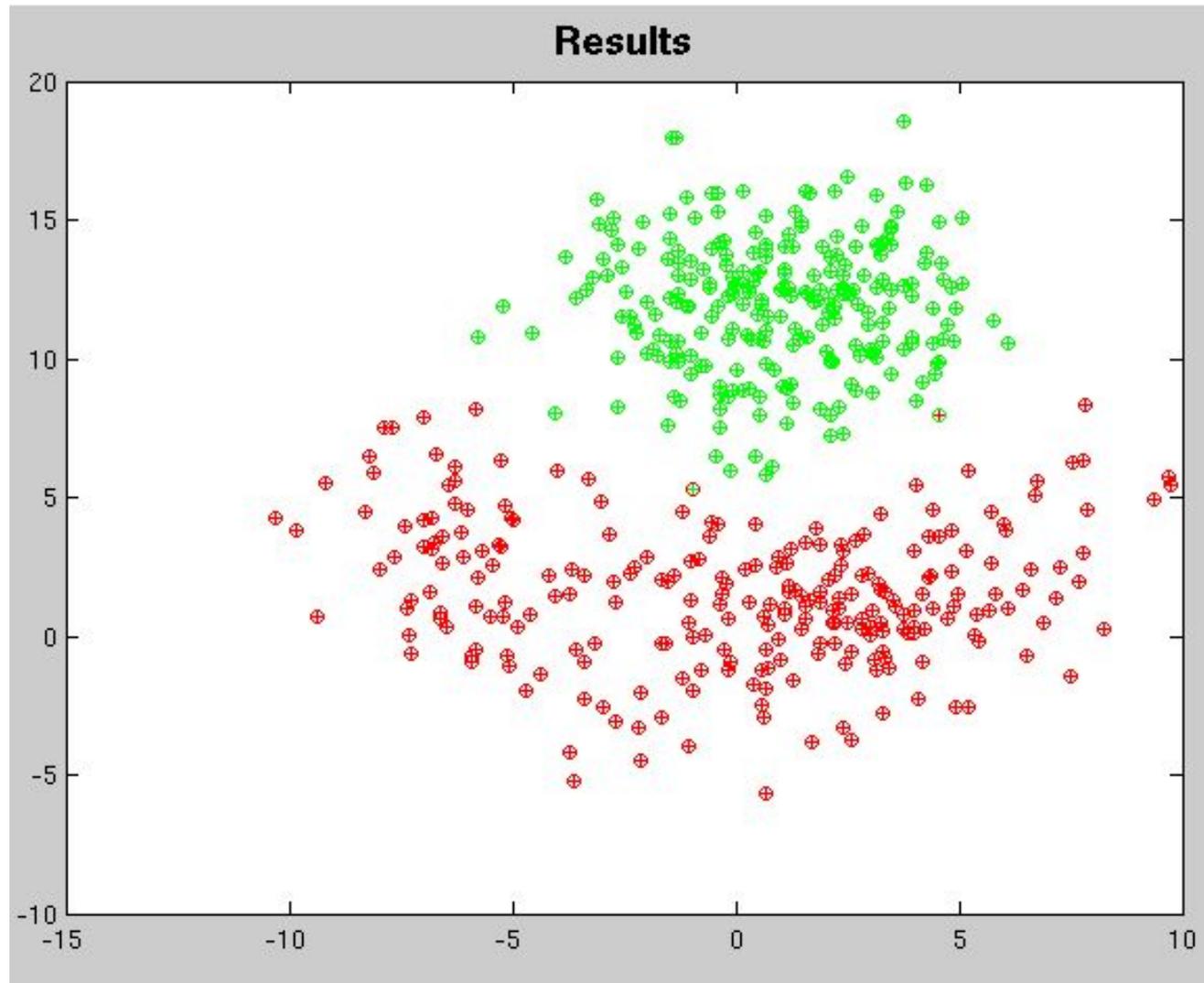
```
load('Training.mat');
load('Testing.mat');
figure
subplot(1,2,1)
plot(Train(label==0,1),Train(label==0,2),'r+');
hold on;
plot(Train(label==1,1),Train(label==1,2),'g+');
hold off
title('Training')

subplot(1,2,2)
plot(Test(label==0,1),Test(label==0,2),'r+');
hold on;
plot(Test(label==1,1),Test(label==1,2),'g+');
hold off
title('Testing')

labels_test = my_knn(Train,label,Test,11);
```



K-NN



K-NN - NOISY DATASET

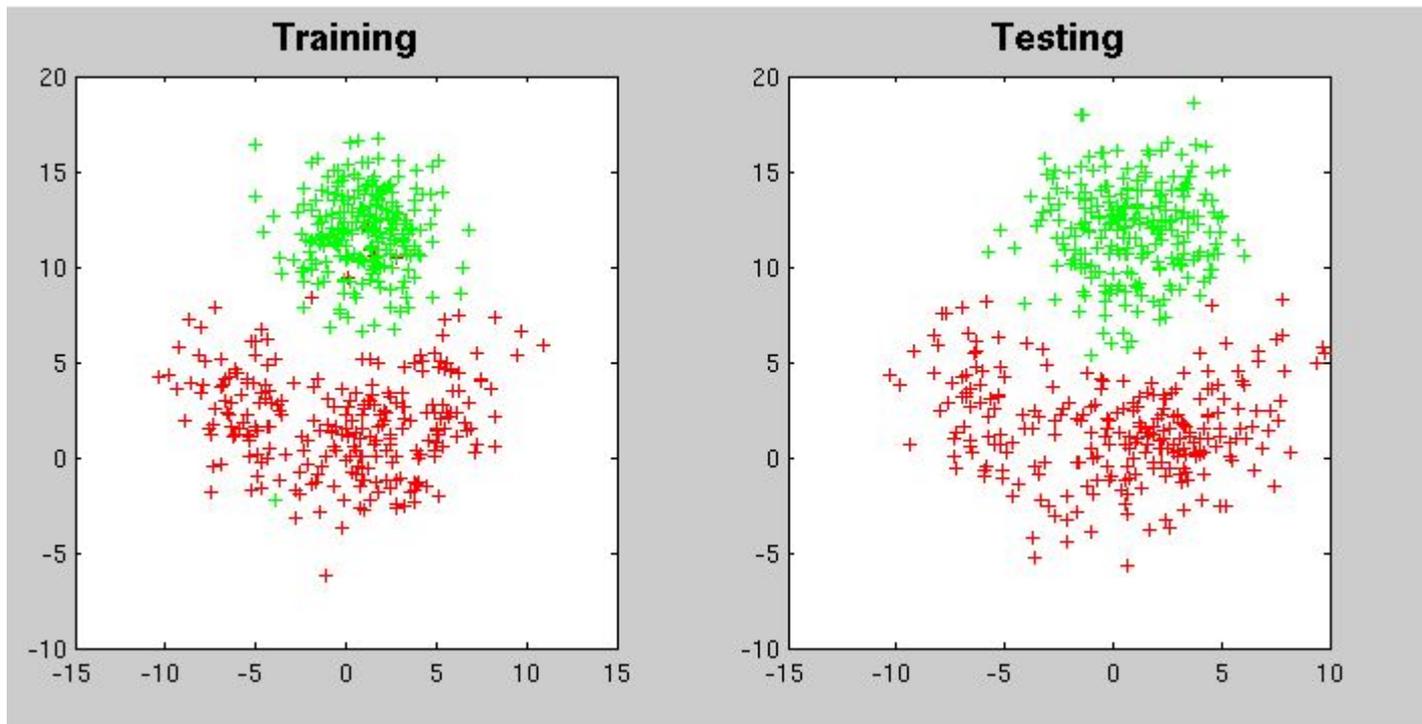
- Scombiniamo un po' il dataset di training!

`l = length(label);`

`label2 = label;`

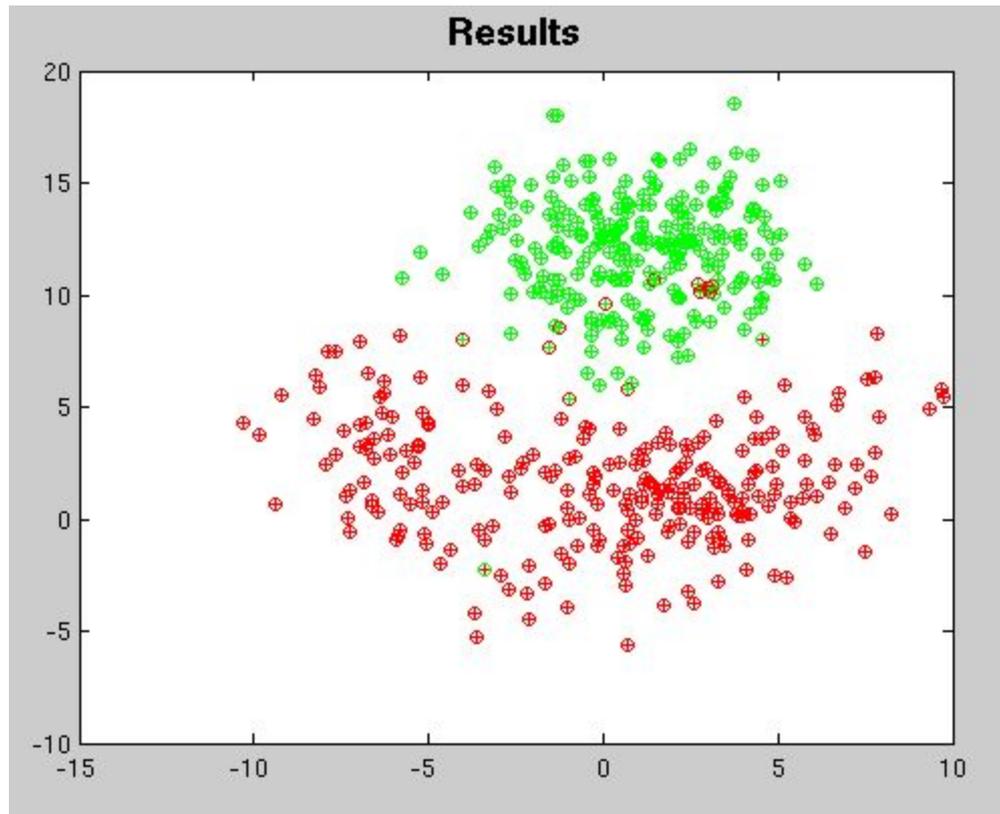
`r = rand(l,1) > 0.98;`

`label2(r) = abs(label2(r)-1);`



K-NN - NOISY DATASET

- K troppo bassi portano ad errori di classificazione



K-MEANS

- Algoritmo di clustering
- Possiamo clusterizzare i colori!

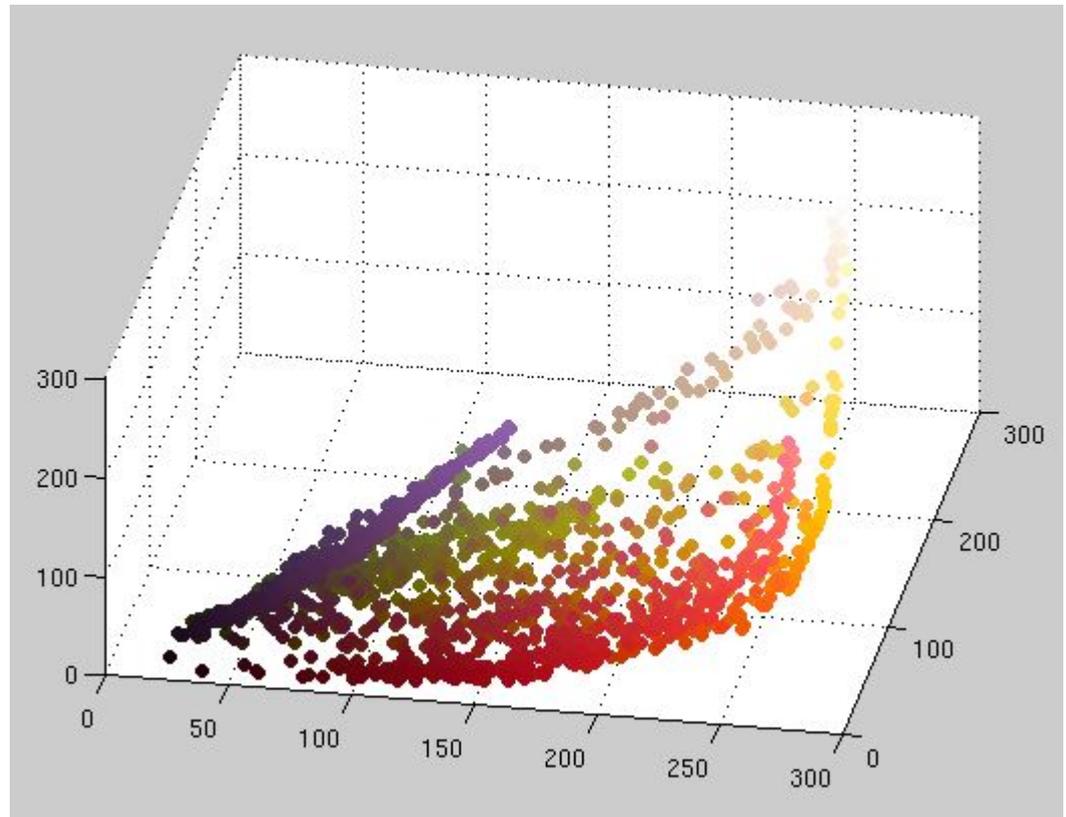


K-MEANS

- Visualizziamo lo spazio colori

```
img = imread('peppers.png');  
img = imresize(img,0.1);  
[x,y,c] = size(img);
```

```
img2 = reshape(img,[x*y,c]);
```



```
figure;  
scatter3(img2(:,1),img2(:,2),img2(:,3),[], double(img2)/256.0,'filled')
```

K-NN - NOISY DATASET

- K-MEANS con inizializzazione random

```
k = 5;
```

```
M = (rand(k,3)*255);
```

```
labels= my_kmeans(img2, M,k);
```

```
labels = reshape(labels,[x,y,3]);
```

```
figure
```

```
subplot(1,2,1)
```

```
imshow(img)
```

```
subplot(1,2,2)
```

```
imshow(labels)
```



FINE!

The End
