Machine Learning

Week 8: Unsupervised Learning (Cont'd) Support Vector Machines

Mahesan Niranjan

School of Electronics and Computer Science
University of Southampton

Autumn Semester 2015/16

Gaussian Mixture Model and K-Means Clustering

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_j \mathcal{N}(\boldsymbol{\mu}_k, \Sigma_k) \quad \pi_j \geq 0, \sum_{J=1}^{K} \pi_j = 1.$$

Input: $\boldsymbol{X} = \{ \boldsymbol{x}_n^t \}_{n=1}^N, K$
Output: $\boldsymbol{C}$, `Idx`
`initialize:` $\boldsymbol{C} = \left\{ \boldsymbol{c}_j^t \right\}_{j=1}^K$

`repeat`
. assign n^{th} sample to nearest $\boldsymbol{c}_j$
. $\texttt{Idx(n)} = \min_j ||\boldsymbol{x}_n - \boldsymbol{c}_j||^2$

. recompute $\boldsymbol{c}_j = \frac{1}{N_j} \sum_{n=j} \boldsymbol{x}_n$
`until` no change in $\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots \boldsymbol{c}_k$

`return` $\boldsymbol{C}$, `Idx`

Objective Function for Clustering

Setting up an error function and minimizing it

$$J_e = \sum_{i=1}^{K} \sum_{\boldsymbol{x} \in \mathcal{D}_i} ||\boldsymbol{x} - \boldsymbol{m}_i||^2$$

$$\boldsymbol{m}_i = \frac{1}{n_i} \sum_{\boldsymbol{x} \in \mathcal{D}_i} \boldsymbol{x}$$

Which is also the same as (in terms of scatter)

$$J_e = \frac{1}{2} \sum_{i=1}^{K} n_i \, \bar{s}_i$$

$$\bar{s}_i = \frac{1}{n_i^2} \sum_{\boldsymbol{x} \in \mathcal{D}_i} \sum_{\boldsymbol{y} \in \mathcal{D}_i} ||\boldsymbol{x} - \boldsymbol{y}||^2$$

Homework: Show this *i.e.* sum of average distance to cluster means and sum of within cluster scatter are the same.

Iterative Optimization

Note: Discrete optimzation

$$\begin{aligned} J_e &= \sum_{i=1}^{K} J_i \\ &= \sum_{i=1}^{K} \sum_{\boldsymbol{x} \in \mathcal{D}_i} ||\boldsymbol{x} - \boldsymbol{m}_i||^2 \end{aligned}$$

- Mean of each cluster: $\boldsymbol{m}_i = \sum_{\boldsymbol{x} \in \mathcal{D}_i} \boldsymbol{x}$
- Move sample (data) $\widehat{\boldsymbol{x}}$ from cluster $\mathcal{D}_i$ to $\mathcal{D}_j$; Say new J_j is J_j^* and new $\boldsymbol{m}_j$ is $\boldsymbol{m}_j^*$

$$\boldsymbol{m}_j^* = \boldsymbol{m}_j + \frac{1}{n_j + 1} (\widehat{\boldsymbol{x}} - \boldsymbol{m}_j)$$

$$\begin{aligned} J_j^* &= \sum_{\boldsymbol{x} \in \mathcal{D}_j} \left\| \boldsymbol{x} - \boldsymbol{m}_j^* \right\|^2 + \left\| \widehat{\boldsymbol{x}} - \boldsymbol{m}_j^* \right\|^2 \\ &= \left(\sum_{\boldsymbol{x} \in \mathcal{D}_j} \left\| \boldsymbol{x} - \boldsymbol{m}_j - \frac{1}{n_j + 1} (\widehat{\boldsymbol{x}} - \boldsymbol{m}_j) \right\|^2 \right) = \left\| \frac{n_j}{n_j + 1} (\widehat{\boldsymbol{x}} - \boldsymbol{m}_j) \right\|^2 \\ &= J_j + \frac{n_j}{n_j + 1} ||\widehat{\boldsymbol{x}} - \boldsymbol{m}_j||^2 \end{aligned}$$

Iterative optimization (cont'd)

...similarly, $\boldsymbol{m}_i$ changes to

$$\boldsymbol{m}_i^* = \boldsymbol{m}_i - \frac{1}{n_i - 1}(\widehat{\boldsymbol{x}} - \boldsymbol{m}_i)$$

$$J_i = J_i - \frac{n_i}{n_i - 1} \|\widehat{\boldsymbol{x}} - \boldsymbol{m}_i\|^2$$

So, if

$$\frac{n_i}{n_i - 1} \|\widehat{\boldsymbol{x}} - \boldsymbol{m}_i\|^2 > \frac{n_j}{n_j + 1} \|\widehat{\boldsymbol{x}} - \boldsymbol{m}_j\|^2$$

then it is advantageous to move $\widehat{\boldsymbol{x}}$ from $\mathcal{D}_i$ to $\mathcal{D}_i$

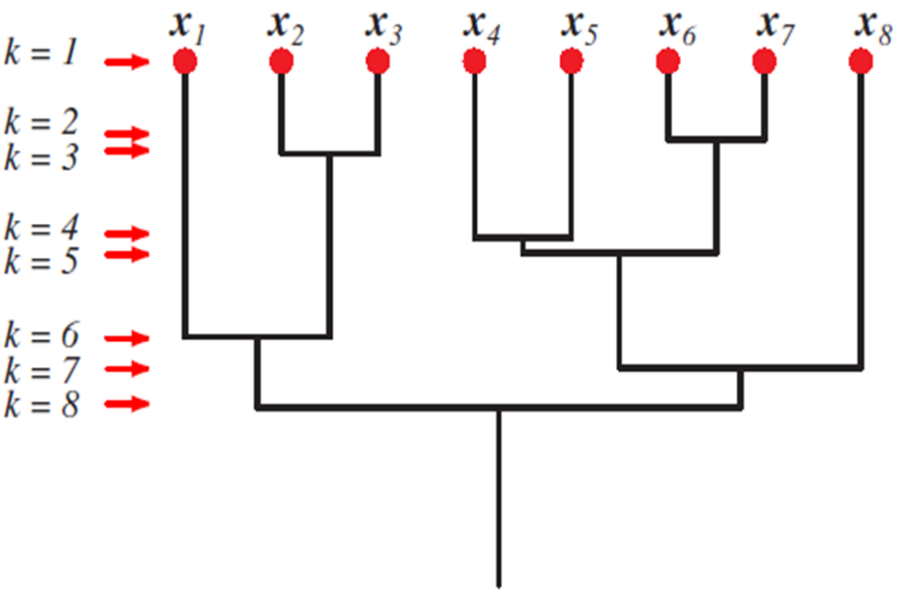
Algorithm:

- Select a data point at random
- Move it to cluster for which $\frac{n_j}{N_j+1} \|\widehat{\boldsymbol{x}} - \boldsymbol{m}_j\|^2$ is minimum.
- Recalculate means $\boldsymbol{m}_i$, i=1,...,K

This will be a sequential version of $K-$means algorithm; *i.e.* update at each data, rather than wait till we classify all data.

Hierarchical Clustering

Dendrogram



Homework: Use MATLAB to draw a dendrogram for the Boston Housing data

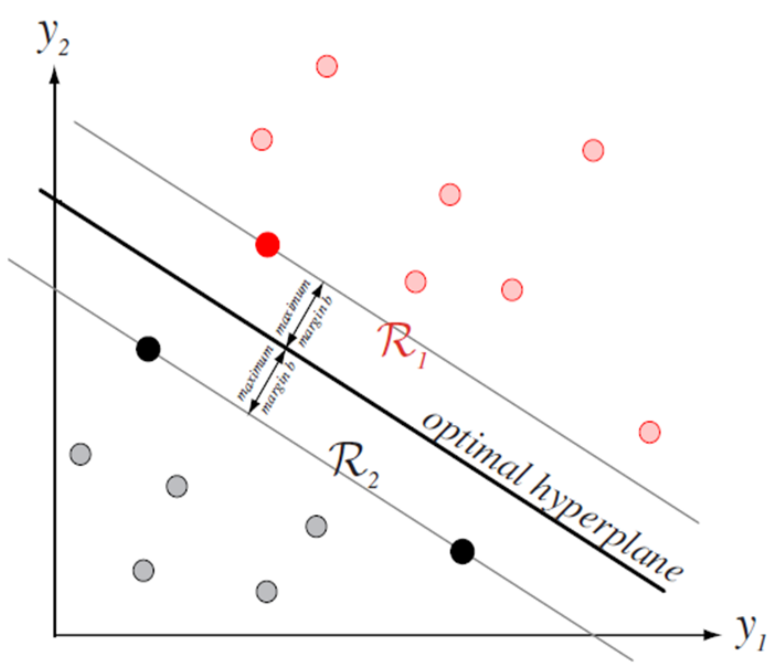
Agglomerative Hierarchical Clustering

- Initialize: $\widehat{K} = n$ (No. clusters = No. data)
- Repeat (until $\widehat{K} = K$)
  - find nearest clusters $\mathcal{D}_i$ and $\mathcal{D}_j$
  - merge $\mathcal{D}_i$ and $\mathcal{D}_j$
  - $\widehat{c} \leftarrow \widehat{c} - 1$

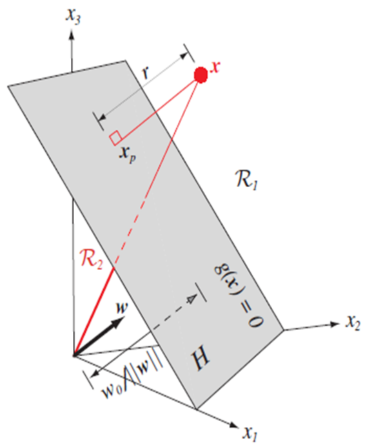
Defining nearest clusters

$$
\begin{aligned}
D_{\min}\left(\mathcal{D}_i, \mathcal{D}_j\right) &= \min_{\boldsymbol{x}\in\mathcal{D}_i \boldsymbol{y}\in\mathcal{D}_j} \|\boldsymbol{x} - \boldsymbol{y}\|^2 \\
D_{\text{avg}}\left(\mathcal{D}_i, \mathcal{D}_j\right) &= \frac{1}{n_i n_j} \sum_{\boldsymbol{x}\in\mathcal{D}_{\rangle}} \sum_{\boldsymbol{y}\in\mathcal{D}_j} \|\boldsymbol{x} - \boldsymbol{y}\|^2
\end{aligned}
$$

Now for something completely different!



Margin



(b in formula is w_0 in figure)

- Hyperplane: $\boldsymbol{w}^t \, \boldsymbol{x} \, + \, b \, = \, 0$ See Lab 2
  vfill
- Data:

$$\mathcal{D} \, = \, \{\boldsymbol{x}_n, \, y_n\}_{n=1}^N \, , \, \boldsymbol{x}_n \in \mathcal{R}^d, \, y_n \in \{-1, \, +1\}$$

  vfill
- Learning problem:

$$y_n \left[\boldsymbol{w}^t \, \boldsymbol{x}_n \, + \, b \right] \geq 1, \; n = 1, ..., N$$

Margin

- Distance from data $\boldsymbol{x}_n$ to a hyperplane $(\boldsymbol{w}, b)$:

$$d(\boldsymbol{w}, b, \boldsymbol{x}_n) \, = \, \frac{|\boldsymbol{w}^t \boldsymbol{x}_n + b|}{||\boldsymbol{w}||}$$

- The margin – distance between data closest to the hyperplane on either side

$$\begin{aligned}
\rho(\boldsymbol{w}, b) \; &= \; \min_{\boldsymbol{x}_n : y_n = -1} d(\boldsymbol{w}, b, \boldsymbol{x}_n) \, + \, \min_{\boldsymbol{x}_n : y_n = +1} d(\boldsymbol{w}, b, \boldsymbol{x}_n) \\
&= \; \min_{\boldsymbol{x}_n : y_n = -1} \frac{|\boldsymbol{w}^t \boldsymbol{x}_n + b|}{||\boldsymbol{w}||} \, + \, \min_{\boldsymbol{x}_n : y_n = +1} \frac{|\boldsymbol{w}^t \boldsymbol{x}_n + b|}{||\boldsymbol{w}||} \\
&= \; \frac{1}{||\boldsymbol{w}||} \left(\min_{\boldsymbol{x}_n : y_n = -1} |\boldsymbol{w}^t \boldsymbol{x}_n + b| \, + \, \min_{\boldsymbol{x}_n : y_n = +1} |\boldsymbol{w}^t \boldsymbol{x}_n + b| \right) \\
&= \; \frac{2}{||\boldsymbol{w}||}
\end{aligned}$$

Lagrangian for SVM Classification

$$\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}||\boldsymbol{w}||^2 - \sum_{n=1}^{N} \alpha_n \left(y_n \left[\boldsymbol{w}^t \boldsymbol{x}_n + b \right] - 1 \right), \; \alpha_n \geq 0$$

- Setting $\frac{\partial \mathcal{L}}{\partial b}$ to zero, gives $\sum_{n=1}^{N} \alpha_n y_n = 0$
- Setting $\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}}$ to zero, gives $\boldsymbol{w} = \sum_{n=1}^{N} \alpha_n y_n \boldsymbol{x}_n$
- Note: the unknown weights are computed as a weighted sum of the training examples; do you see a similarity to the perceptron algorithm?
- Substitute to get the dual problem

$$\max_{\boldsymbol{\alpha}} \; -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^t \boldsymbol{x}_j + \sum_{k=1}^{N} \alpha_k$$

$$\min_{\boldsymbol{\alpha}} \; \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^t \boldsymbol{x}_j - \sum_{k=1}^{N} \alpha_k$$

$$\text{subject to } \alpha_n \geq 0 \text{ and } \sum_{n=1}^{N} \alpha_n y_n = 0$$

- Quadratic programming

```
MATLAB> help quadprog
```

$$\min_{\boldsymbol{x}} \; \frac{1}{2} \boldsymbol{x}^t \boldsymbol{H} \boldsymbol{x} + \boldsymbol{f}^t \boldsymbol{x}$$

Subject to

$$\begin{aligned} \boldsymbol{A}\boldsymbol{x} &\leq \boldsymbol{b} \\ \boldsymbol{A}_{\text{eq}}\boldsymbol{x} &= \boldsymbol{b}_{\text{eq}} \\ lb \leq \boldsymbol{x} &\leq ub \end{aligned}$$

```
MATLAB> x = quadprog( H,f,A,b,Aeq,beq,lb,ub );
```

Calculating the Bias Term

- Constraints $\alpha_n \geq 0$; Parameters $\boldsymbol{w} = \sum_{n=1}^{N} y_n \alpha_n \boldsymbol{x}_n$
- Non-zero α_n's correspond to Support Vectors
- For any of these support vectors ($\boldsymbol{x}_s$): $y_s[\boldsymbol{w}^t \boldsymbol{x}_s + b] = 1$; we can compute the bias term b from this.

$$y_s \left[\sum_{m \in \mathcal{S}} \alpha_m y_m \boldsymbol{x}_m^t \boldsymbol{x}_s \, + \, b \right] \, = 1$$

$$y_s^2 \left(\sum_{m \in \mathcal{S}} \alpha_m y_m \boldsymbol{x}_m^t \, \boldsymbol{x}_s \, + \, b \right) \, = y_s$$

$$\text{Note}: y_s^2 = 1; \ \text{Hence} \ b \, = \, y_s - \sum_{m \in \mathcal{S}} \alpha_m y_m \boldsymbol{x}_m^t \boldsymbol{x}_s$$

- In practice, instead of using any one support vector, use we average:

$$b = \frac{1}{N_s} \sum_{s \in \mathcal{S}} \left(y_s - \sum_{m \in \mathcal{S}} \alpha_m y_m \boldsymbol{x}_m^t \boldsymbol{x}_s \right)$$