Recitation 8

Ensemble Learning, Recommender Systems, SVMs, Graphical Models

10-601: Introduction to Machine Learning

04/26/2019

1 Ensemble Learning

Reminder:

Ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone. In the lecture, we have talked about:

- Weighted Majority Algorithm, which is a typical example of ensemble method. It assumes we have a bunch of learned weak classifiers, and it only learns (majority vote) weight for each classifiers.
- AdaBoost is an example of a boosting method, and boosting is a typical type of ensemble method.It simultaneously learns the weak classifiers and (majority vote) weight for each classifiers.

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in X, y_i \in Y = \{-1, +1\}$
Initialize $D_1(i) = 1/m$.
For $t = 1, \ldots, T$:

- Train weak learner using distribution D_t.
- Get weak hypothesis $h_t : X \to \{-1, +1\}$ with error

$$\epsilon_t = \Pr_{i \sim D_t}\left[h_t(x_i) \neq y_i\right].$$

- Choose $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$.
- Update:

$$\begin{aligned} D_{t+1}(i) &= \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases} \\ &= \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t} \end{aligned}$$

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis:

$$H(x) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right).$$

Applying adaboost algorithm for two rounds and answer the following questions.

X	0	1	2	3	4	5	6	7	8	9
Y	+	+	+	-	-	-	+	+	+	-

Throughout this question, assume our weak learner is a depth-1 decision stump (threshold classifier).

1. What is the error rate at the first round?

2. What are the weights for the samples after the first round (choose the smallest boundary to break tie)?

3. What is the error rate at the second round?

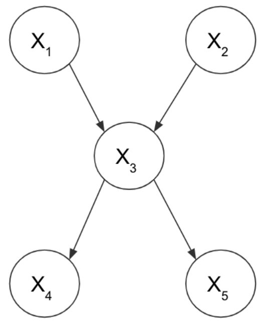
2 Recommender Systems

1. Reminder: we are taught with the following concepts in class.
    - Recommender systems: Answer to the question "Can represent ratings numerically as a user/item matrix?"
        - Content Filtering
        - Collaborative Filtering: The assumption is that personal tastes are correlated (e.g. Bestseller lists, Top 40 music lists, etc.).
            * Neighborhood Methods: Recommend movies that those neighbors (based on similarity of movie preferences) watched.
            * Latent Factor Methods (e.g. Matrix Factorization): Assume that both movies and users live in some low-dimensional space describing their properties.
2. For collaborative filtering algorithm, user tastes must either be generally stable or if changing, they must change in sync with other users' tastes. [True or False]

3. Collaborative filtering would be better suited for the following situation than content filtering:
    - The items being recommended don't have good attributes or key words to describe them (e.g., children's drawings without tags).[True or False]
    - Explicit ratings are not available for new items.[True or False]

4. What it the objective function for Unconstrained Matrix Factorization?

3 Bayes Net

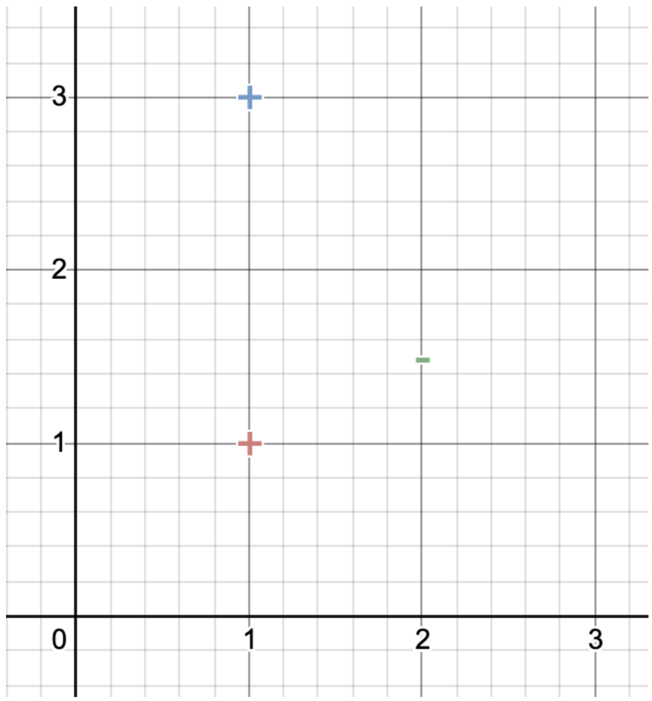


1. Write down the factorization of the above directed graphical model.

2. Given X_3, what are the relationships(independent or not) between the random variables listed below?
   - X_1_____$X_4 \mid X_3$
   - X_1_____$X_2 \mid X_3$
   - X_4_____$X_5 \mid X_3$

4 SVMs

1. What is the decision boundary and the margin if we run a Hard-Margin SVM on the following set of points?



2. A few additional data points are added to the data set in figures 2 (a) and 2 (b). Draw the new decision boundaries and give the margins corresponding to this boundaries. In which case does the decision boundary undergo a change and why?

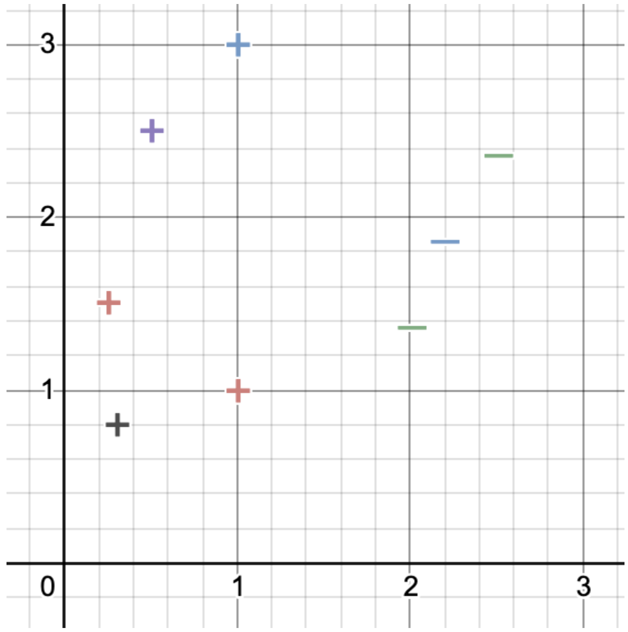


Figure 2(a)

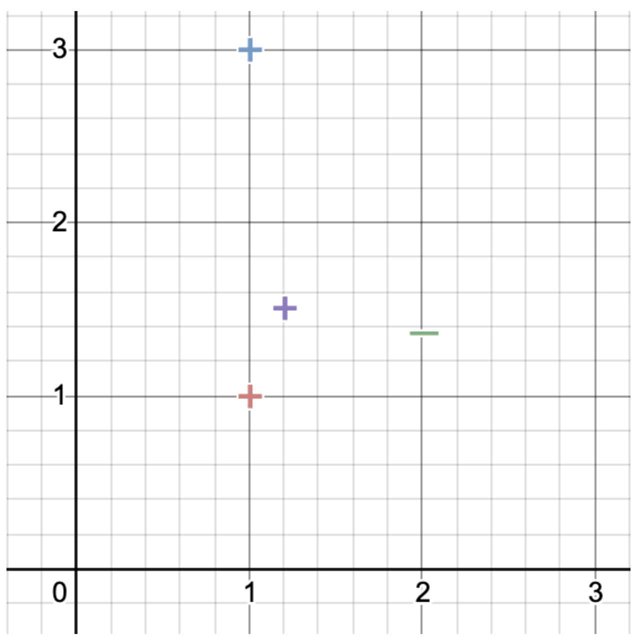


Figure 2(b)

5 Principal Component Analysis (Take Home)

1. The data set we will be working with is given by $\mathbf{D} \in \mathbb{R}^{n \times m}$ where n is the number of data points (5) and m is the number of the features (2). Given $\mathbf{D} = \begin{bmatrix} 2 & 1 \\ 3 & 4 \\ 5 & 0 \\ 7 & 6 \\ 9 & 2 \end{bmatrix}$, let's **center** the data. Centering is simply subtracting the mean of every feature from the data points.

   $\mathbf{D_c} = \begin{bmatrix} -3.2 & -1.6 \\ -2.2 & 1.4 \\ -0.2 & -2.6 \\ 1.8 & 3.4 \\ 3.8 & -0.6 \end{bmatrix}$

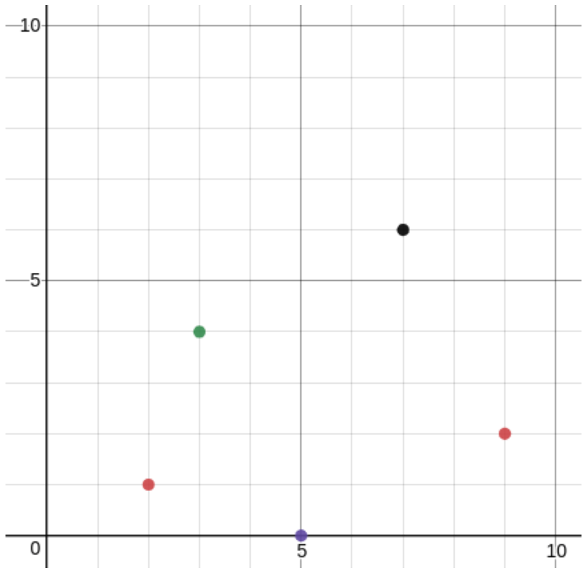
2. Let's call this centered data set $\mathbf{D_c}$. Now, we would like to find the co-variance between the features. Recall the co-variance matrix($\mathbf{S}$) is given by $\frac{\mathbf{D_c}^T . \mathbf{D_c}}{n-1}$

   $\mathbf{S} = \begin{bmatrix} 8.2 & 1.6 \\ 1.6 & 5.8 \end{bmatrix}$

3. Then, we decompose $\mathbf{S}$ into its Eigenvalues($\boldsymbol{\lambda}_k$) and Eigenvectors($\mathbf{V}_k$) where $k \in [0, m]$. Recall we perform eigen decomposition by solving $det(\mathbf{S} - \boldsymbol{\lambda}\mathbf{I}) = 0$ to get the eigenvalues and then solving $\mathbf{SV} = \boldsymbol{\lambda}\mathbf{V}$ to get the eigenvectors.

   $\boldsymbol{\lambda} = 9, 5$ $\mathbf{V} = \begin{bmatrix} 0.8944 & -0.4472 \\ 0.4472 & 0.8944 \end{bmatrix}$

4. Finally, the Eigenvector corresponding to the largest Eigenvalue is our first Principal Component given by PC_1. This is the primary axis of our transformed feature space. Plot on the graph below, the axes PC_1 and PC_2. Remember, the heading of an axis is given by the corresponding eigenvector and its magnitude is given by the corresponding eigen value.


10
5
0
5
10