CS 391D Data Mining: A Mathematical Perspective Fall 2009

Solutions to Homework 2

Lecturer: Inderjit Dhillon Date Due: October 8, 2009

Keywords: *Probability, Principal Component Analysis, Classification*

1. (5 points)

   Let random variable B represent the selected "Bag", which can take values from $\{b_1, b_2, b_3\}$. Similarly, let random variable C represent the selected "candy", which can take values from $\{a, c, o\}$, where $a = \{\text{selected candy is an } \textit{apple}\}$, $c = \{\text{selected candy is a } \textit{cherry}\}$, and $o = \{\text{selected candy is an } \textit{orange}\}$. By this definition, we can use simple notation to denote the probability of an event happened. For example, the probability of selecting an *apple* can be represented by $p(C = a)$. For simplicity, we use $p(a)$ instead of $p(C = a)$ when there is no confusion.

   (a) (1 point)

   $$p(o) = \sum_{i=1}^{3} p(o|b_i)p(b_i) = 0.3 \times 0.2 + 0.5 \times 0.2 + 0.3 \times 0.6 = 0.34.$$

   (b) (2 points)

   $$p(b_2|o) = \frac{p(o|b_2)p(b_2)}{\sum_{i=1}^{3} p(o|b_i)p(b_i)} = \frac{0.5 \times 0.2}{0.34} = 5/17 \approx 0.29.$$

   (c) (2 points) Let o_1 denote the event that the first selected candy is an *orange* and o_2 denote the event that the second selected candy is an *orange*. Then this problem is asking what is the value of $p(o_2|o_1)$. Similar to problem (b), we can compute

   $$p(b_1|o) = \frac{p(o|b_1)p(b_1)}{\sum_{i=1}^{3} p(o|b_i)p(b_i)} = 3/17 \approx 0.18,$$

   and

   $$p(b_3|o) = \frac{p(o|b_3)p(b_3)}{\sum_{i=1}^{3} p(o|b_i)p(b_i)} = 9/17 \approx 0.53.$$

   Therefore,

   $$\begin{aligned} p(o_2|o_1) &= \sum_{i=1}^{3} p(o_2, b_i|o_1) = \sum_{i=1}^{3} p(o_2|b_i, o_1)p(b_i|o_1) = \sum_{i=1}^{3} p(o_2|b_i)p(b_i|o_1) \\ &= 0.3 \times 3/17 + 0.5 \times 5/17 + 0.3 \times 9/17 \approx 0.36. \end{aligned}$$

2. (6 points)

   (a) (2 point) The sample code for PCA is as follows:

```
function PCs = PCA(X,nPC)

% function PCA performs Principal Component Analysis.
```

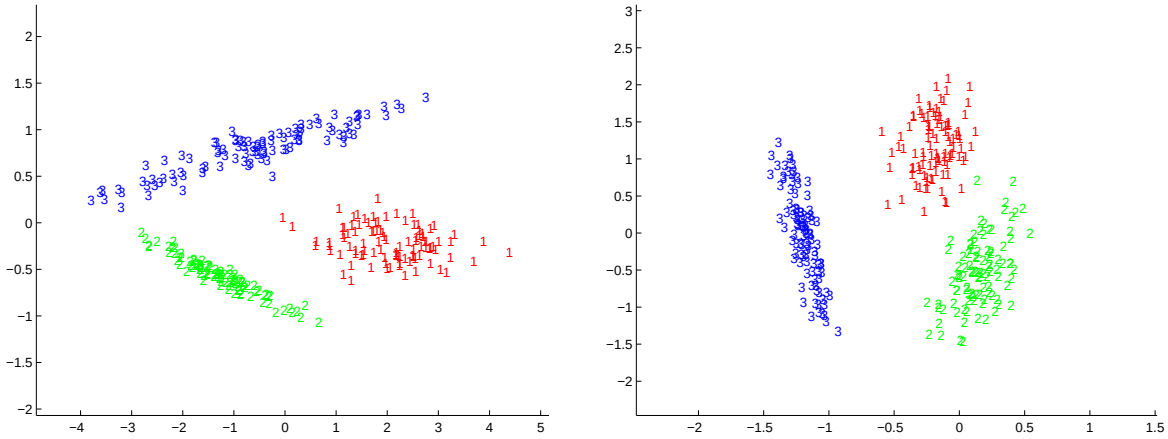
```
% Input:
%    X: (N by d matrix) where each row represents an instance
%  nPC: (scalar) the number of required principal components
%
% Output:
%  PCs: (N by nPC matrix) N instances represented by nPC principal components

[N,d] = size(X);
if d < nPC
    error(['The number of pricipal components is larger than the data ' ...
           'dimension']);
end

% compute covariance matrix
X = X - ones(N,1)*mean(X);
C = X'*X/N;

% compute eigenvalues
[V,D] = eig(C);
[val,idx] = sort(diag(D),'descend');
PCs = X*V(:,idx(1:nPC));

end % pca function
```

The sample code of LDA is as follows:

```
function DFs = LDA(X,labels)

% function LDA performs Fisher's Linear Discriminent Analysis.
% Input:
%       X: (N by d matrix) where each row represents an instance
%  labels: (N by 1 vector) the class labels of instances
%
% Output:
%  DFs: (N by nClass-1 matrix) N instances projected onto the
%       (nClass-1) discriminent, where nClass is the number of classe

c = unique(labels);

% compute the between-class covariance matrix
Sb = zeros(size(X,2));
for i = 1:length(c)
    Xc = X(labels==c(i),:);
    Sb = Sb+(mean(Xc)-mean(X))'*(mean(Xc)-mean(X))*size(Xc,1);
end

% compute the within-class covariance matrix
Sw = zeros(size(X,2));
for i = 1:length(c)
```

```
    Xc = X(labels==c(i),:);
    Sw = Sw+(Xc-ones(size(Xc,1),1)*mean(Xc))'*(Xc-ones(size(Xc,1),1)*mean(Xc));
end

% solve the generalized eigenvalue problem
[V,D] = eig(Sb,Sw);
[val,idx] = sort(diag(D),'descend');
DFs = X*V(:,idx(1:length(c)-1));

end % LDA function
```

(b) (2 points) For data set 1, the sample plots are shown in Figure 1.



By PCA By LDA

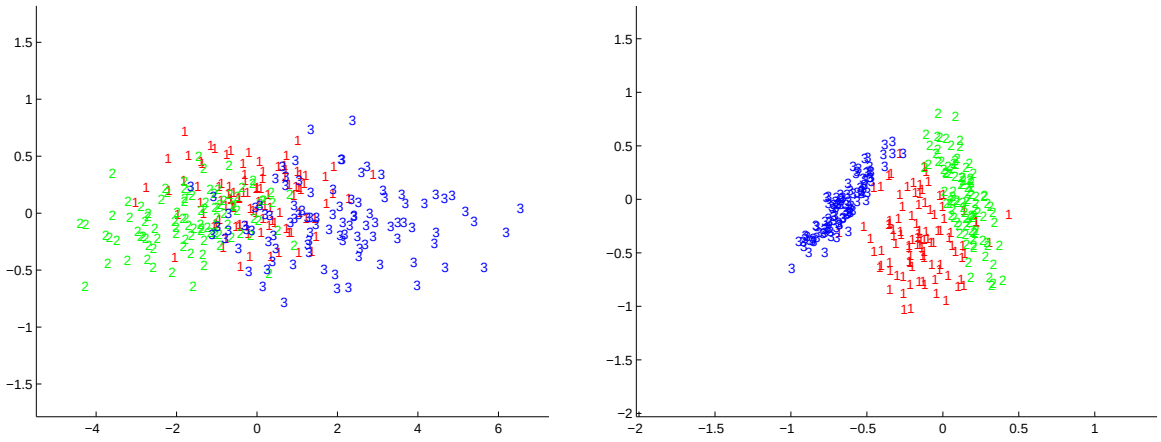
Figure 1: Projected Points in Data Set 1.

(c) (2 points) For data set 2, the sample plots are shown in Figure 2, from which we observe that LDA performs better than PCA in scattering instances according to their class labels in the reduced low-dimensional subspace.

3. (4 points)

(a) (2 points) **Proof.** Given a non-singular covariance matrix Σ, the normal to the separating hyperplane is $\boldsymbol{w} = \Sigma^{-1}(\boldsymbol{m}_2 - \boldsymbol{m}_1)$. We need to show that $\boldsymbol{w}^T(\boldsymbol{m}_2 - \boldsymbol{m}_1) = (\boldsymbol{m}_2 - \boldsymbol{m}_1)^T\Sigma^{-1}(\boldsymbol{m}_2 - \boldsymbol{m}_1) \neq 0$. Since the covariance matrix Σ is non-singular and hence positive definite, which means $\boldsymbol{v}^T\Sigma\boldsymbol{v} > 0$ for any vector $\boldsymbol{v} \neq \boldsymbol{0}$. Since $(\Sigma\boldsymbol{v})^T\Sigma^{-1}(\Sigma\boldsymbol{v}) = \boldsymbol{v}^T\Sigma\boldsymbol{v} > 0$ for any vector $\boldsymbol{v} \neq \boldsymbol{0}$, Σ^{-1} is also positive definite. Therefore, $(\boldsymbol{m}_2 - \boldsymbol{m}_1)^T\Sigma^{-1}(\boldsymbol{m}_2 - \boldsymbol{m}_1) > 0$ $(\boldsymbol{m}_2 - \boldsymbol{m}_1 \neq 0)$.

(b) (2 points) An example is shown in Figure 3. We can see that the shapes of the sample's distributions in two classes are both "skewed" along the direction of $\boldsymbol{m}_2 - \boldsymbol{m}_1$, which make the separating hyperplane almost parallel to $\boldsymbol{m}_2 - \boldsymbol{m}_1$. In fact, $\boldsymbol{w}^T(\boldsymbol{m}_2 - \boldsymbol{m}_1) \approx 0.0692$. The code and data used to generate this plot can be found at `http://www.cs.utexas.edu/~wtang/cs391d/hw2p3.tar.gz`. Note that in Figure 3 the x-axis and y-axis are not in the same scale.



By PCA By LDA

Figure 2: Projected Points in Data Set 2.

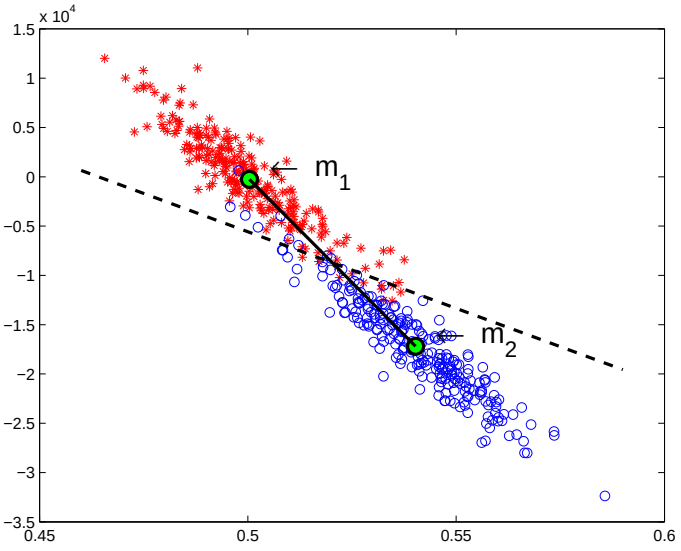


Figure 3: An example showing that $\boldsymbol{w}^T(\boldsymbol{m}_2 - \boldsymbol{m}_1) \approx 0$.