Homework Set One
ECE 271B - Winter 2019
Department of Electrical and Computer Engineering
University of California, San Diego

Probem 1. Consider the hyperplane

$$\mathbf{w}^T\mathbf{x} + b = 0$$

of known parameters $\mathbf{w}$ and b.

a) Use Lagrangian optimization to find the point $\mathbf{x}_0$ in the plane closest to the origin, as a function of $\mathbf{w}$ and b.

b) Use the result of **a)** to derive a geometric interpretation for the parameters $\mathbf{w}$ and b, when $||\mathbf{w}|| = 1$.

Problem 2. The entropy of probability density function (pdf) $p(x)$ is defined as

$$h(x) = -\int p(x)\log p(x)dx.$$

a) Use Lagrangian optimization to find the pdf of maximal entropy among those whose expected value is μ and variance is σ^2.

b) Find the values of the Lagrange multipliers associated with the optimal solution.

Problem 3. Extensive experimental evidence in the area of image compression has shown that the Discrete Cosine Transform (DCT) of image patches is a very good approximation to their PCA. It is also well known that all but one of the DCT coefficients (features) have zero mean, and only one has non-zero mean. The latter is the so-called DC coefficient because it results from projecting the image patch into the vector $\mathbf{1} = (1, 1, \ldots, 1)^T$ and, therefore, is proportional to the average (DC) value of the patch. In this problem, we are going to explore the connection between the DCT and PCA to explain this fact. For this, we are going to assume the following.

- An image patch is a collection of random variables $\mathbf{X} = \{X_1, \ldots, X_{64}\}$ which are identically distributed
$$P_{X_i}(x) = f(x),\ \forall i \in \{1, \ldots, 64\},$$
where $f(x)$ is a common probability density function.
- The pixels in the image patch are correlated, according to the *correlation coefficient*
$$\rho_{ij} = \frac{E[X_iX_j]}{\sqrt{E[X_i^2]}\sqrt{E[X_j^2]}}.$$
This obviously implies that we do not have an iid sample.

a) Consider the PCA of $\mathbf{X}$. Show that it is not affected by a change of variables of the type $\mathbf{Z} = \mathbf{X} - \boldsymbol{\mu}_x$, where $\boldsymbol{\mu}_x = E[\mathbf{X}]$.

b) Given **a)**, we can assume that $\mathbf{X}$ has zero mean. We will do so for the remainder of the problem. Show that in the extreme of highly correlated pixel values, i.e. when

$$\rho_{ij} \to 1, \quad \forall_{i,j},$$

the vector $\mathbf{1}$ is the largest principal component. Since neighboring image pixels do tend to be highly correlated, this helps explain why the DC coefficient is always present.

c) Let $\boldsymbol{\Phi}$ be the matrix whose columns $\boldsymbol{\phi}_i$ are the principal components and consider the set of coefficients (features) $\mathbf{Z}$ resulting from the projection of an arbitrary image patch $\mathbf{X}$ into these components, i.e.

$$\mathbf{z} = \boldsymbol{\Phi}^T \mathbf{x}.$$

Consider the DCT coefficients $z_i = \boldsymbol{\phi}_i^T \mathbf{x}$, noting that $z_1 = \mathbf{1}^T \mathbf{x}$ is the DC coefficient. Show that

$$E[z_i] = 0, \ \ \forall i > 1,$$

i.e. that the remaining (AC) coefficients have zero mean.

Problem 4. Consider a classification problem with two Gaussian classes

$$P_{\mathbf{X}|Y}(\mathbf{x}|i) = \mathcal{G}(\mathbf{x}, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \ i \in \{1, 2\}$$

and class probabilities

$$P_Y(1) = P_Y(2) = 1/2.$$

a) The random variable $\mathbf{X}$ is not Gaussian, but we can still compute its mean and covariance. Show that

$$\boldsymbol{\mu}_x = E[\mathbf{X}] = \frac{1}{2}[\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2]$$

and

$$\boldsymbol{\Sigma}_x = E[(\mathbf{X} - \boldsymbol{\mu}_x)(\mathbf{X} - \boldsymbol{\mu}_x)^T] = \frac{1}{2}\left[\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2\right] + \frac{1}{4}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T.$$

b) In the remainder of this problem, we consider the 2-dimensional case with

$$\boldsymbol{\mu}_1 = -\boldsymbol{\mu}_2 = \boldsymbol{\mu} = \begin{bmatrix} \alpha \\ 0 \end{bmatrix},$$

where $\alpha > 0$, and

$$\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Gamma} = \begin{bmatrix} 1 & 0 \\ 0 & \sigma^2 \end{bmatrix}.$$

Using MATLAB, sample 1,000 points from the two Gaussian classes and make a plot of the sampled points for each of the following conditions:

- condition A: $\alpha = 10, \sigma^2 = 2$;
- condition B: $\alpha = 2, \sigma^2 = 10$.

c) Use MATLAB to perform a principal component analysis for the random variable $\mathbf{X}$. Determine the direction of the best 1-dimensional subspace, i.e. the transformation

$$z = \boldsymbol{\phi}^T \mathbf{x},$$

where $\boldsymbol{\phi}$ is the principal component of largest variance. Hand in a plot containing both the datapoints and this principal component for each of the conditions of **b**).

d) Repeat **c)**, but now for the 1-dimensional subspace spanned by the linear discriminant $\boldsymbol{\phi}'$ produced by LDA.

e) From the results above, is the PCA approach to dimensionality reduction always a good one from the classification point of view? How does it compare to LDA?

f) We now pursue a theoretical explanation for the observations above. Using the results of **a)**, derive the principal component $\boldsymbol{\phi}$ and the linear discriminant $\boldsymbol{\phi}'$ as functions of the parameters α and σ^2. From a classification point of view, under what conditions is it 1) better to use PCA, 2) better to use LDA, or 3) identical to use any of the two?

Problem 5. In this problem, we consider a dataset with faces from 6 people. These were divided into two sets. A training set is provided in the file `trainset.zip` and a test set in the file `testset.zip`. The original colored images of 150×150 pixels were converted to 50×50 grayscale pixels, resulting in a vector space of $2,500$ dimensions.

a) Using the training data, compute the PCA of the data and make a 4×4 plot showing the 16 principal components of largest variance.

b) Consider the 15 different pairs of classes that can be derived from the 6 face classes. Perform an LDA between the classes in each pair. Make a 4×4 plot containing the 15 linear discriminants (plus one empty plot). (*Note:* You will likely realize that the within scatter matrix is singular. To address this, use an RDA of regularization parameter $\gamma = 1$ instead of the LDA.)

c) For each image $\mathbf{x}$ in the training dataset, compute the projection vector

$$\mathbf{z} \quad = \quad \boldsymbol{\Phi}\mathbf{x},$$

where the rows of $\boldsymbol{\Phi}$ are the 15 principal components of largest variance of the face training data. Note that this produces a datasets $\{\mathbf{z}_i\}$ of 15 dimensional vectors. Use the PCA projection vectors $\mathbf{z}_i$ of the training images to learn a mean $\boldsymbol{\mu}_i$ and variance $\boldsymbol{\Sigma}_i$ per class. Use these to implement the Gaussian classifier.

Compute the PCA projections of the test images and use the Gaussian classifier above to classify them. Compute the average probability of classification error for each face class and the average error accross all classes.

d) Repeat **c**) using LDA projections $\mathbf{z}_i$, instead of PCA projections. (*Note:* You will likely realize that the within scatter matrix is singular. To address this, use an RDA of regularization parameter $\gamma = 1$ instead of the LDA.)

e) A popular alternative to RDA, when LDA requires the inversion of singular scatter matrix, consists of

first applying a PCA to an intermediate dimension k and then apply LDA to the resulting k-dimensional vectors. This approach is denoted as PCA+LDA. Repeat **d**) using a PCA+LDA with $k = 30$.

Note: For a random variable $\mathbf{X}$ with multiple Gaussian classes of mean $\boldsymbol{\mu}_i$ and covariance $\boldsymbol{\Sigma}_i$, the Gaussian classifier has decision rule

$$i^* = \arg\min_i (\mathbf{x} - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i) + \log |\boldsymbol{\Sigma}_i|.$$

Given a training set $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ for class i, the class mean and variance are estimated with

$$\begin{aligned} \boldsymbol{\mu}_i &= \frac{1}{n} \sum_{k=1}^{n} \mathbf{x}_k \\ \boldsymbol{\Sigma}_i &= \frac{1}{n} \sum_{k=1}^{n} (\mathbf{x}_k - \boldsymbol{\mu}_i)(\mathbf{x}_k - \boldsymbol{\mu}_i)^T. \end{aligned}$$