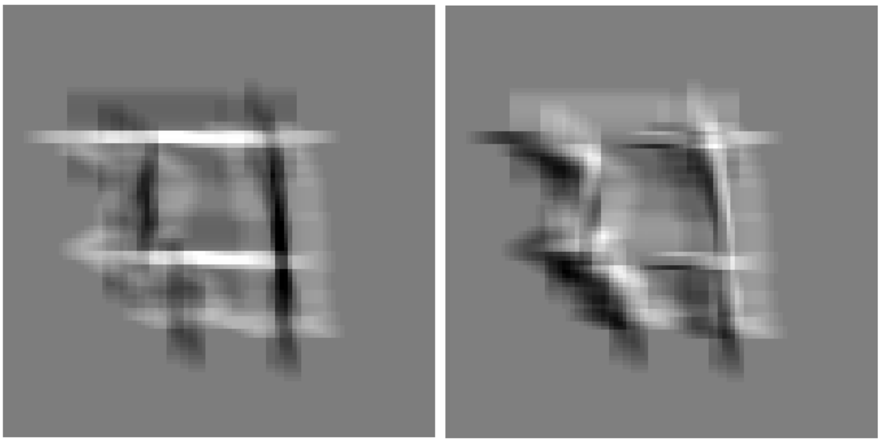
EEL709: Re-Minor II

April 13, 2013

Maximum Marks: 25

1. The images below depict the first two principal components obtained from running PCA on a set of 100×100 pixel images of a handwritten character from the Devanagari script. White pixels denote positive weights, black pixels denote negative weights, and gray pixels denote zero weights.



(a) $\lambda_1 = 5.43 \times 10^6$

(b) $\lambda_2 = 4.66 \times 10^6$

(a) The first component appears to show sharper horizontal white lines, whilst the second has sharper vertical or near-vertical white lines. What does this tell you about the respective kinds of variation they're capturing? **[1]**

(b) The corresponding eigenvalues are also given to you. The sum of all 10,000 eigenvalues from the data covariance matrix S was 3.04×10^7. What is the percentage variance not captured by the two depicted components? **[1]**

(c) Given the kinds of variation that the first two principal components appear to represent, do you think they're fully capturing that variability? Why or why not? What kind of method would you suggest to gauge the intrinsic dimensionality of the data? **[2]**

2. Consider the following generative model. I have K unbalanced 6-sided dice, where the probability of getting the number i with the k^{th} die is μ_{ki}. Also, I have an associated prior probability for each die, π_k, such that $\pi_k > 0$ and

$$\sum_{k=1}^{K} \pi_k = 1.$$

Now, a data point $x \in \{1, 2, 3, 4, 5, 6\}$ is generated by picking a die at random from the prior distribution, tossing it, and setting x to the value turned up.

(a) What is the probability distribution for x? What kind of model is this? **[2]**

(b) Obtain expressions for the mean and variance of x, in terms of the given parameters. **[2]**

(c) Now introduce an appropriate latent variable for this model (please make sure to clearly specify your use of notation). What is the joint distribution over the latent and observed variables? **[1]**

(d) Suppose you have observed a data set $\mathbf{X} = \{x_1, x_2 ..., x_N\}$. Write down the complete-data log likelihood, including your assumed latent variable. **[1]**

(e) Work out the E and M steps for the EM algorithm to estimate the values of the model parameters, $\{\mu_{ki}\}$ and $\{\pi_k\}$, that maximise the expected complete-data log likelihood. You should clearly show what updates are to be done in the two steps, and derive each of these. How would you interpret your results in words? **[5]**

3. We have seen in class that a soft-margin separating hyperplane of the form $\mathbf{w}^{\mathrm{T}}\mathbf{x} + b = 0$ can be obtained via optimising the following Lagrangian:

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\mu}, \boldsymbol{\lambda}) = \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum_{i=1}^{N}\xi_i + \sum_{i=1}^{N}\mu_i[1 - \xi_i - y_i(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b)] - \sum_{i=1}^{N}\lambda_i\xi_i,$$

where $\{(\mathbf{x}_1, y_1); (\mathbf{x}_2, y_2); ...; (\mathbf{x}_N, y_N)\}$ denote the observed data points, with $y_i \in \{-1, 1\}$; $\boldsymbol{\xi} = \{\xi_i, \xi_2, ..., \xi_N\}$ are the slack variables, and $\boldsymbol{\mu} = \{\mu_1, \mu_2, ..., \mu_N\}$ and $\boldsymbol{\lambda} = \{\lambda_1, \lambda_2, ..., \lambda_N\}$ are the Lagrange multipliers.

(a) By deriving and substituting in the values of the hyperplane parameters and the slack variables that minimise L, obtain the dual of this Lagrangian, as a function of just the multipliers $\boldsymbol{\mu}$. **[3]**

(b) Also obtain the constraints on $\boldsymbol{\mu}$ for this dual. How do these differ from the hard-margin case? **[1]**

(c) Let $f : \mathbb{R}^D \mapsto \mathbb{R}$ be a real-valued function, where D is the dimensionality of the $\mathbf{x}_i$. Is the function $K(\mathbf{x}_1, \mathbf{x}_2) = f(\mathbf{x}_1)^2 + f(\mathbf{x}_2)^2$ guaranteed to be a valid kernel? Prove either way. **[2]**

4. Suppose I am using a neural network for regression; so a given data point $\mathbf{x}$ is to be mapped to a target value $t \in \mathbb{R}$. I have a single output,

$$y = \sum_{j=1}^{M} w_j^{(2)} z_j,$$

where $z_1, z_2, ..., z_M$ denote the outputs of the hidden units, which are functions of $\mathbf{x}$:

$$z_j = h\left(\sum_{i=1}^{D} w_{ji}^{(1)} x_i\right).$$

Here $h()$ is a non-linear mapping, $w_{ji}^{(1)}$ is the weight of the link from the i^{th} input to the j^{th} hidden unit, and $w_j^{(2)}$ is the weight of the link from the j^{th} hidden unit to the output y. Let the interpretation of y be that $p(t|\mathbf{x}) \sim \mathcal{N}(y, \beta^{-1})$, where we are assuming Gaussian noise with precision β. Write down the network error function (negative log likelihood) for a single data point $(\mathbf{x}, t)$. Calculate δ, the gradient of the error function with respect to y. Interpret this in words. Show (by deriving the corresponding expressions) how this quantity is used in the computation of the error gradient with respect to both the layers of weights. **[4]**