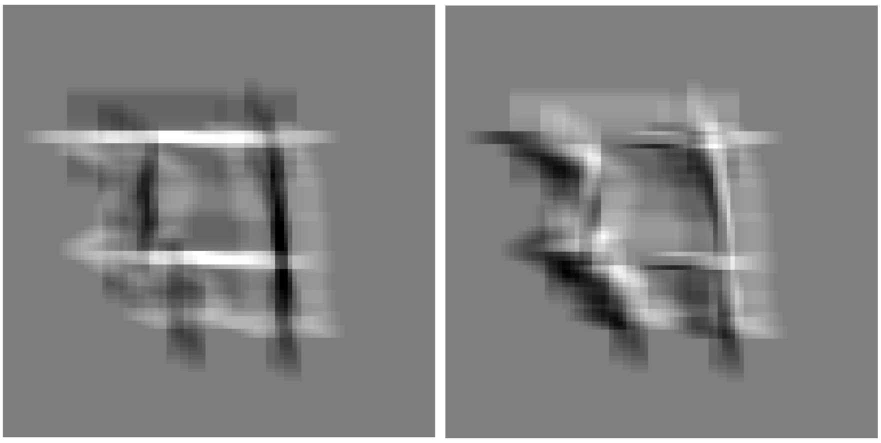
EEL709: Minor II

26th March, 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) Which character do you think the original images contained? Can you intepret roughly what sort of variation the two depicted principal components are 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 of variance in the data captured by each of the first two components? **[1]**

(c) Now suppose I inform you that the given data set contained only two independent sorts of variation. What would be your guess as to what these two sorts were? Are the first two principal components adequately capturing them? If not, why not? In this case, could you suggest an alternative method for finding the two intrinsic dimensions? **[2]**

2. Consider the following generative model. I have K biased coins, where the probability of getting heads with the k^{th} coin is μ_k. Also, I have an associated prior probability for each coin, π_k, such that $\pi_k > 0$ and

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

Now, a data point $x \in \{0, 1\}$ is generated by picking a coin at random from the prior distribution, tossing it, and setting $x = 1$ if it turns up heads and $x = 0$ otherwise.

(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, $\{\mu_k\}$ 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 maximum-margin separating hyperplane of the form $\mathbf{w}^{\mathrm{T}}\mathbf{x}+b=0$ can be obtained via optimising the following Lagrangian:

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

where $\{(\mathbf{x}_1, y_1); (\mathbf{x}_2, y_2); ...; (\mathbf{x}_N, y_N)\}$ denote the observed data points, with $y_i \in \{-1, 1\}$; and $\boldsymbol{\mu} = \{\mu_1, \mu_2, ..., \mu_N\}$ are the Lagrange multipliers.

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

(b) Explain why this dual formulation is amenable to the application of the 'kernel trick'; write down the modified form of the dual in this 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 $K(\mathbf{x}_1, \mathbf{x}_2) = (f(\mathbf{x}_1) + f(\mathbf{x}_2))^2$ guaranteed to be a valid kernel? Prove either way. **[2]**

4. Suppose I am using a neural network for binary classification; so a given data point $\mathbf{x}$ is to be assigned to a label $t \in \{0, 1\}$. I have a single output, $y = \sigma(a)$, where σ denotes the logistic sigmoid function, and

$$a = \sum_{j=1}^{M} w_j z_j,$$

where $z_1, z_2, ..., z_M$ denote the outputs of the hidden units (which are functions of $\mathbf{x}$), and w_j is the weight of the link from the j^{th} hidden unit to the output y. Let the interpretation of y be that $y = p(t = 1|\mathbf{x})$. 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 a. Interpret this in words. How is this quantity useful in training the network? **[4]**