PAPER 8 Image Processing - 2007 Sample Exam Question

Below is a 5-part question. The actual exam question will have 3 parts.

1. Consider a set of N images $\mathcal{S} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ where each image is represented as a vector of M real-valued features, e.g. $\mathbf{x}_n = (x_{n1}, \ldots, x_{nM})$ and $x_{nm} \in \Re$.

   Assume you use a Gaussian model for these images:

$$p(\mathbf{x}_n|\boldsymbol{\mu}) = \prod_{m=1}^{M} p(x_{nm}|\mu_m)$$

   where $p(x_{nm}|\mu_m)$ is Gaussian with mean μ_m and variance 1.

   (a) Write down the likelihood of the vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_M)$ for data set $\mathcal{S}$.

   (b) Derive the maximum likelihood estimate of μ_m.

   (c) Assume a Gaussian prior on μ_m with zero mean and unit variance denoted $p(\mu_m) = \mathcal{N}(0, 1)$. Derive the posterior distribution $p(\mu_m|\mathcal{S})$.

   (d) Describe some limitations of the above model for modelling features of images.

   (e) Given two data sets of images, $\mathcal{S}$ and $\mathcal{S}'$, for example representing images of two concepts (e.g. "sheep" and "clouds"), describe an automatic method (algorithm and equations if needed) for determining whether an image $\mathbf{x}$ fits better with $\mathcal{S}$ and $\mathcal{S}'$.

SOLUTIONS

1. Answers to different parts...

   (a)

$$
\begin{aligned}
P(\mathcal{S}|\boldsymbol{\mu}) &= \prod_{n=1}^{N}\prod_{m=1}^{M}(2\pi)^{-1/2}\exp\{-\frac{1}{2}(x_{nm}-\mu_m)^2\} \\
&= (2\pi)^{-\frac{NM}{2}}\exp\left\{-\frac{1}{2}\sum_{nm}(x_{nm}-\mu_m)^2\right\}
\end{aligned}
$$

   (b) Take log likelihood as a function of μ_m dropping all constants:

$$L(\mu_m) = -\frac{1}{2}\sum_n (x_{nm}-\mu_m)^2$$

   Maximize this as a function of μ_m, by taking derivatives and setting to zero:

$$\frac{\partial L(\mu_m)}{\partial \mu_m} = \sum_n (x_{nm}-\mu_m) = 0$$

   Solving for μ_m we get:

$$\mu_m = \frac{1}{N}\sum_n x_{nm}$$

   which is the sample mean of the mth image feature.

   (c)

$$p(\mu_m|\mathcal{S}) \propto p(\mathcal{S}|\mu_m)p(\mu_m)$$

   Again, dropping constants that don't depend on μ_m we get;

$$p(\mu_m|\mathcal{S}) \propto \exp\{-\frac{1}{2}\sum_n (x_{nm}-\mu_m)^2\}\exp\{-\frac{1}{2}\mu_m^2\}$$

   Clearly this is a Gaussian in μ_m. It suffices to compute the mean and variance of this Gaussian by matching terms to the expression for a standard Gaussian:

$$\exp\{-\frac{1}{2s^2}(\mu_m-u)^2\}$$

   The variance is $s^2 = \frac{1}{N+1}$ and the mean is $u = \frac{1}{N+1}\sum_n x_{nm}$. [Note that for no data points, this posterior is equal to the prior, which it obviously should be].

   (d) This model has numerous limitations: (a) the features are all independent, no correlations between features are modelled! (b) the noise variance is fixed at 1, rather than being learned; (c) feature distributions may be poorly modelled by the Gaussian distribution.

(e) There are several correct answers to this: (a) you could find the nearest neighbor to all elements of these two sets and judge $\mathbf{x}$ to fit with the set containing the nearest neighbor; (b) you could compute the mean of $\mathcal{S}$ and of $\mathcal{S}'$, and find which of these two means $\mathbf{x}$ is closer to; (c) you could learn a probabilistic model from $\mathcal{S}$, and from $\mathcal{S}'$ with parameters $\boldsymbol{\mu}$ and $\boldsymbol{\mu}'$ respectively, and see which gives $\mathbf{x}$ higher probability; i.e. select $\mathcal{S}$ if:

$$p(\mathbf{x}|\boldsymbol{\mu}) > p(\mathbf{x}|\boldsymbol{\mu}')?$$

(d) you could do the same as in (c) but integrating over parameters:

$$p(\mathbf{x}|\mathcal{S}) > p(\mathbf{x}|\mathcal{S}')?$$

(e) you could build a classifier to classify $\mathcal{S}$ from $\mathcal{S}'$ [if you've somehow learned about this].