Introduction to Machine Learning **Fall Semester**

Homework 5: Jan 22th, 2017

Due: Feb 12th, 2017 (See the submission guidelines in the course web site)

Theory Questions

1. Consider a collection of points, $x_1, \ldots x_m$, sampled i.i.d. according to an exponential distribution. (Recall that an exponential distribution has a parameter λ and the density of x is $\lambda e^{-\lambda x}$.)

   (a) What is the Maximum Likelihood (ML) estimate of λ?

   (b) What is the Maximum A Posteriori (MAP) value of λ given that its prior distribution is exponential with parameter 1?

   When maximizing, remember to check also the second derivative.

2. Given a sample $\mathbf{x}_1, \ldots, \mathbf{x}_n$, where $\mathbf{x}_i \in \mathcal{X}$, denote the empirical distribution by $\hat{p}$. That is, for every $\mathbf{x} \in \mathcal{X}$,

   $$\hat{p}(\mathbf{x}) = |\{i|\mathbf{x}_i = \mathbf{x}\}|/n$$

   Let $\mathcal{F}$ be a family of distributions. Show that the maximum likelihood distribution is minimizing the KL-divergence to the empirical distribution:

   $$\arg\min_{p\in\mathcal{F}} D_{KL}[\hat{p}; p] = \arg\max_{p\in\mathcal{F}} \Pr_p(\mathbf{x}_1, \ldots, \mathbf{x}_n)$$

3. (a) A sample $\mathbf{x}_1, \ldots, \mathbf{x}_n$ is generated as follows. For $i = 1, \ldots, n$, first, 5 bit binary string $\mathbf{z} \in \{\mathbf{s}_0 \ldots \mathbf{s}_{31}\}$ is generated with probability p_r, where $\mathbf{s}_r \in \mathbb{Z}_2^5$ is the binary representation of the integer $r = 0, \ldots, 31$ (for example $s_{18} =$ "10010"). Then, two random bits of $\mathbf{s}_r$ are *masked* (that is, replaced with "2") to create $\mathbf{x}_i$. For example, the result of masking $\mathbf{s}_{18}$ may be one of "22010", "20210", "20020", etc., each with probability $\binom{5}{2}^{-1} = 0.1$.

   What are the parameters of the model? What is the log-likelihood of the parameters given the data $\mathbf{x}_1, \ldots, \mathbf{x}_n$?

   (b) We wish to use EM to estimate the parameters of the model via MLE. Derive the EM algorithm's iterative update rules.

Programming Assignment

In this exercise, we will study the behaviour of the GMM on the MNIST dataset, with which you are by now *excessively* familiar. We divide the data into *training set* and *test set*. Under

`~schweiger/courses/ML2016-7/hw5.py`

(in the file system accessible from `nova`) you will find the code to load the training and test sets. It is recommended to use `numpy` and `scipy` where possible. We will not mean-center the data.

4. **GMM with EM.** We will see how well the Gaussian Mixture Model (GMM) can model the training data. Ideally, each cluster will capture one digit, where we focus only on a subset of the digits (0,1,3,4,8). Note that, as we are in an unsupervised scenario, we will not use the labels for learning the model.

   (a) The EM algorithm presented in class for GMM was for one-dimensional points ($d = 1$). Generalize the algorithm for multivariate normal distributions, where the covariance matrix of each of the multivariate Gaussians is of the form $\sigma_i^2 \cdot I$, where $0 < \sigma_i^2 \in \mathbb{R}$. What are the new EM update rules?

   (b) Implement the GMM EM algorithm.. Suggest a way to initialize the parameters, and suggest a criterion to stop the iterative process.

   Important note: Depending on your implementation, when calculating the posterior probabilities you may experience underflow (i.e., very small probabilities that will be rounded to 0). To avoid such problems, one solution is to calculate the probabilities $\Pr(Z_i = z_i, X_i = x_i)$ in log scale, and use the `logsumexp` function to substract the log of their sum, to get the (log of the) posterior probabilities. You are welcome to use other solutions if they work for you.

   (c) Run your implementation on the training data. Plot the likelihood of the data as a function of the current parameter estimates, and show that it is non-decreasing in each iteration.

   (d) Show the mean vectors of the 5 clusters you found, as images. Verify that each one corresponds to one digit. What are the cluster probabilities and variances you estimated?

   (e) Check the quality of your clustering on the test data. For each test image, calculate the probability that it was sampled from each one of the clusters. Classify it according to the cluster with the highest probability. What is the accuracy rate?