E0:270 Homework 4

Feb 28, 2012

Due: Mar 13, 2012

Problem 1

[A] Gaussian Mixture Model You are given a training dataset $\mathbf{X}$, which is in the form of an $N \times 2$ matrix (N data instances, each of dimension 2). You are also given a held-out test data set $\mathbf{X}_{test}$ of the same form. Both data sets are contained in `GMM_data.mat` and were generated from a GMM with K^* components (where K^* is unknown for you). Use Kevin Murphy's HMM toolbox to do the following. (Do not get alarmed that you are using HMM code to learn GMMs. The code is able to do both. The README file tells you how to do this.)

1. Given the number of components K in the GMM, use the function `mhmm_em.m` (in the above toolbox) to learn the parameters of the model $GMM(K)$ from dataset $\mathbf{X}$ using EM. Vary the number of components K from 1 to 10 to get 10 different models $\{GMM(K)\}_{K=1}^{10}$.

2. Write a MATLAB function `GMM_llhood.m` to compute the log-likelihood of a given dataset for a model $GMM(K)$ with known parameters. Plot the log-likelihood of the test dataset $\mathbf{X}_{test}$ for the 10 different models $\{GMM(K)\}_{K=1}^{10}$ trained as in (1) using the training data set $\mathbf{X}$.

3. Copy the EM code in `mhmm_em.m` to `mhmm_em_dump.m`. Change the code so that it outputs the log-likelihood and the model parameters after every 3 iterations. For the model $GMM(3)$ with $K = 3$, use `mhmm_em_dump.m` to plot the log-likelihood of the training data $\mathbf{X}$ and the contours of the Gaussian conditionals (using code from the last assignment) for every third iteration. (You may set the maximum number of iterations to 10.)

[B] Hidden Markov Model You are given a training dataset $\mathbf{X}$, in the form of an $N \times T \times 2$ matrix (N data sequences, each of length T and each instance of dimension 2). You are also given a held-out test data set $\mathbf{X}_{test}$ of the same form. Both data sets are contained in file `HMM_data.mat`, and were generated from an HMM with K^* components (where K^* is unknown for you), with each state having a Gaussian conditional.

1. Given the number of states K, use the function `mhmm_em.m` to learn the parameters of the model $HMM(K)$ from the training data $\mathbf{X}$ using the EM algorithm. Vary the number of states K from 1 to 10 to get 10 different models $\{HMM(K)\}_{K=1}^{10}$.

2. Use the function `fwdback.m` in the library to plot the log-likelihood of the test data $\mathbf{X}_{test}$ for the 10 different models $\{HMM(K)\}_{K=1}^{10}$ trained as in (1) using the training data set $\mathbf{X}$.

3. For the model $HMM(3)$ with $K = 3$, use the function `mhmm_em_dump.m` to plot the log-likelihood of the training data $\mathbf{X}$ in every third EM iteration. (You may set the maximum number of iterations to 10.)

4. Using the function `viterbi_path.m`, and the model $HMM(3)$, predict the most likely configuration of states $z_1 \ldots z_T$ for *only the first test sequence* in $\mathbf{X}_{test}$. In the first plot, plot the two dimensional

points $\{x_t\}_{t=1}^T$ using a particular color for a specific value of z_t. Additionally, plot the contours of the 3 Gaussian conditionals used for prediction. In the second plot, plot the sequence $\{(t, z_t)\}_{t=1}^T$, where z_t takes values from $\{1, 2, 3\}$, to observe the state transitions.

Problem 2

(a) Consider a set of M binary variables $x_i \in \{0, 1\}$ where $i = 1, \ldots, M$, each of which is governed by a Bernoulli distribution with parameter μ_i so that $p(x; \mu) = \prod_{i=1}^M \mu_i^{x_i}(1 - \mu_i)^{(1-x_i)}$, where $x = \{x_1 \ldots x_M\}^T$, $\mu = \{\mu_1 \ldots \mu_M\}^T$. Consider a finite mixture of K such M-dimensional Bernoulli distributions: $p(x; \mu, \pi) = \sum_{k=1}^K \pi_k p(x; \mu_k)$, where $\mu = \{\mu_1 \ldots \mu_K\}$ with $\mu_i = \{\mu_{i1} \ldots \mu_{iM}\}^T$ and $\pi = \{\pi_1 \ldots \pi_K\}$, with $\pi_i \geq 0$ and $\sum_{i=1}^K \pi_i = 1$. Derive the steps of the EM algorithm for finding the maximum likelihood estimates for this mixture distribution.

Suggest a clustering application where this model will be useful.

(b) Assume that the parameters $\{\mu_{ki}\}$ for the k^{th} component are random variables with the prior distributions $p(\mu_{ki}; a_{k1}, a_{k2})$ given by the same Beta distribution $Beta(a_{k1}, a_{k2})$, and similarly π is a random variable with prior distribution $p(\pi; b_1, \ldots, b_k)$ given by a Dirichlet $Dir(b_1 \ldots b_2)$. Derive the steps of the EM algorithm for maximizing the *posterior distribution* over parameters, instead of the likelihood.

(c) (For extra credit) Now imagine the M variables to be categorical $x_i \in \{1, \ldots, L\}$ where $i = 1, \ldots, M$, each of which is governed by a Multinomial distribution with parameter $\mu_i = \{\mu_{ij}\}_{j=1}^L$ so that $p(x; \mu) \propto \prod_{i=1}^M \prod_{j=1}^L \mu_{ij}^{x_{ij}}$, where $x = \{x_1 \ldots x_M\}^T$, $\mu = \{\mu_1 \ldots \mu_M\}^T$. Consider a finite mixture of K such M-dimensional Multinomial distributions: $p(x; \mu, \pi) = \sum_{k=1}^K \pi_k p(x; \mu_k)$, where $\mu = \{\mu_1 \ldots \mu_K\}$ with $\mu_k = \{\mu_{kij}\}$, $i = 1 \ldots M$, $j = 1 \ldots L$; and $\pi = \{\pi_1 \ldots \pi_K\}$ as before. Adapt the steps of the earlier EM algorithm from part (a) for finding the maximum likelihood estimates for this multinomial mixture distribution.

Suggest a clustering application where this model will be useful.

(d) (For extra credit) Assume that the parameters $\{\mu_{ki}\}$ for the k^{th} component are random variables with prior distribution $p(\mu_{ki}; a_{k1} \ldots a_{kL})$ given by the same Dirichlet $Dir(a_{k1}, \ldots, a_{kL})$, and as before $p(\pi; b_1, \ldots, b_k)$ is given by a Dirichlet. Use the results from part (b) and part (c) to write the E and M steps for maximizing the posterior distribution over parameters.

Problem 3

(a) Consider an alternative inference problem for the Hidden Markov Model, where the problem is to find the maximum probability over all configuration of hidden state variables: $max_{z_1 \ldots z_T}\ p(x_1 \ldots x_T, z_1 \ldots z_T)$. How would you change the forward-backward algorithm to solve this problem? Explain why your proposed algorithm is correct.

(b) Change the above algorithm to find the configuration of hidden state variables corresponding to this maximum probability: $argmax_{z_1 \ldots z_T}\ p(x_1 \ldots x_T, z_1 \ldots z_T)$. You may assume that there exists a unique maximizing configuration.

(c) (Not for submission) Does your algorithm work when there are multiple configurations with the same maximum probability. If it does not, how would you need to change your algorithm to deal with this?