10-701/15-781, Machine Learning: Homework 4

Aarti Singh
Carnegie Mellon University

- The assignment is due at 10:30 am (beginning of class) on **Mon, Nov 15, 2010**.
- Separate you answers into five parts, one for each TA, and put them into 5 piles at the table in front of the class. Don't forget to put both your name and a TA's name on each part.
- If you have question about any part, please direct your question to the respective TA who designed the part (however send your email to 10701-instructors@cs list).

1 Gaussian Mixture Models [TK, 20 points]

A Gaussian mixture model is a family of distributions whose pdf is in the following form:

$$gmm(\mathbf{x}) \ := \ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k),$$

where $\mathcal{N}(\cdot|\boldsymbol{\mu}, \Sigma)$ denotes the Gaussian pdf with mean $\boldsymbol{\mu}$ and covariance Σ, and $\{\pi_1, \dots, \pi_K\}$ are mixture weights satisfying

$$\sum_{k=1}^{K} \pi_k = 1, \quad \pi_k \geq 0, \quad k \in \{1, \dots, K\}.$$

The Expectation Maximization algorithm for learning a GMM from a set of sample points $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ is outlined below:

- Initialize $\boldsymbol{\mu}_k$, Σ_k, and π_k, $k \in \{1, \dots, K\}$.
- Repeat the following until convergence:
  1. E-step:
  $$\tilde{z}_{ik} \ \leftarrow \ \text{Prob}\Big(\mathbf{x}_i \in \text{ cluster } k \ \Big| \ \{(\pi_j, \boldsymbol{\mu}_j, \Sigma_j)\}_{j=1}^{K}, \mathbf{x}_i\Big),$$
  2. M-step:
  $$\{(\pi_k, \boldsymbol{\mu}_k, \Sigma_k)\}_{k=1}^{K} \ \leftarrow \ \arg\max \sum_{i=1}^{n}\sum_{k=1}^{K} \tilde{z}_{ik}\Big(\log \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_k, \Sigma_k) + \log \pi_k\Big)$$

1. **[6 points]** Consider a simplified GMM where all mixture components share the same covariance matrix, i.e., $\Sigma_k = \Sigma$. Derive the update rule for Σ in the M-step. (Your answer can rely on the value of $\boldsymbol{\mu}_k$ at the current M-step.)
2. **[6 points]** Consider an even more simplified GMM where all mixture components share a known covariance matrix $\sigma^2 I$, $\sigma^2 > 0$ and I being the identity matrix. Given a set of sample points $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, we apply the EM algorithm to estimate the means and the mixture weights, and get cluster probabilities for each sample point. Assume the following are true throughout the EM algorithm:
   - The mixture weights $\{\pi_1, \dots, \pi_K\}$ are bounded away from zero, i.e, $\exists\ \epsilon > 0$ such that $\pi_k \geq \epsilon\ \forall k \in \{1, \dots, K\}$ throughout the iterations.
   - Throughout the iterations,
   $$\|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2 \ \neq \ \|\mathbf{x}_i - \boldsymbol{\mu}_{k'}\|^2 \quad \forall\ i \in \{1, \dots, n\},\ k \neq k'.$$

Show that as $\sigma^2 \to 0$, the E-step converges to the update rule for γ in the Lloyd's algorithm (see Problem 5 of Homework 3), i.e., the soft assignment becomes hard.

3. **[8 points]** In this problem you will investigate connections between the EM algorithm and gradient ascent. Consider a GMM where $\Sigma_k = \sigma_k^2 I$, i.e., the covariances are spherical but of different spread. Moreover, suppose the mixture weights π_k's are known. The log likelihood then is

$$l\left(\{\boldsymbol{\mu}_k, \sigma_k^2\}_{k=1}^K\right) \;=\; \sum_{i=1}^n \log\left(\sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_k, \sigma_k^2 I)\right).$$

A maximization algorithm based on gradient ascent is as follows:

- Initialize $\boldsymbol{\mu}_k$ and σ_k^2, $k \in \{1, \ldots, K\}$. Set the iteration counter $t = 1$.
- Repeat the following until convergence:
  - For $k = 1, \ldots, K$,
  $$\boldsymbol{\mu}_k^{(t+1)} \;\leftarrow\; \boldsymbol{\mu}_k^{(t)} + \eta_k^{(t)} \nabla_{\boldsymbol{\mu}_k} l\left(\{\boldsymbol{\mu}_k^{(t)}, (\sigma_k^2)^{(t)}\}_{k=1}^K\right)$$
  - For $k = 1, \ldots, K$,
  $$(\sigma_k^2)^{(t+1)} \;\leftarrow\; (\sigma_k^2)^{(t)} + s_k^{(t)} \nabla_{\sigma_k^2} l\left(\{\boldsymbol{\mu}_k^{(t+1)}, (\sigma_k^2)^{(t)}\}_{k=1}^K\right)$$
  - Increase the iteration counter $t \leftarrow t + 1$.

Show that with properly chosen step sizes $\eta_k^{(t)}$ and $s_k^{(t)}$, the above gradient ascent algorithm is equivalent to the following modified EM algorithm:

- Initialize $\boldsymbol{\mu}_k$ and σ_k, $k \in \{1, \ldots, K\}$. Set the iteration counter $t = 1$.
- Repeat the following until convergence:
  (a) E-step:
  $$\tilde{z}_{ik}^{(t+0.5)} \;\leftarrow\; \mathrm{Prob}\Big(\mathbf{x}_i \in \text{ cluster } k \;\Big|\; \{(\boldsymbol{\mu}_j^{(t)}, (\sigma_j^2)^{(t)})\}_{j=1}^K, \mathbf{x}_i\Big),$$
  (b) M-step:
  $$\{\boldsymbol{\mu}_k^{(t+1)}\}_{k=1}^K \;\leftarrow\; \arg\max_{\{\boldsymbol{\mu}_k\}_{k=1}^K} \sum_{i=1}^n \sum_{k=1}^K \tilde{z}_{ik}^{(t+0.5)} \Big(\log \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_k, (\sigma_k^2)^{(t)} I) + \log \pi_k\Big)$$
  (c) E-step:
  $$\tilde{z}_{ik}^{(t+1)} \;\leftarrow\; \mathrm{Prob}\Big(\mathbf{x}_i \in \text{ cluster } k \;\Big|\; \{(\boldsymbol{\mu}_j^{(t+1)}, (\sigma_j^2)^{(t)})\}_{j=1}^K, \mathbf{x}_i\Big),$$
  (d) M-step:
  $$\{(\sigma_k^2)^{(t+1)}\}_{k=1}^K \;\leftarrow\; \arg\max_{\{\sigma_k^2\}_{k=1}^K} \sum_{i=1}^n \sum_{k=1}^K \tilde{z}_{ik}^{(t+1)} \Big(\log \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_k^{(t+1)}, \sigma_k^2 I) + \log \pi_k\Big)$$
  (e) Increase the iteration counter $t \leftarrow t + 1$.

The main modification is inserting an extra E-step between the M-step for $\boldsymbol{\mu}_k$'s and the M-step for σ_k^2's. (Hint: the choices of the step sizes should be related to the E-steps.)

2 Expectation Maximization [Jayant, 25 points]

In this problem, you will implement the EM algorithm to learn the parameters of a two-class Gaussian mixture model. Recall that a mixture model is a density created by drawing each instance X from one of two possible distributions, $P(X|Y=0)$ or $P(X|Y=1)$. Y is a hidden variable over classes that simply indicates the distribution each instance is drawn from. We will assume that $P(Y)$ is a Bernoulli distribution and each $P(X|Y)$ is a 1-dimensional Gaussian with unit variance. The joint density is therefore:

$$P(X=x) = \sum_{y \in \{0,1\}} P(X=x|Y=y) \times P(Y=y)$$
$$P(X=x;\mu,\theta) = \sum_{y \in \{0,1\}} \frac{1}{\sqrt{2\pi}} \exp\{-\frac{(x-\mu_y)^2}{2}\} \times \theta_y$$

The parameters of this model are $\mu = [\mu_0, \mu_1]$ and $\theta = [\theta_0, \theta_1]$, where μ_y is the mean of the Gaussian for class y, and $\theta_y = P(Y=y)$ is the probability that an instance is drawn from class y. (Note that $\theta_0 + \theta_1 = 1$.) We will use EM to estimate these parameters from a data set $\{x^i\}_{i=1}^n$, where $x^i \in \mathbf{R}$.

1. **[3 points]** Let p_{iy} denote the probability that the ith instance is drawn from class y (i.e., $p_{iy} = P(Y=y|X=x^i)$). During the iteration t, the E-step computes p_{iy} for all i, y using the parameters from the previous iteration, $\mu^{(t-1)}$ and $\theta^{(t-1)}$. Write down an expression for p_{iy} in terms of these parameters.

2. **[3 points]** The M-step treats the p_{iy} variables as fractional counts for the unobserved y values and updates μ, θ as if the point (x^i, y) were observed p_{iy} times. Write down an update equation for $\mu^{(t)}$ and $\theta^{(t)}$ in terms of p_{iy}.

3. **[9 points]** Implement EM using the equations you derived in parts 1 and 2. Print out your code and submit it with your solution.

4. **[5 points]** Download the data set from `http://www.cs.cmu.edu/~aarti/Class/10701/hws/hw4.data`. Each row of this file is a training instance x^i. Run your EM implementation on this data, using $\mu = [1, 2]$ and $\theta = [.33, .67]$ as your initial parameters. What are the final values of μ and θ? Plot a histogram of the data and your estimated mixture density $P(X)$. Is the mixture density an accurate model for the data?

   To plot the density in Matlab, you can use:

```
density = @(x) (<class 1 prior> * normpdf(x, <class 1 mean>, 1)) + ...
               (<class 2 prior> * normpdf(x, <class 2 mean>, 1));
fplot(density, [-5, 6]);
```

Recall from class that EM attempts to maximize the marginal data loglikelihood $\ell(\mu, \theta) = \sum_{i=1}^n \log P(X = x^i; \mu, \theta)$, but that EM can get stuck in local optima. In this part, we will explore the shape of the loglikelihood function and determine if local optima are a problem. For the remainder of the problem, we will assume that both classes are equally likely, i.e., $\theta_y = \frac{1}{2}$ for $y = 0, 1$. In this case, the data loglikelihood ℓ only depends on the mean parameters μ.

1. **[5 points]** Create a contour plot of the loglikelihood ℓ as a function of the two mean parameters, μ. Vary the range of each μ_k from -1 to 4, evaluating the loglikelihood at intervals of .25. You can create a contour plot in Matlab using the `contourf` function. Print out your plot and include in with your solution.

   Does the loglikelihood have multiple local optima? Is it possible for EM to find a non-globally optimal solution? Why or why not?

3 Learning Theory [Leman, 15 points]

3.1 VC Dimension

In this section you will calculate the lower-bound for the VC-dimension of some hypothesis classes.

1. **[5 points]** Consider the hypothesis class of linear classifiers with offset in d dimensions:

$$\mathcal{H} = \{\text{sign}(\theta \cdot x + \theta_0) : \theta \in R^d, \theta_0 \in R\}$$

   Show that there exists a set of $d+1$ points $\{x_1, x_2, \ldots, x_{d+1}\}$ that can be shattered by $\mathcal{H}$. Specifically, first specify the points, and then given any labeling $y_1, y_2, \ldots, y_{d+1}$, describe explicitly how to construct a classifier in $\mathcal{H}$ that agrees with the labeling.

2. **[5 points]** Consider the hypothesis class of convex d-gons in the plane. A point is labeled positive if it is inside the d-gon. Demonstrate that there exists a set of $2d+1$ points on which any labeling can be shattered. *Hint:* You may think of data points on a circle.

3.2 Sample Complexity

In this part, you will use sample complexity bounds to determine how many training examples are needed to find a good classifier.

Let $\mathcal{H}$ be the hypothesis class of convex d-gons in the plane. In part 1, you showed that the VC dimension of d-gons in R^2 is at least $2d+1$. It can be shown that the upper bound is also $2d+1$.

Suppose we sample a number of m training examples i.i.d. according to some unknown distribution $\mathcal{D}$ over $R^2 \times \{+, -\}$.

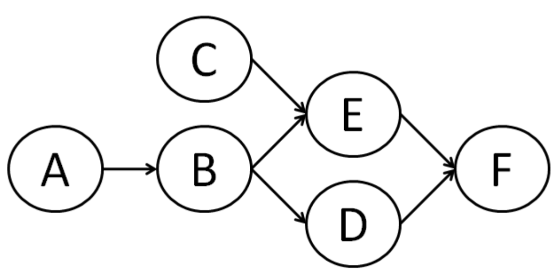
3. **[5 points]** What is the least number of training examples $m > 1$ you need to have such that with probability at least 0.95 the convex 4-**gon** separator in the plane with the smallest training error $\hat{h}_{ERM} = \arg\min_{h \in \mathcal{H}} \text{error}_{train}(h)$ has the following? Please show all your work.

$$\text{error}_{true}(\hat{h}_{ERM}) - \text{error}_{train}(\hat{h}_{ERM}) \leq 0.05$$

   Note that you may <u>not</u> assume $\text{error}_{train}(\hat{h}_{ERM})$. You may use any formulas from the lecture slides, textbook, or readings from the website, but please tell us where you found the formula(s) you use.

4 Bayesian Network [Rob, 20 points]

This problem will concern the below Bayesian network.



4.1 [3 points] Joint Probability

Write down the factorization of the joint probability distribution over A, B, C, D, E, F which corresponds to this graph.

4.2 [12 points] Inference

For this section we suppose that all the variables are binary, taking on the values $0, 1$. The conditional probability distributions on the graph have the following form:

- Nodes with a single parent take the value of their parent with probability $\frac{3}{4}$ otherwise they take the other value.
- Nodes with two parents take the value of the first parent with probability $\frac{1}{2}$ otherwise they take the value of the second parent.
- $P(A = 1) = p$, $P(C = 1) = q$.

1. [**2 points**] **CPT Tricks I**. If some node X has a single parent Y, and $P(Y = 1) = a$, what is a simple expression for $P(X = 1)$? Please assume that there is no child of X to worry about.
2. [**3 points**] **CPT Tricks II**. If some node X has a two independent parents Y, Z, and $P(Y = 1) = a$, $P(Z = 1) = b$, what is a simple expression for $P(X = 1)$? Please assume that there is no child of X to worry about.
3. [**7 points**] **Forwards Inference**. What is $P(F = 1)$ in the above graph?

4.3 [5 points] Conditional Inference

If $B = b, F = f$ are observed, what is the conditional probability that $E = 1$? For this question please leave your answer in terms of probability distributions e.g., $P(B = b|A = a)$ etc., but only those which could be computed directly from the local probabilities in the definition of the Bayes net.

5 HMM [Min Chi, 20 points]

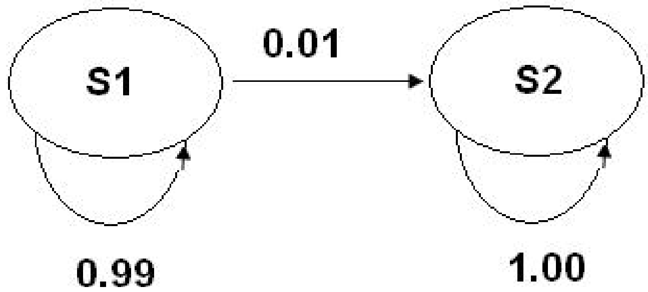
1. [**16 points**] Figure **??** shows a two-state HMM. The transition probabilities of the Markov chain are given in the transition diagram. The output distribution corresponding to each state is defined over $\{1, 2, 3, 4\}$ and is given in the table next to the diagram. The HMM is equally likely to start from either of the two states.

   (a) [**3 points**] Give an example of an *output sequence* of length 2 which can not be generated by the HMM in Figure **??**. Justify your answer.

   (b) [**3 points**] We generated a sequence of $10, 701^{2010}$ observations from the HMM, and found that the last observation in the sequence was 3. What is the most likely hidden state corresponding to that last observation?

   (c) [**3 points**] Consider an output sequence $\{3, 3\}$. What is the most likely sequence of hidden states corresponding to this output observation sequence? Show your work.

   (d) [**3 points**] Now, consider an output sequence $\{3, 3, 4\}$. What are the first two states of the most likely hidden state sequence? Show your work.



	S1	S2
P(x=1)	0	0.1
P(x=2)	0.199	0
P(x=3)	0.8	0.7
P(x=4)	0.001	0.2

Figure 1: A two-state HMM

(e) [**3 points**] We can try to increase the modeling capacity of the HMM a bit by breaking each state into two states. Following this idea, we created the diagram in Figure **??**. Can we set the initial state distribution and the output distributions so that this 4-state model, with the transition probabilities indicated in the diagram, would be equivalent to the original 2-state model (i.e. for for any output sequence O, $P(O|HMM_{2states}) = P(O|HMM_{4states})$)?. If yes, how? If no, why not?

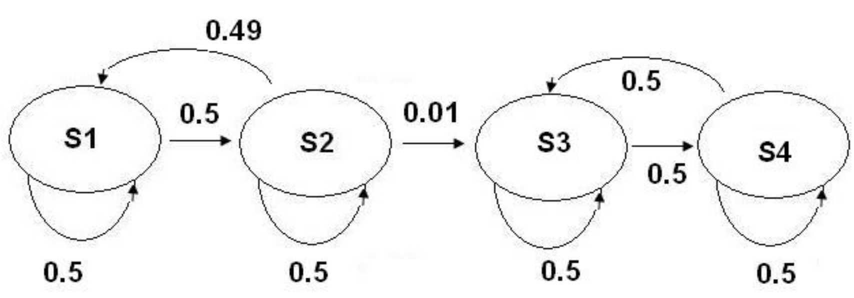


Figure 2: An alternative, four-state HMM

2. **[5 points]** The HMM (Hidden Markov Model) is a probabilistic model of the joint probability of a collection of random variables with both observations and states. The EM (Expectation-Maximum) algorithm is a general method to improve the gradient descent algorithm for finding the Maximum Likelihood Estimates. So we can derive the EM algorithm for finding the maximum-likelihood estimate of the parameters of a HMM given a set of observed feature vectors.

   Suppose that the initial "guess" in the transition probability matrix provided to the EM is set to zero for an entry state s_i and next state s_j. In that words, we have the initial parameter for $p(s_{t+1} = j | s_t = i) = 0$. Prove that $p(s_{t+1} = j | s_t = i)$ will remain zero in the result obtained with the EM.