Homework Assignment 1

Due: January 30, 2018 at 12:01 am

Total points: 100

Deliverables: hw1.pdf containing typeset solutions to Problems 1-2.
Source code containing your implementation for Problem 3.
README file explaining how to compile and run the source code on Linux or Windows.

Guidelines: All files must be submitted by Dropbox. You can brainstorm with others, but please solve the problems and write up the answers and code by yourself. You may use textbooks (Koller & Friedman, Russell & Norvig, etc.), lecture notes, and standard programming references (e.g., online Java API documentation). Please do NOT use any other resources or references (e.g., example code, online problem solutions, etc.) without asking.

1. Probability Theory

1.1. *(5 points)* A fire alarm rings when it detects fire. We can connect multiple alarms one after another to improve the overall system reliability, and we evacuate people if any of the alarms rings. Here we know the reliability (chance to ring when fire exists) is 0.96 for a two-alarm system. How many alarms shall we connect if we are going to build a system that has at least 0.9999 reliability? Assume the alarms are identical and independent.

1.2. *(10 points)* Cinemas A and B are competing among 1,000 customers. Assume each customer independently chooses between them with equal probability. How many seats should cinema A have, so that the probability is less than 1% for a customer being unable to buy a ticket because the seats are sold out.

1.3. *(10 points)* When sending one of three letters (A, B, C) through a telephone line, the probability of receiving a correct letter is a, and $\frac{1-a}{2}$ for receiving each of the wrong letters. Now we know the probabilities are p_1, p_2, p_3 $(p_1 + p_2 + p_3 = 1)$, respectively, for sending $AAAA, BBBB, CCCC$. What is the probability of sending $AAAA$ if we have received $ABCA$? Assume sending and receiving a letter at different steps are independent events.

2. Bayesian Analysis of the Uniform Distribution

2.1. *(10 points)* Consider the uniform distribution $Unif(0, \theta)$. The maximum likelihood estimate is $\hat{\theta} = \max D$, but this is unsuitable for predicting future data since it puts zero probability mass outside the training data. In this exercise, we will perform a Bayesian analysis of the uniform distribution. The conjugate prior is the Pareto distribution, $p(\theta) = Pareto(\theta | b, K)$, the pdf of which is defined in the following form:

$$Pareto(\theta|b,K) = Kb^{K}\theta^{-(K+1)}\mathbf{1}_{(\theta\geq b)}$$

Given a Pareto prior, the joint distribution of θ and $D = (x_1, \dots, x_N)$ is

$$p(D,\theta) = \frac{Kb^K}{\theta^{N+K+1}}\mathbf{1}_{\theta\geq\max(b,\max(D))}$$

Let $m = \max(D)$. The evidence (the probability that all N samples came from the same uniform distribution is

$$p(D) = \int_{\max(b,m)}^{\infty} \frac{Kb^K}{\theta^{N+K+1}}d\theta = \begin{cases} \frac{K}{(N+K)b^N}, \text{if } m \leq b \\ \frac{Kb^K}{(N+K)m^{N+K}}, \text{if } m > b \end{cases}$$

Derive the posterior $p(\theta|D)$, and show that it can be expressed as a Pareto distribution.

2.2. *(15 points)* Suppose you arrive in a new city and see a taxi number 100. How many taxis are there in this city? Let us assume taxis are numbered sequentially as integers starting from 0, up to some unknown upper bound θ. (We number taxis from 0 for simplicity; we can also count from 1 without changing the analysis.) Hence the likelihood function is $p(x) = U(0,\theta)$, the uniform distribution. The goal is to estimate θ. We will use the Bayesian analysis from the previous question.

(a) Suppose we see one taxi numbered 100, so $D = \{100\}, m = 100, N = 1$. Using an (improper) non-informative prior on θ of the form $p(\theta) = Pa(\theta|0{,}0) \propto 1/\theta$, what is the posterior $p(\theta|D)$?

(b) Rather than trying to compute a point estimate of the number of taxis, we can compute the predictive density over the next taxicab number using

$$p(D'|D,\alpha) = \int p(D'|\theta)p(\theta|D,\alpha)d\theta = p(D'|\beta),$$

where $\alpha = (b,K)$ are the hyperparameters, $\beta = (c, N+K)$ are the updated hyperparameters. Now consider the case $D = \{m\}$, and $D' = \{x\}$. Write down an expression for $p(x|D,\alpha)$. As above, use a non-informative prior $b = K = 0$.

3. The EM Algorithm

3.1. *(50 points)* Implement the EM algorithm for mixtures of Gaussians in your choice of programming language. (C, C++, Java, Perl, Python, and OCaml are all fine. Please ask about any others.) Assume that means, covariances, and cluster priors are all unknown. For simplicity, you can assume that covariance matrices are diagonal (i.e., all you need to estimate is the variance of each variable). Initialize the cluster priors to a uniform distribution and the standard deviations to a fixed fraction of the range of each variable. Your algorithm should run until the relative change in the log likelihood of the training data falls below some threshold (e.g.,stop when log likelihood improves by<0.1%). The program should be run on the command line with the following arguments:

```
./gaussmix <# of clusters> <data file> <model file>
```

It should read in data files in the following format:

<# of examples> <# of features>

<ex.1, feature 1> <ex.1, feature 2> … <ex.1, feature n> <ex.1, label>

<ex.2, feature 1> <ex.2, feature 2> … <ex.2, feature n> <ex.2, label>

…

And output a model file in the following format:

<# of clusters> <# of features>

<clust1.prior> <clust1.mean1> <clust1.mean2> … <clust1.var1> …

<clust2.prior> <clust2.mean1> <clust2.mean2> … <clust2.var1> …

…

Train and evaluate your model on the vehicle silhouette dataset, available from the course Web page. Each data point represents a vehicle, with features representing geometrical properties of the silhouette. We provide a single default train/test split to test generalization, and you can use the test data labels only for evaluation. The full dataset and more information can be found in the UCI repository (and linked from the course Web page). Start by using 4 clusters, since the dataset includes four different vehicles. Evaluate your models on the test data.

Two recommendations:

- To avoid underflows, work with logs of probabilities, not probabilities.
- To compute the log of a sum of exponentials, use the "log-sum-exp" trick:

$$log \sum_i \exp(x_i) = x_{max} + log \sum_i \exp(x_i - x_{max})$$

Answer the following questions with both numerical results and discussion.

(a) Plot train and test set likelihood vs. iteration. How many iterations does EM take to converge?

(b) Experiment with two different methods for initializing the mean of each Gaussian in each cluster: random values (e.g., uniformly distributed from some reasonable range) and random examples (i.e., for each cluster, pick a random training example and use its feature values as the means for that cluster). Does one method work better than the other or do the two work approximately the same? Why do you think this is? (Use whichever version works best for the remaining questions.)

(c) Run the algorithm 10 times with different random seeds. How much does the log likelihood change from run to run?

(d) Infer the most likely cluster for each point in the training data. How does the true clustering label compare to your inference?

(e) Graph the training and test set log likelihoods, varying the number of clusters from 1 to 10. Discuss how the training set log likelihood varies and why. Discuss how the test set log likelihood varies, how it compares to the training set log likelihood, and why. Finally, comment on how train and test set performance with the "true" number of clusters (4) compares to more and fewer clusters and why.