Homework 1

Due at noon on January 27, 2016

GUIDELINES: You can brainstorm with others, but please solve the problems and write up the answers by yourself. You may use textbooks (Koller & Friedman, Russel & 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.

SUBMISSION INSTRUCTIONS: Submit this assignment by Dropbox. You submission should include: A PDF containing written answers; source code for the mixture model; and a README explaining how to compile and run the source code under Linux (e.g., tricycle) or Windows (e.g., aqua).

1 PROBABILITY THEORY (25 POINTS)

(a) **Sum and Product Rule (5 points)**

Suppose that we have three colored boxes r (red), b (blue), and g (green). Box r contains 3 apples, 4 oranges, and 3 limes, box b contains 1 apple, 1 orange, and 0 limes, and box g contains 3 apples, 3 oranges, and 4 limes. If a box is chosen at random with probabilities $p(r) = 0.2$, $p(b) = 0.2$, $p(g) = 0.6$, and a piece of fruit is removed from the box (with equal probability of selecting any of the items in the box), then what is the probability of selecting an apple? If we observe that the selected fruit is in fact an orange, what is the probability that it came from the green box?

(b) **Reasoning by Cases (4 points)**

An often useful rule in dealing with probability is known as *reasoning by cases*. Let X, Y, and Z be random variables, then

$$P(X|Y) = \sum_z P(X, z|Y).$$

Prove this equality using the chain rule of probabilities and basic properties of (conditional) distribution.

(c) **Conditional Independence (16 points)**

Prove or disprove (by providing a counterexample) each of the following properties of independence:

(i) $(X \perp Y, W|Z)$ implies $(X \perp Y|Z)$.

(ii) $(X \perp Y|Z)$ and $(X, Y \perp W|Z)$ imply $(X \perp W|Z)$.

(iii) $(X \perp Y, W|Z)$ and $(Y \perp W|Z)$ imply $(X, W \perp Y|Z)$.

(iv) $(X \perp Y|Z)$ and $(X \perp Y|W)$ imply $(X \perp Y|Z, W)$.

2 Bayesian Analysis of the Uniform Distribution (25 points)

(a) **(10 points)** Consider the uniform distribution Unif$(0,\theta)$. The maximum likelihood estimate is $\hat{\theta} = \max \mathcal{D}$, 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, $p(\theta) = \text{Pareto}(\theta|b,K)$, the pdf of which is defined in the following form

$$\text{Pareto}(\theta|b,K) = Kb^K\theta^{-(K+1)}\mathbb{I}(\theta \geq b).$$

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

$$p(\mathcal{D},\theta) = \frac{Kb^K}{\theta^{N+K+1}}\mathbb{I}(\theta \geq max(\mathcal{D}))$$

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

$$p(\mathcal{D}) = \int_m^\infty \frac{Kb^K}{\theta^{N+K+1}}d\theta \tag{2.1}$$

$$= \begin{cases} \frac{K}{(N+K)b^N}, \text{ if } m \leq b \\ \frac{Kb^K}{(N+K)m^{N+K}}, \text{ if } m > b. \end{cases} \tag{2.2}$$

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

(b) **(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.

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

(ii) 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 hyper-parameters, $\beta = (c, N + K)$ are the updated hyper-parameters. Now consider the case $D = \{m\}$, and $D' = \{x\}$. Using $P(\mathcal{D})$ derived in question 2.(a), write down an expression for $p(x|D, \alpha)$. As above, use a non-informative prior $b = K = 0$.

3 PROGRAMMING (50 POINTS)

EM. 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.2, feature 1> <ex.2, feature 2> ... <ex.2, feature n>
...
```

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 Seeds dataset, available from the course Web page. Each data point represents a wheat kernel, with features representing geometrical properties of kernels. We provide a single default train/test split with the class removed to test generalization. You can find the full dataset and more information in the UCI repository (and linked from the course Web page). Start by using 3 clusters, since the Seeds dataset has three different classes. 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 (see Seeds-true.data) compare to yours?

(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 (3) compares to more and fewer clusters and why.