CSE 590ST: Statistical Methods in Computer Science

Homework 3

Due in class on May 19, 2004

The purpose of this homework is to test your understanding of Bayesian estimation of parameters, and using the EM algorithm to learn parameters for Bayesian networks.

1. **Tossing Thumbtacks**. Suppose you toss a thumbtack and it comes up heads 3 times and tails 7 times. Let θ be the parameter of the thumbtack which determines its probability of landing heads.
    a. What is the maximum likelihood (ML) estimate for θ?
    b. Suppose your prior belief in P(heads) is the beta distribution, with $\alpha_h=\alpha_h=\alpha$. Plot the posterior distribution for θ when $\alpha=1$, $\alpha=2$, $\alpha=10$, and $\alpha=100$.
    c. What is the probability of seeing heads on the next toss of the thumbtack (equivalently, what is the expected value of θ) for each $\alpha=\{1, 2, 10, 100\}$?
    d. Derive the equation for the MAP estimate of θ. What is the MAP estimate of θ for each $\alpha=\{1, 2, 10, 100\}$? How do these relate to the graphs in (b)?
    e. Write a sentence or two describing the relation between the ML estimate, the MAP estimate, and the strength of the prior (α).

2. **EM**. Write a program that learns parameters for Bayesian networks by using the EM algorithm. This program will need to take as input a Bayesian network, and some training data, and will output a Bayesian network, with parameters trained according to the training data. You will also need to be able to vary the amount of training data the program uses (e.g. truncate the training data to the first N instances), and make some of the attributes unknown (e.g. for each attribute of each data instance, make it unknown with some probability U. Note that the decision of which attributes become missing is done independently for each training example, so the attributes that become missing vary from training example to training example).

    You then need a second program which takes as input a Bayesian network, and test data, and outputs the average log-likelihood of the network on the test data. There is already one such program, *beliefnetscore*, in VFML, which you may use. In case you need to write your own, the equation for average log-likelihood is given in the appendix.

    From http://www.cs.washington.edu/education/courses/cse590st/CurrentQtr, download the alarm network (alarm.bif), the training data (atrain.data and atrain.names), and the test data (atest.data and atest.names). The alarm network is a network by medical experts for monitoring patients in intensive care. Using **EM** with

maximum-likelihood parameter estimation, fill in the following table with the log-likelihood of your trained model on the test data (using atrain for training, and atest for testing):

(Log-Likelihood)		Number of training examples		
		100	1000	10000
Probability that each attribute value is missing (U)	0%			
	20%			
	50%			

What do you conclude about the effect of the fraction of missing data and the number of samples on the quality of maximum-likelihood estimates in this domain?

3. Now, assume a Dirichlet prior with $\alpha_i=2$ for all parameters of all rows of all CPTs (this is equivalent to initializing all your counts with 1). Perform the same set of 9 experiments as you did in problem #2, but using the **MAP** parameter estimate instead.

(Log-Likelihood)		Number of training examples		
		100	1000	10000
Probability that each attribute value is missing (U)	0%			
	20%			
	50%			

What do you conclude about the effect of the fraction of missing data and the number of samples on the quality of MAP estimates in this domain?

4. For 20% missing data and 1000 training examples, vary the value of α_i in your prior. Create a plot with α_i on the x-axis and the log-likelihood of the resulting model on the y-axis. Experiment for an interesting range of α_i. What seems to be the best value for α_i?

5. What do you conclude on the relative merits of maximum likelihood and MAP estimation in this domain?

Appendix

VFML

Notice the training and testing data each contain two files: x.names and x.data. The first file is used to define the specification of the data, and the second contains one data instance per line. The following code demonstrates reading the examples using the builtin VFML functions *ExampleSpecRead* and *ExamplesRead.*

```
// Set up the input data
sprintf(filename, "%s.names", fileStem);
es = ExampleSpecRead(filename);
DebugError(es == 0, "Unable to open the .names file");

sprintf(filename, "%s.data", fileStem);
FILE *in = fopen(filename, "r");
DebugError(in == 0, "Unable to open the .data file");

VoidListPtr examples = ExamplesRead(in,es);
DebugError(!params->gDataMemory, "Unable to read the .data file");
fclose(in);

// Walk through the list of examples:
for (i=0; i<VLLength(examples); i++) {
  ExamplePtr example = (ExamplePtr)VLIndex(examples, i);
  //
  // Do something with example here
  //
}
```

Of course, you can also write examples in a similar fashion.

Log-Likelihood

The average log-likelihood of a set of test data, *D*, is given by:

$$avgLL(D) = \frac{1}{n}\sum_{i=1}^{n}\log\left(P(x_i)\right)$$

where x_i is the i^{th} data instance, and $P(x_i)$ is simply the probability of the instance according to the Bayesian network. Recall that this decomposes to:

$$P(x_i) = \sum_j P(x_{ij} \mid pa(x_{ij}))$$

where x_{ij} is the value assigned to the j^{th} node in the i^{th} example.