# Computer Science 6100/4100: Assignment 1

Due: September 29, 2008

**Note:** The first problem consists of programming and writing a report. It may be done either individually or in a team of up to 2 people. The other problems must be done individually (although you are allowed to talk to each other about them – see the collaboration policy in the syllabus). If you are having difficulty finding a partner to work with for Problem 1, send me an e-mail and I will try to pair up interested people. Submit only one writeup for Problem 1, with the names of all members of the team. Submit the answers to other problems separately.
**A note on the writeup for Problem 1:** You must submit a maximum 3-page paper that strictly adheres to the official ACM style file that can be found here (use the "tighter alternate style"):
`http://www.acm.org/sigs/publications/proceedings-templates`
This means you have to make significant editorial decisions about what to include and what not to, but that's part of the challenge of academic writing, at least in Computer Science! If your writeup does not meet the formatting requirements I **will not read or grade it**. You rill receive a 0. I am serious. You will be graded as much on the quality of your writeup as on the design and evaluation of experiments.

## 1 Problem 1 (60 points)

For this problem you will implement the Naive Bayes algorithm discussed in class and experiment with its performance on a dataset from the UCI repository of machine learning databases, which can be found at:
`http://mlearn.ics.uci.edu/MLSummary.html`

The particular database on which you will experiment with Naive Bayes is the "Adult" (formerly "Census Income") database which can be found at:
`ftp://ftp.ics.uci.edu/pub/machine-learning-databases/adult/`
The file `adult.names` contains a description of the data, the file `adult.data` contains the training data, and the file `adult.test` contains a separate test set.

### 1.1 Part 1: Implementation

First, implement the Naive Bayes classifier in a programming language of your choice (I recommend using R or Matlab, they will make your life much simpler in terms of running experiments and presenting the data you gather from these experiments). In doing so you will have to decide how to deal with missing values and a mixture of continuous and discrete variables. You will also have to make a decision about what kind of smoothing to implement. Be sure to document your design decisions as you make them and then summarize them in the writeup. Write code after looking at the data to make sure you understand the formatting and the types of examples present in the training data.

### 1.2 Part 2: Estimating Accuracy With Small Training Sets

To start the experiments, construct two (random) subsamples of the training data, one with 100 examples (Sample A) and one with 500 examples (Sample B). When documenting your work be sure to provide some summary statistics about these samples (at least the percentages of positive and negative examples).

For Sample A, find the accuracies of Naive Bayes using ten-fold cross-validation and leave-one-out cross-validation (just on Sample A). For Sample B, find the accuracies of Naive Bayes using random training/test set splits for every training percentage from 10% to 90% of the data (again, just on Sample B, treat other data as if it didn't exist for the moment). Draw a learning curve based on these experiments, and also show 95% confidence intervals for each number plotted on the curve. Also use ten-fold cross-validation on Sample B to estimate the accuracy of the classifier.

Now train two classifiers, one using the entirety of Sample A (call it Classifier A) and another using the entirety of Sample B (call it Classifier B). Report the accuracies of these two classifiers on the held-out test set in the file `adult.test`. Which of your estimates based on Sample A came closest to the test accuracy of Classifier A? Which of your estimates based on Sample B came closest to the test accuracy of Classifier B? Why? Interpret your results.

### 1.3 Part 3: How Big/Small?

Finally, use the entire training set in `adult.data` to learn a classifier (Classifier C). Find out how well it performs on the held-out test set in `adult.test` in terms of accuracy. Devise an experiment to reliably estimate how many (randomly sampled) examples from the training set you would need in order to attain the kind of accuracy you attain using the entire training set. Report the results of this experiment, along with your interpretation.

### 1.4 Part 4: Reflection

Under a separate heading, write a brief summary of what you found to be the most challenging part of this problem and why. Also, comment on your choice of programming language, and how you think the experience would have been different if you had used a substantially different language.

### 1.5 A Note on Writing and Presenting Results

Please write up your implementation, experiments, and interpretations in a professional, well-formatted manner. I strongly recommend using LaTeX for the document and R or Matlab to produce graphs. If you haven't used these tools before, this is the best time to learn! You do not need to go into details of your code in the writeup, although I may ask some students to e-mail me their implementations. Be sure to state the language you used and present the design choices you made. Also include some summary information about how long it takes your code to train on differently-sized training sets, and how long it takes to classify an example (this is purely informational; you will not be assessed on the speed of your implementation, although it is a good idea to take efficiency into account, at least to the extent that it doesn't hold up your experiments). Present numerical results in tables and graphs, and spend some time thinking about (and writing about!) what your results mean – do not just present the graphs.

Keep in mind that these are suggestions – feel free to add anything else that you think is interesting to your report, or to run additional experiments and report on them. Finally, keep

in mind that good writing style and meaningful, comprehensible presentation of results are as important as the results themselves.

## 2 Problem 2 (15 points)

Let $X \sim \mathcal{N}(\theta, \sigma^2)$ where $\sigma^2$ is known and we are trying to estimate $\theta$. Suppose we start with a normal prior on $\theta$ with mean $\mu$ and variance $\tau^2$. Prove that after getting a sample $x$ and updating, the posterior distribution of $\theta$ is also normal, with mean $\frac{\tau^2}{\tau^2+\sigma^2}x + \frac{\sigma^2}{\sigma^2+\tau^2}\mu$ and variance $\frac{\sigma^2\tau^2}{\sigma^2+\tau^2}$. How do you think this result could be useful?

## 3 Problem 3 (10 points)

Let $X_1, X_2, X_3, X_4$ be random samples from a binomial $(k, p)$ population with unknown $k$ and $p = 0.5$. Suppose the random variables have realized values $X_1 = 0, X_2 = 20, X_3 = 1, X_4 = 19$. What is the maximum likelihood estimate of $k$?

## 4 Problem 4 (15 points)

Consider a regression problem that is set up exactly the same way as the case we did in class, except that we are regressing $x$ on $y$, rather than $y$ on $x$. Using the same notation as in class, derive regression coefficients by minimizing RSS. Does this yield the same line? Why or why not?