CSE 250A Assignment 4

This assignment is due at the start of class on Thursday November 1, 2012. Instructions are the same as for previous assignments. You must work in partnership with one other student, but you may keep the same partner or change partners, as you wish. Acknowledgment: These questions are versions of ones written by Lawrence Saul.

1. Statistical language modeling

This problem explores simple statistical models of English text. Download the data files `unigram.txt`, `bigram.txt`, `vocab.txt`, and `readme.txt` from the course web site. These files contain unigram and bigram counts for 500 frequently occurring tokens in English text. The tokens include actual words as well as punctuation symbols and other textual markers. In addition, an “unknown” token represents all words not in this basic vocabulary. Turn in a printed copy of all your source code and results for the following problems. You may program in the language of your choice.

(a) Compute the maximum likelihood estimate of the unigram distribution \( p_u(w) \) over words \( w \). Print out a table of all the words that start with the letter “s” with their numerical unigram probabilities.

(b) Compute the maximum likelihood estimate of the conditional bigram distribution \( p_b(w'|w) \). Print out a table of the ten most likely words to follow the word “the” with their numerical bigram probabilities.

(c) Consider the sentence “The stock market fell a hundred points on Friday.” Ignoring punctuation, compare the log-likelihoods of this sentence under the unigram and bigram models:

\[
U = \log[p_u(\text{the}) \cdot p_u(\text{stock}) \cdots p_u(\text{on}) \cdot p_u(\text{Friday})]
\]

\[
B = \log[p_b(\text{the}|\langle s \rangle) \cdot p_b(\text{stock}|\text{the}) \cdots p_b(\text{points}|\text{on}) \cdot p_b(\text{Friday}|\text{on})].
\]

In the equation for the bigram log-likelihood, the token \( \langle s \rangle \) marks the beginning of a sentence. Which model yields the highest log-likelihood?

(d) Consider the sentence “The agency officials sold securities.” Ignoring punctuation, compute and compare the log-likelihoods of this sentence under the unigram and bigram models. Are any pairs of adjacent words in this sentence not observed in the training corpus? What effect do such pairs have on the log-likelihood from the bigram model?

(e) Consider a so-called mixture model that predicts words from a weighted interpolation of the unigram and bigram models:

\[
p_m(w'|w) = \lambda p_u(w') + (1 - \lambda) p_b(w'|w)
\]
where $\lambda \in [0, 1]$ determines how much weight is attached to each model. Plot the log-likelihood of the sentence from part (d) under the mixture model as a function of the parameter $\lambda$. What is the optimal value of $\lambda$ to two significant digits, for this sentence?

2. Multinomial logistic regression

A generalization of logistic regression is to predict an $m$-valued label $y \in \{1, 2, \ldots, m\}$ given a real-valued vector $x \in \mathbb{R}^d$. Consider the model

$$p(Y = i | X = x) = \frac{\exp w_i \cdot x}{\sum_{j=1}^{m} \exp w_j \cdot x}.$$

The parameters of this model are one weight vector $w_i$ for each possible label value. Consider a dataset of $T$ training examples $\{(x_t, y_t)\}_{t=1}^T$. For notational convenience, define

$$y_{it} = \begin{cases} 1 & \text{if } y_t = i \\ 0 & \text{otherwise.} \end{cases}$$

Also, let $p_{it} \in [0, 1]$ be the conditional probability assigned by the model to the $i$th label value for the $t$th example:

$$p_{it} = \frac{\exp w_i \cdot x_t}{\sum_{j=1}^{m} \exp w_j \cdot x_t}.$$

The weight vectors can be obtained by maximum likelihood estimation using gradient ascent. The conditional log-likelihood is $L = \sum_t \log p(y_t | x_t)$. Show that its gradient is

$$\frac{\partial L}{\partial w_i} = \sum_t (y_{it} - p_{it}) x_t.$$

Essentially, the differences between observed values $y_{it}$ and predictions $p_{it}$ are error signals for learning.

3. Convergence of gradient descent

One way to gain intuition for gradient descent is to analyze its behavior in simple settings. For a one-dimensional function $f(x)$ over the real line, gradient descent takes the form

$$x_{n+1} = x_n - \eta f'(x_n),$$

where $\eta$ is the learning rate.
(a) Consider minimizing the function \( f(x) = (\alpha/2)(x - x^*)^2 \) by gradient descent, where \( \alpha > 0 \). Derive an expression for the error \( \epsilon_n = x_n - x^* \) at the \( n \)th iteration in terms of the initial error \( \epsilon_0 \) and the learning rate.

(b) For what values of the learning rate does the update rule converge to the minimum? What learning rate leads to the fastest convergence? How is it related to the second derivative \( f''(x_n) \)?

The gradient descent learning rule is sometimes modified by adding a so-called momentum term. In one dimension, the modified learning rule is

\[
x_{n+1} = x_n - \eta f'(x_n) + \beta (x_n - x_{n-1}).
\]

Intuitively, the name arises because the optimization continues of its own momentum (stepping in the same direction as its previous update) even when the derivative vanishes.

(c) Consider minimizing the function in part (a) by gradient descent with momentum. Show that the error at the \( n \)th iteration satisfies the recursion relation

\[
\epsilon_{n+1} = (1 - \alpha \eta + \beta) \epsilon_n - \beta \epsilon_{n-1}.
\]

(d) Suppose that \( \alpha = f''(x^*) = 1 \), \( \eta = 4/9 \), and \( \beta = 1/9 \). Show that one solution to the recursion in part (c) is \( \epsilon_n = c^n \epsilon_0 \) where \( \epsilon_0 \) is the initial error and \( c \) is a numerical constant that you must determine. (Other solutions are also possible, depending on the way that the momentum term is defined at time \( t = 0 \); do not concern yourself with this.) How does this rate of convergence compare to that of gradient descent with the same learning rate but no momentum (\( \beta = 0 \))?  

4. Stock market prediction

In this problem, you will apply a simple linear model to predict the stock market. Download the files nasdaq00.txt and nasdaq01.txt, which contain NASDAQ index values at the close of business days in 2000 and 2001, from the course web site.

(a) It is always important to understand the context of data. What important, relevant, event happened around March 2000? How does this event show up in the data?

(b) It is also always important to be confident about data validity. Use a resource of your choice (for example Yahoo Finance) to investigate whether the data appear to be correct. Can you find any discrepancies?

(c) For trading and for making predictions, changes in price are more significant than price levels. For this reason, transform the data so that \( x_t \) for day \( t \) is the change in the index from day \( t - 1 \) to day \( t \). Specifically, define \( x_t = \ln(v_t/v_{t-1}) \)
where \( v_t \) is the value of the index on day \( t \). Explain why it makes sense to use a logarithm in this formula. Also explain why the natural logarithm (base \( e \)) is the best logarithm to use.

(d) Explain how to compute linear coefficients \( w_1 \) to \( w_k \) that maximize the conditional log probability

\[
L = \sum_t \log p(x_t|x_{t-1}, \ldots, x_{t-k})
\]

where the conditional distribution of \( x_t \) is Gaussian:

\[
p(x_t|x_{t-1}, \ldots, x_{t-k}) = \frac{1}{\sqrt{2\pi}} \exp \left( -\frac{1}{2} [x_t - \sum_{i=1}^{k} w_i x_{t-i}]^2 \right).
\]

The outer sum is over business days in the year 2000. Which day \( t \) can you use as the first day for this sum? Implement the computation. In a table, show the learned coefficients for each \( k \).

(e) Using your implementation, compute the mean squared error of each model for \( k = 1 \) to \( k = 5 \) on the training data. Also, compute the mean squared error of the trained models for the year 2001. Show the results in a straightforward table.

(f) Explain conclusions that you draw from the table. Are your results consistent with general wisdom about machine learning? Which \( k \) between 1 and 5, if any, would you recommend for stock market prediction?

Attach your source code to your answers. You may use the programming environment of your choice, and you may solve optimization problems, and systems of equations, using library functions. However, do not use a library function for linear regression.