CS769 Advanced Natural Language Processing
Homework 1

Assigned 1/20/2010
Due 1/27/2010 before class

What to hand in: answer sheets, printed or neatly handwritten. You do not need to handin any code. Write your name, email, and hand in date on top of the answer sheet. See course webpage for homework policy.

1. (15) Solve x by hand.

$$\begin{pmatrix}1 & 1\end{pmatrix}\begin{pmatrix}1 & 2\\ 3 & x\end{pmatrix}\begin{pmatrix}1\\ 1\end{pmatrix} = x^2$$

2. (15) Compute the derivative (with respect to x) of the function

$$\frac{1}{1+e^{-x}}$$

3. (15) Find the minimum of the function $f(x, y) = x + y$, where (x, y) must be on the unit circle.

4. (15) Let x be a random variable drawn from a Gaussian distribution with mean 0 and variance $\frac{1}{2\lambda}$. Write down the expression for $\log p(x)$.

5. Download the particular version of *Alice's Adventures in Wonderland* from `http://pages.cs.wisc.edu/~jerryzhu/cs769/dataset/alice.txt`. This is the document we'll be working on.

   (a) Sentence Segmentation. Download MXTERMINATOR, a sentence boundary detector, from `http://pages.cs.wisc.edu/~jerryzhu/cs769/code/jmx.tar.gz`. Follow the instruction in `MXTERMINATOR.html`. If you use *tcsh*, simply do
   `setenv CLASSPATH mxpost.jar`
   then you should be able to run it. Use the `eos.project` that comes with the package. Apply it to *Alice*.

   (b) Tokenization. Once you have segmented out sentences, it's time to separate individual words. Download the Penn Treebank tokenizer from `http://pages.cs.wisc.edu/~jerryzhu/cs769/code/tokenizer.tar.gz`. This is a UNIX *sed* program. Run it with `sed -f`. It needs an input file with one sentence per line. Apply the tokenizer to the processed *Alice* corpus.

(c) Stemming. Download and compile the Porter stemmer from `http://pages.cs.wisc.edu/~jerryzhu/cs769/code/porter.c`. Run the stemmer on *Alice* from the previous step. You will notice that it maps all words to lower case, and some words look funny.

Question 5.1. (10) Do not strip punctuations or otherwise change the tokens out of the stemmer. How many *word tokens* and *word types* are there?

Question 5.2. (10) List the top 10 most frequent words (they can be punctuations) and their counts.

Question 5.3. (10) In Matlab, plot rank r (x-axis) vs. count f (y-axis) for all words. Each word would be a dot in such a plot. In a second plot, plot the same thing but use log scale on both axes.

Question 5.4. (10) Assume the following relation: $f = ar^b$. Use Matlab `polyfit` function to find a, b. Hint: take log on both sides.