Project Assignment 4 - Winter 2011 Document Categorisation with Latent Dirichlet Allocation

Andrew Heiberg
aheiberg@ucsd.edu

André Christoffer Andersen
andre@andersen.im

Fabian Siddiqi
fsiddiqi@ucsd.edu

March 17, 2011

Abstract

This paper discusses the challenges and final results of performing multi-label classification on text documents via Latent Dirichlet Allocation (LDA). Two different datasets, the *classic400* and the *Reuters-21578*, are analyzed. To begin, the LDA algorithm is presented along with a brief discussion of its input parameters. Following this, optimizations to this algorithm, as well as the pre-processing steps taken for the Reuters dataset, are presented. Next, the experiments are run and the obtained results are examined. Techniques for comparing the output of LDA to the true document labelings are given particular attention. This leads to analyzing why LDA outputs qualitatively distinct topics for the *classic400* but failed to do so for the *Reuters-21578*. The concluding discussion investigates the limitations on finding optimal input parameters, a method for comparing the suitability of LDA outputs, and the detection of overfitting data.

1 Introduction

Text mining can be defined as the automatic extraction of information from sets of sentences or documents by use of statistical techniques. Many of these methods can be generalized and applied to other data types which exhibit a similar structure to text documents, such as genomic data, and is thus an important research area. Other applications include document classification or BioNLP (natural language processing of biomedical literature).

LDA is an unsupervised learning scheme which requires a *bag-of-words* representation of a set of documents (i.e. the order of the words is lost). Documents are classified into K different topics (this parameter is fixed and given). Each of the M documents has an associated topic distribution $\{\theta_m\}_{m=1}^{M}$ and each topic has an associated word distribution $\{\phi_k\}_{k=1}^{K}$, both of which are multinomials. θ_m determines the probability that any word w within document m is of topic k, while ϕ_k indicates the probability of word w being of topic k throughout the whole corpus. These distributions are drawn from two different Dirichlet distributions:

$$\theta \sim \mathrm{Dir}(\alpha)$$
$$\phi \sim \mathrm{Dir}(\beta)$$

where α and β are parameter vectors of length K and V (length of the dictionary) respectively. Both are fixed and given. The only constraints are that $\alpha_i, \beta_i > 0$ for all i. Dirichlet distributions are used since they are the conjugate prior of multinomial distributions.

To define the mentioned Dirichlet distributions, it is necessary to create a dictionary of length V which contains the most informative words. Very frequent and uninformative terms are discarded (e.g. *the*, *and*, etc. which are called stop-words) as well as words which occur only once since they are generally misspellings or a hapax legomenon that don't carry over to other documents. The corpus is thus represented as a sparse $V \times M$ matrix, where each element x_{tm} is the word count for term t in document m. The length of document m is therefore $n_m = \sum_{t=1}^{V} x_{tm}$.

Though LDA requires a *bag-of-words* representation, each position within the corpus (of which there are $N = \sum_{m=1}^{M} n_m = \sum_{m=1}^{M} \sum_{t=1}^{V} x_{tm}$) is considered, each of which has an associated topic. These are denoted $\{z_i\}_{i=1}^{N}$. When applying LDA, each z_i is sampled from a distribution depending on the words

and topics of the other positions as well as the fixed α and β parameter vectors. This technique is referred to as Gibb's sampling. Once the distribution of each z_i value has stabilized after a sufficient number of iterations, we can infer that the algorithm has converged. Finally, $\{\theta_m\}_{m=1}^{M}$ and $\{\phi_k\}_{k=1}^{K}$ can be inferred from these values.

We can therefore say that, given $\{\alpha, \beta, K, M, \{n_m\}_{m=1}^{M}, |V|\}$, the end-goal of training using LDA is to determine $\{\theta_m\}_{m=1}^{M}$ and $\{\phi_k\}_{k=1}^{K}$.

Building on this brief introduction to LDA, a more detailed explanation of the algorithm is shown in Section 2. Section 3 describes experiments which performed to analyze the accuracy of the system, while results are shown in Section 4. A conclusion or discussion can be found in Section 5.

2 Design of Algorithms

As described in Section 1, calculation of the z-values is the main step in applying the Latent Dirichlet Allocation algorithm. The set of all z-values and words are subsequently denoted $\bar{z}$ and $\bar{w}$, while the set of all z-values except for z_i (z-value at position i) and words except w_i are denoted $\bar{z_i}'$ and $\bar{w_i}'$ respectively. $\bar{w}$ is not a word count, though the actual order of the words within a document is arbitrary (this assumption is acceptable since only *distributions* are being considered).

During the training step, the value of z_i is updated by sampling from the following distribution:

$$p(z_i|\bar{z_i}', \bar{w}) = \frac{p(\bar{w}|\bar{z})p(\bar{z})}{p(w_i|\bar{z_i}')p(\bar{w_i}'|\bar{z_i}')p(\bar{z_i}')} \propto \frac{p(\bar{w}|\bar{z})p(\bar{z})}{p(\bar{w_i}'|\bar{z_i}')p(\bar{z_i}')}$$

By calculating each of the above probabilities, it is possible to reduce the expression to known values and parameters:

$$p(z_i = j|\bar{z_i}', \bar{w}) \propto \frac{q'_{j,w_i} + \beta_{w_i}}{\sum_{t=1}^{V} q'_{j,t} + \beta_t} \cdot \frac{n'_{m,j} + \alpha_j}{\sum_{k=1}^{K} n'_{m,k} + \alpha_k} \tag{1}$$

where q_{j,w_i} is the number of times word w_i occurs in topic j, α_j and β_{w_i} is the corresponding Dirichlet parameters and $n_{m,j}$ is the number of words in topic j that occur in document m. As before, the $'$ symbol indicates that the corresponding value at position i is being ignored or is considered to be unknown when making this calculation. When implementing the algorithm,

the second denominator is ignored since it does not depend on j for a given i. Equation 1 can be further simplified:

$$p(z_i = j|\bar{z_i}', \bar{w}) = \frac{1}{Z}\frac{q'_{j,w_i} + \beta_{w_i}}{Q'_j + \beta} \cdot (n'_{m,j} + \alpha_j) \tag{2}$$

where $\beta = \sum_{t=1}^{V} \beta_t$, $Q'_j = \sum_{t=1}^{V} q'_{j,t}$ and Z is the partition function (which bounds all probabilities to the range $[0, 1]$), defined as follows:

$$Z = \sum_{j=1}^{K} \frac{q'_{j,w_i} + \beta_{w_i}}{Q'_j + \beta} \cdot (n'_{m,j} + \alpha_j)$$

One epoch of the training scheme involves updating all z_i once. $\bar{z}$ is initialized to random values within the set $\{1, 2, \ldots, K\}$. When a z_i changes topics, the appropriate q, Q and n values can be updated in constant time.

2.1 Convergence

The algorithm is stopped when the distribution of $\bar{z}$ converges (not the exact values of $\bar{z}$). However, to prevent a decrease in speed, a 10% random subsample of positions is considered. When the averaged mean-squared-error of these distributions falls below an acceptable threshold, typically $0.001 < \gamma < 0.01$ or 50 epochs, the training process is terminated. This is done in order to allow for a burn-in' period that ensures that we are sampling from the true distribution.

2.2 Optimization

To sample a new z-value, Equation 2 must be computed K times. To avoid this, an upper bound on Z, $\hat{Z}$, is computed. Using $\hat{Z}$ gives a lower bound on each $p(z_i = k|\bar{z}', \bar{w})$. If the random number r is less than this value, we are done. If not, the estimated probability of the next label (added to the sum of the previous estimated probabilities) is compared with r. Note that the proportionality of the topics is preserved.

If either r or $\hat{Z}$ is large, there is a distinct chance r will be greater than the sum of underestimated probabilities and reach the end of the loop without returning. In this case, all the unweighted probabilities have been computed, so deriving the weighted probabilities and sampling the label as before is possible.

Computing $\hat{Z}$ can be done by using

$$Z(i) = \sum_{j=1}^{K} (q'_{jw_i} + \beta_{w_i})(Q'_j + \beta)^{-1}(n'_{mj} + \alpha_j)$$

Removing the dependence of the current topic on i

$$Z(i) \leq \sum_{j=1}^{K} (q_{jw_i} + \beta_{w_i})(Q_j - 1 + \beta)^{-1}(n_{mj} + \alpha_j)$$

By the generalized Holder inequality we have

$$Z(i) \leq ||q_{jw_i} + \beta_{w_i}||_p \cdot ||Q_j - 1 + \beta^{-1}||_q \cdot ||n_{mj} + \alpha_j||_r$$

where $\frac{1}{p} + \frac{1}{p} + \frac{1}{r} = 1$.

As suggested, three choices for p, q, r were entertained $\langle 2, 2, \infty \rangle$,$\langle 2, \infty, 2 \rangle$, and $\langle \infty, 2, 2 \rangle$. With some basic algebra, every time a topic changes the p, q, r norms can be updated in constant time. When it is time to compute $\hat{Z}$, the minimum of the the choices above is used to provide the tightest bound on Z thereby increasing the chances that the algorithm can escape having to compute the probability for every label. One caveat is that this optimization will have to be forgone for the specific positions which we are using to check for convergence. In these positions, we need the full, true distribution of labels.

2.3 Pre-processing

The Reuters data is a collection of text files with one or more topic labels. These documents were converted into *bag-of-words* and stored in a sparse matrix, a la the *classic400* dataset. Text tokenization was done using the NLTK RegExp tokenizer; only contiguous blocks of alphabetic characters were considered tokens. We used the NLTK Porter stemmer to stem similar words to their roots. This consolidates different parts of speech to one base word. Of all the word stems present in the files, the most frequent 20 were excluded from the dictionary, as were words with only one occurrence.

3 Design of Experiments

When executing the LDA algorithm, the hyperparameters α and β are fixed and given.

Given a set $\{\theta_i\}_{i=1}^{M}$ with one distribution per document, it is known that these K-dimensional distributions lie on a $(K-1)$-simplex in $\mathbb{R}^K$ where each corner represents a topic, since $\sum_{i=1}^{K} \theta_{i,m} = 1$ for each document m. The error produced by LDA given α and β is therefore calculated by averaging the Euclidean distance between each point θ_m on the simplex and the ground truth. Labels are given for both datasets, though they are only used to determine the accuracy of the system, not during the training phase. For this to be possible, each label is transformed to one of the corners of the simplex. [1]

The length of α is K and the length of β is V. For the first dataset, *classic400*, $[K, V] = [3, 6205]$. The high-dimensionality of these vectors prevented the authors from performing parameter sweeps. Initial 'base vectors', α_b and β_b, are chosen and scaled by a factor μ. The parameter μ is swept and an optimal value chosen.

Choosing α_b The set of all θ_m distributions are sampled from $\mathrm{Dir}(\alpha)$. The larger the parameter α_k, the higher the average value of $\theta_{k,m}$ when sampling from the Dirichlet distribution (i.e. given a very large α_k, most documents will have a large proportion of words in topic k). To prevent any topic bias, $\alpha_b = [1, 1, 1]$.

Choosing β_b Similarly, each distribution ϕ_k is sampled from $\mathrm{Dir}(\beta)$. Given a large parameter β_t translates to each topic having a high probability for word w_t. Since topics are known for both datasets, an initial histogram, h, of word frequencies is generated. This provides a base parameter vector: $\beta_b = h_n$, where h_n is the normalized histogram (such that $h_{\max} = 1$ and all other bins are scaled accordingly).

Choosing these base vectors is acceptable since the Dirichlet distributions express prior knowledge about the word and topic distributions. [2] Another intuition about Dirichlet distributions that proves to be useful is consideration of the average parameter value, $\bar{\alpha} = K^{-1} \sum_{i=1}^{K} \alpha_i$, examples of which are

[1] Answer to question 4 in CSE250B (WI'11) project assignment 4

[2] Answer to question 3 in CSE250B (WI'11) project assignment 4

shown in Figure 1. If $\bar{\alpha}$ is small, sampled distributions tend to be polarized at the simplex corners; large $\bar{\alpha}$ results in sampled distributions tending towards the middle of the simplex (not strongly identified with any given topic).

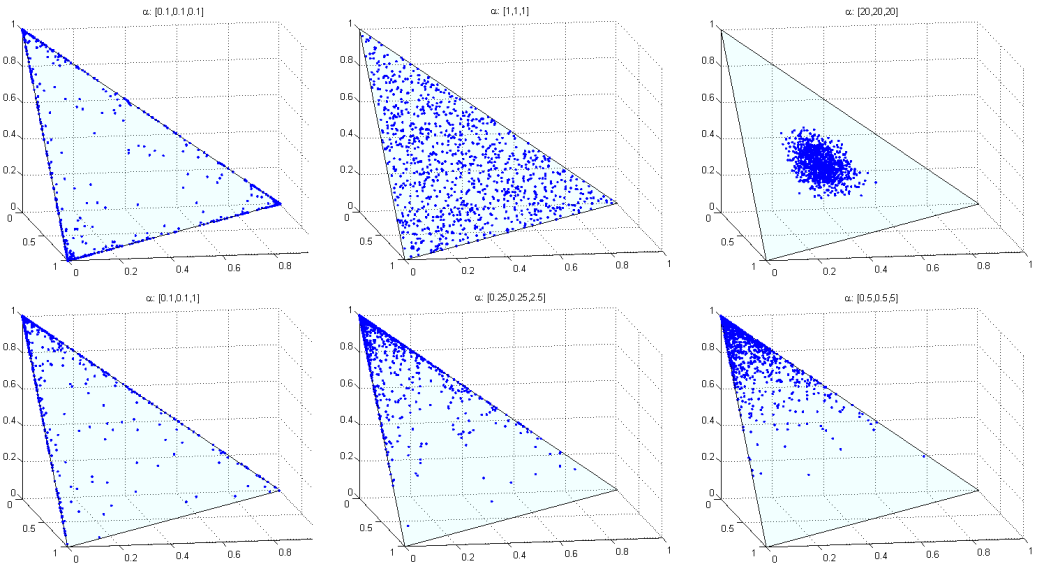

Figure 1: Examples of sampled distributions from a Dirichlet with a parameter vector of length 3. In the top row, all parameters are equal. In the second row, on parameter is ten times larger than the others. For all parameteres equal to 1, there is no bias towards any of the corners.

A 10,000 document subset of the *Reuters-21578* dataset is also analyzed. This data set contains news stories from Reuters, mostly concerning world economics, food supply, commodity prices, etc. There are 36 topic labels, ranging from 'earn' to 'corn' to 'rubber' to 'livestock'. The topic counts are unevenly distributed: the maximum number of documents per topic is 3964, the minimum number of documents per topic is 2, the mean is 148.11 and the standard deviation is 489.10.

The experiment looks at documents that have only one topic label, provided that topic is among the five most popular. The number of documents for each topic is capped at 300. The resulting document set contains 784 documents and a vocabulary of 7907.

Two different values for the hyperparameters are used:

1. $\alpha = 1$ and $\beta = 1$
2. $\alpha = \frac{50}{K}$ and $\beta = 0.01$.

In order to analyze the results, the error is calculated relative to the true labels. Additionally, the most common words from each topic will be examined to see if the chosen topics make semantic sense.

4 Results of Experiments

4.1 Classic400

To determine the optimal value for the hyperparameters α and β, the scaling factor μ is swept for various values. The optimal value $\hat{\mu}$ was found to be 1. The training algorithm was executed 140 times and the results averaged. A results for a single execution of the algorithm as well as the averaged results can be seen in Figure 2.

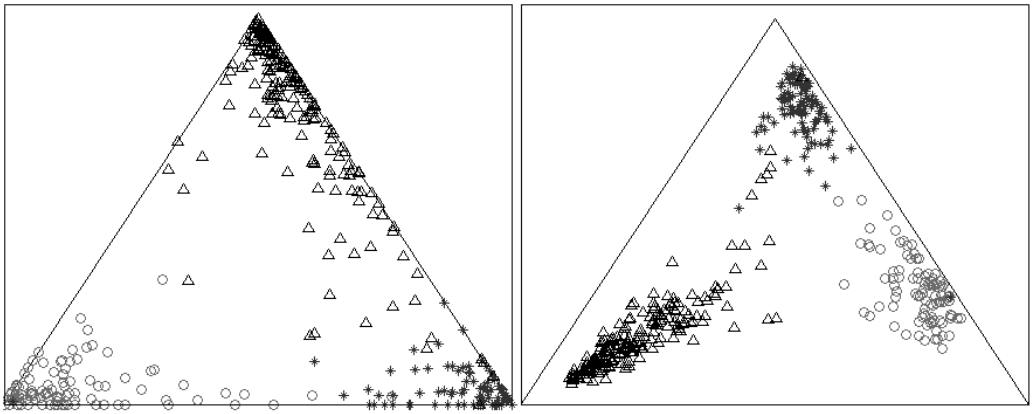

Figure 2: Each point on the simplex represents a θ distribution and each corner, a topic. Each of the three markers indicates a document label (ground truth). Distributions from a single training are plotted on the left, while the same results are averaged after having executed the algorithm 140 times on the right. On average, each epoch took 4.30 seconds. The mean of the error is calculated to be $\mu = 0.42$ with a variance $\sigma^2 = 0.02$.

Once $\{\theta_m\}_{m=1}^{M}$ and $\{\phi_k\}_{k=1}^{K}$ are inferred from the dataset, it is possible to display the most common words in each of the three topics. Using the averaged results described in Figure 2, the most common words can be found in Table 1.

Topic 1	Boundary	Layer	Velocity	Solution	Plate
Topic 2	Wing	Mach	Supersonic	Wings	Lift
Topic 3	Patients	Ventricular	Cases	Fatty	Left

Table 1: Most common words for the *Classic400* dataset given three topics. These words are sampled from ϕ_k, where $k \in \{1, 2, \dots\}$. This is done by taking the c largest values within each of these distributions, where $c = 5$ in this table. Topic 1 and 2 show some overlap, while Topic 3 is completely independent. Similar results are shown in Figure 2.

4.2 Reuters-21578

Multiple models are generated using the newspaper dataset. By setting $\alpha = \beta = 1$, the overall error is calculated to be 0.85. When changing the hyperparameters to $\alpha = \frac{50}{K}$ and $\beta = 0.01$ the error is decreased to 0.76. K is chosen to be 5. Examples of the most common words for each of the five topics are shown in Table 2.

Topic 1	Hungary	Weaken	Dominated	Miller	Panels
Topic 2	Renegotiated	Area	Purchasing	Criticism	Dave
Topic 3	Degrade	Edged	Spectacular	Implying	Retained
Topic 4	Vulnerability	Costs	Effot	Willingness	Ad
Topic 5	Fawc	Four	Rocketing	Woods	Regularize

Table 2: Most common words for the *Reuters-21578* dataset given five topics.

5 Findings & Analysis

5.1 Classic400

Classification of documents in the *Classic400* dataset using LDA proved to be effective. As can be seen in Figure 2, three distinct clusters are formed (produced by LDA) and almost all points within the same cluster have the same marker (ground truth). These visual representations provide intuitive insight into the efficacy of the algorithm, or lack thereof, and are complimented by printing the most common words within a topic as shown in Table 1. It is important, however, to divise a quantitative metric to be able to objectively distinguish between one model or another. The method used is described in Section 3.

5.2 Reuters-21578

When evaluting the *classic400*, the error measure, e, that corresponds to good clustering in the simplex visualizaiton is $e \in (0.2, 0.4)$. The Reuters data gave larger errors. However, the error number depends on Euclidean distance. As the number of dimensions (in our case, topics) increases, the distance between all points gets larger, meaning the e will have to increase as well. Unfortunately, we did not devise a procedure for visualizing or clustering higher dimensions.

Inspecting the word distributions does not give the topic distinctions one would expect based on the true labelings, namely 'corn', 'livestock', 'trade', 'coffee' and 'money-supply'. The highest probability words across the topics are adjectives and verbs, e.g 'vulnerability' and 'dominated'. The nouns present are not ones that are typically associated with corn, coffee, livestock, etc. Additionally, the appearance of words such as 'Dave', 'Miller' and 'Ad' are cause for concern. These suggest that the pre-processing of the data somehow missed the removal of noisy words such as author names or ads, though an inspection of the data set does not suggest this is the case.

Our hypothesis as to why the topics do not show clear semantic distributions is that the overall topic, the status of world affairs, and the short, no-frills writing style of the articles were not sufficiently heterogeneous enough in terms of vocabulary used across different topics for the model to detect. For example, in the *classic400* dataset, the topics were markedly different: our guesses based on the most frequent words are aerospace engineering, medical

technology, and fluid dynamics. These are much more disparate than the the statuses of corn and livestock markets.

Nevertheless, the LDA model attempts to balance words that occur across topics, e.g 'purchasing', and put more weight on unique words such as 'corn', 'husk', etc. and it is surprising the results were not better.

5.3 Optimization

The optimization discussed in the algorithms section was used in doing LDA for the Reuters data. This modified algorithm first was tested on the *classic400* data to ensure the results were consistent with the standard LDA algorithm. The equation we wish to avoid evaluating for each topic label k is:

$$p(z_i = j|\bar{z_i}', \bar{w}) \propto \frac{q'_{j,w_i} + \beta_{w_i}}{\sum_{t=1}^{V} q'_{j,t} + \beta_t} \cdot \left(n'_{m,j} + \alpha_j\right)$$

Table 3 lists the amount of computations saved depending on the amount of topics being used.

Number of Topics (K)	**Computations Saved**
3	0.82
5	1.33
⋮	⋮
36	9.14

Table 3: Average number of saved computations as a function of the number of topics.

6 Conclusion and Limitations

Ideally, we would like to find optimum hyperparameters α and β rather than make educated guesses and manually compare evaluated results. According to the Maximum Likelihood principle, we would like to maximize $p(w|\alpha, \beta)$. Running LDA using these hyperparameters as a starting point

would generate the most suitable and informative estimated parameter sets: $\underline{\Theta} = \{\theta_m\}_{m=1}^{M}$ and $\underline{\Phi} = \{\phi_k\}_{k=1}^{K}$.

For a given document m, the likelihood can be generated by a particular hyperparameter pair as follows:

$$p(w_m|\alpha,\beta) = \iint \left[\prod_{n=1}^{N_m}\sum_{k=1}^{K} p(w_{mn}|\phi_k)p(\text{topic} = k|\theta_m)\right] p(\theta_m|\alpha)p(\underline{\Phi}|\beta)d\underline{\Phi}d\theta_m \tag{3}$$

The first term defines the probability of the observed words in the document being generated by a particular choice of $\underline{\Theta}$ and $\underline{\Phi}$. The probability of a position instantiating a particular word is determined by marginalizing that word's probability over all possible topics. The product of these probabilities will then yield the probability of the entire observed document being generated. $p(\theta_m|\alpha)$ and $p(\Phi|\beta)$ are defined by the Dirichlet distribution and give the probability of seeing the particular multinomial distributions we are using in the first term to generate words. [3]

The difficulty lies in the evaluating these terms for all possible choices of θ_m and $\underline{\Phi}$. As the number of potential words and topics increases, directly computing Equation 3 becomes impractical.

However, it is possible to select α and β, run LDA, and use the generated $\underline{\Theta}$ and $\underline{\Phi}$ to compute $logp(w|\underline{\Theta},\underline{\Phi})$. It is not ideal, but it is nevertheless a quantifiable measure of the appropriateness of the model. This measure could be used to compare the effectiveness of different initialization strategies. Unfortunately, there is no way to take this measure and determine what changes to α, β and K are improving. [4]

Using this log likelihood measure was not necessary for the *classic400* data because the trained model could be visualized. For the Reuters data, where visualization was not performed, using log likelihood would have given a more rigorously defined measure with which to compare the effects of the input hyperparameters.

Overfitting can roughly be defined as the inability of a model to generalize to new data. In contrast with supervised learning, unsupervised learning lacks a general way to quantify the amount of overfitting present in a model. As with many types of models, LDA is prone to overfitting from the outset. The choices of α, β and K have enormous impacts on the generated $\underline{\Phi}$

[3] Answer to question 1 in CSE250B (WI'11) project assignment 4

[4] Answer to question 2 in CSE250B (WI'11) project assignment 4

and $\underline{\Theta}$. For instance, if there are only two categories salient from a human perspective, yet we instruct LDA to find 5 categories, the topic and word frequencies discovered will bear little relation to the two ideal categories.

Perhaps one way to detect overfitting would be to train the model on two disjoint random subset of the data. If the topic and word distributions vary significantly between the subsets, we could declare the model unsuitable. For stable results we could also redo this process a sufficient number of times in order to get averages. Failing to discern between data sets drawn random from the same distribution would be a indication that model is finding spurious distinctions. [5]

Even though we have described a procedure for detecting overfitting we did not actually implement it. This means that the models presented in this paper may or may not have low classifying ability for unseen data points. Time permitting, it would have been interesting to see how varying K affected the models.

[5] Answer to question 5 in CSE250B (WI'11) project assignment 4