CIS 520 Final Project Report

MengYao Li
Sanjian Chen
Varun Aggarwala
Group name: Awesome Dudes

Introduction:

For the final project for the machine learning class, we developed and tested various learning algorithms for predicting age and gender for two different kinds of data: blog postings (text) and faces (images).

Blogs:

The blogs dataset was an extract of postings from 1999 blogs out of which we had access to 1700 and the rest 299 were kept for testing. We were given the training data (the 1700 blogs) in tokenized form.
The task was to predict the gender of the people and the age group in which they fall in.

Our Approach: We tried a plethora of learning methods for this task. The method which ultimately helped us in beating the baselines and securing an accuracy of over 73% on test data was relatively very simple. Before submitting our results for the nightly checkpoint we used to set aside 200 blogs for development set, and we only submitted our code which did well on the development set. Although we never had access to the test data but if we had the access to it, then this is of paramount importance as our algorithm should be learnt without any access to test data at all.

Analyzing the training data using the *print_blog* function we came to the conclusion that these blogs are full with noisy features, i.e. some useless words.
Therefore we decided to use the concept of *Information Gain* and we used a certain threshold to throw away the useless features.
The threshold value we used to throw away the features was 0.0005 for gender prediction and .0006 for age prediction. We found these values using 6 fold cross-validation. This simple method gave us the maximum accuracy amongst all methods we tried.

On the next page, we show some of the interesting visualizations of how information gain differs for words to words.
Some of these words are actually very informative. The best words for predicting the gender are cute, love, her, she, "!", cry. This makes lots of sense as some of these words are generally associated with females*. The best words for predicting the age group are im, lol, haha, school, work, homework. This too makes sense because generally school, work and homework are associated with people in the first age group*.
(*Please note, we do not believe in any sort of discrimination based on gender, age, sex or religion. We are just reporting our findings on the data set provided to us)

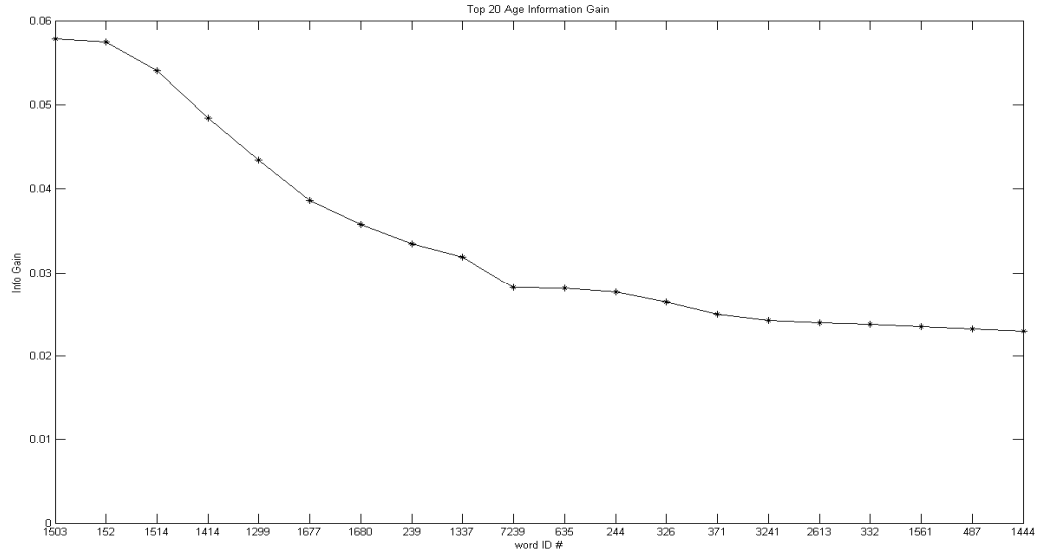

Figure 1: Plot of Information Gain for age vs. the top 20 words

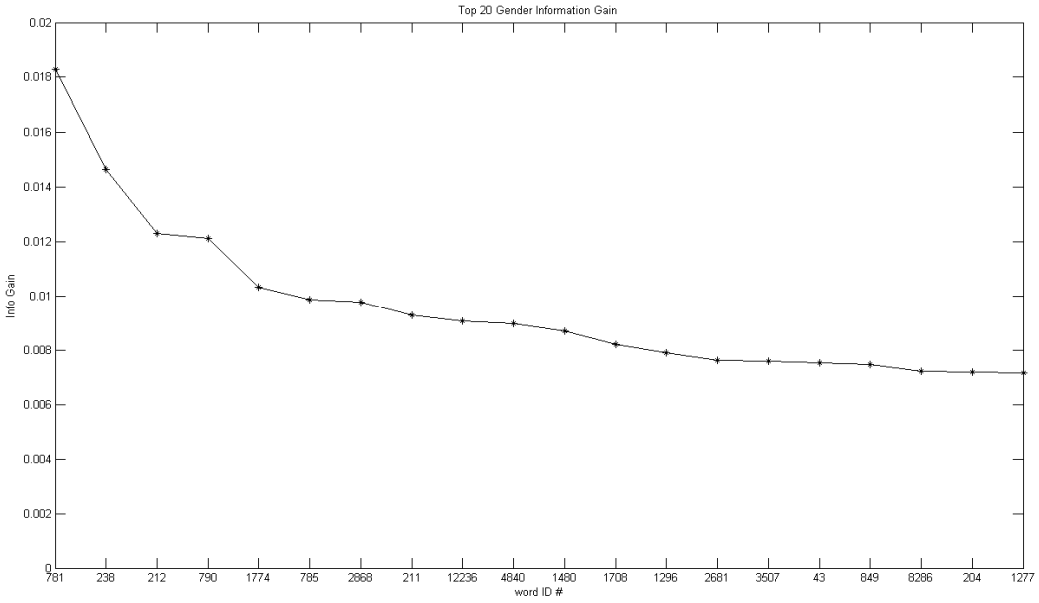

Figure 2: Plot of Information Gain for gender vs. the top 20 words

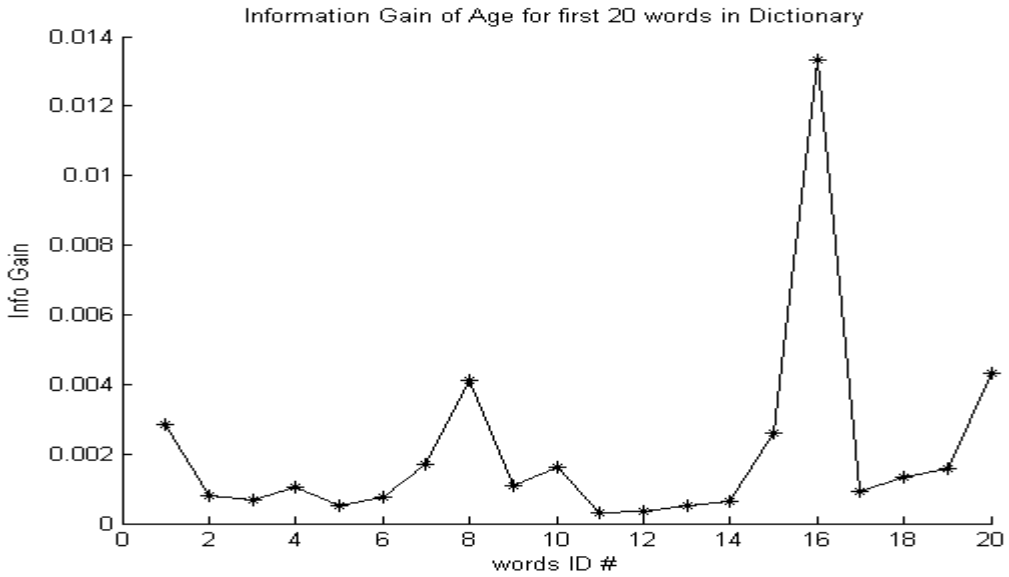

Figure 2: Plot of Information Gain for age vs. the first 20 words

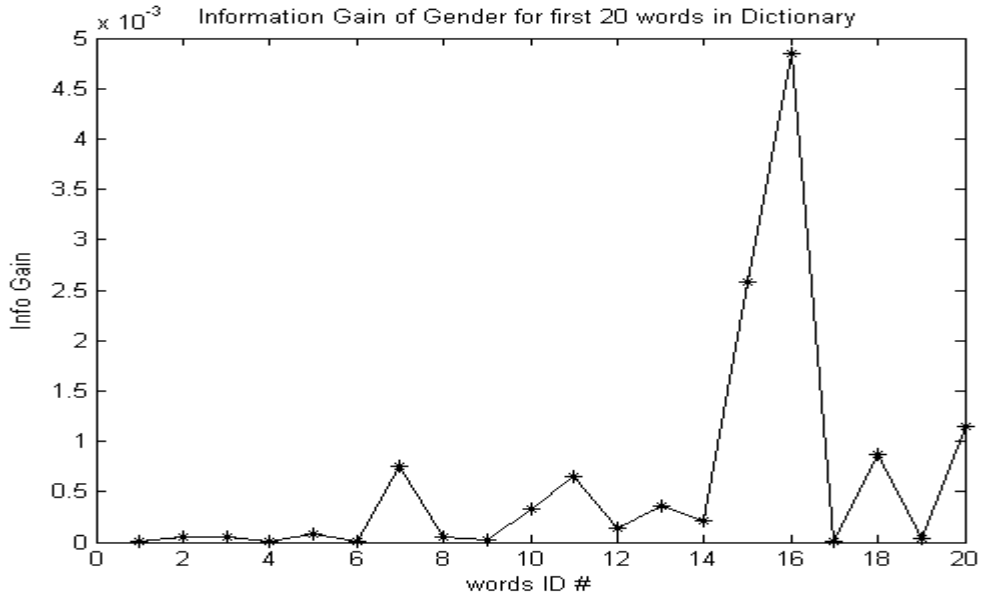

Figure 2: Plot of Information Gain for gender vs. the first 20 words

We also tried the *stemmer* method. Here we try to only take into consideration the stems of the words in the blog. For example if someone is using the words “shopping”, “shop”, “shopped” then we should not have 3 different features for this. Instead we should have just one feature for all these words and it should be “shop”. We used the free Matlab implementation of stemmer [1]. Unfortunately this method did not work well for us because of the fact that most blogs had *lots* of spelling errors. Unfortunately the blogs were not sanitized for spelling mistakes so we did not get the desired advantage from this method. Also, some of the words were foreign words and the stemmer failed to give the stem of those to us. But this approach looks very promising and a better spell check is definitely going to help us a lot.

Another approach we tried was using the SVD (*Singular Value Decomposition*) method. It is a very powerful method to reduce the dimensionality of the dataset. We actually used this in conjunction with LSA (*Latent Semantic Analysis*) method [2]. LSA tries to cluster the similar documents together. It used the SVD method to do so. This looked very promising to us, but it did not work out as well as the Information gain method. The reason behind it is that it tries to cluster the documents together, but it does not take into account the labels that they have. Dimensionality reduction is a good idea, but not at the expense of loosing good features which best predicts the label.

We also tried to use the sparse logistic regression method. [3]. Using this makes immense sense as the L1 logistic regression gives us the sparse matrix of weights for regression. This is in sync with the fact that some features are useful, while some are not. However, even after using lots of tricks the L1 logistic regression method failed to converge on our training data. The number of features (we used uniqueness, to throw away repeated words) was around 50K and the method failed to provide any results to us.

Our final accuracy using the Information Gain method is 71.6 % for Age prediction and 74.2 % for Gender prediction.

An example of Instance Based Method we used is Latent Semantic Analysis.
An example of discriminative method we used is L1 Logistic Regression.

Images:

The images dataset had 600 images with 4 images per person. Our training data set had 488 images and the rest were set aside as test data. The task was to predict the gender of the people and the age group in which they fall in.

Our Approach: We tried lots of learning methods for this task. The method which ultimately helped us in beating the baseline and securing an accuracy of 82.1% for gender and 51.4% for age was relatively very simple. We just used a different cropping of images for this, and we used the SVM implementation provided to us. In order to be confident about our results, we used to set aside some images for development set, but we populated that development set very carefully. Since we knew, we would be tested on images we would not have seen before, so our development set consisted of images which were not in the training set. We used the name field in the data to do so.

We felt that the cropping of images was something that was done in a very arbitrary fashion, because of which we lost a lot of useful information, especially for Gender (consider cropping the hair in the image) and then we would lose some very important information. So we cropped the images differently and we got the nice results using the svm code already given to us. At all the steps we reduced the dimensionality of the dataset using both PCA and SVD. In SVD we did a 100-approximation of the feature matrix. The SVM implementation we used was with a radial basis kernel function, and the slack parameter was found by 6-fold cross validation. We also used *virtual features* in our implementation. We flipped the images on the x axis, so that if a person is looking at the left, we also have in our training set the same person who is looking at the right. This really helped us, as our accuracy improved significantly. Also, we used different

resizing of the image than in the code given to us. The cross validation results told us that the best resizing for gender should be 10*10 and for age it should have been 25*28.

We also tried the L1 logistic regression for predicting age from the images, but the results were not as good, as using SVM. We feel that sparse logistic regression performs best if we know that some features are not useful. Here, we think that after cropping the images, we are not leaving that much scope for L1 logistic regression to perform well. This in fact just reiterates the fact that SVM with a proper kernel function are really useful tools.

We also tried the same information gain concept, to throw away less important features in images. However this did not perform as good as our original algorithm. We feel the reason behind this is that, we do not have lots of features for images as compared to the blog case (we have 83K features for blogs) and throwing away features here was not a very good idea. However there was not a marked degradation in the accuracy after using this approach.

An example of discriminative method we used is L1 Logistic Regression
An example of dimensionality reduction method we used is PCA (or SVD).

We would like to thank Prof. Benjamin Taskar, David Weiss and Jennifer Gillenwater for helping us a lot during this endeavor.

References:

1) Stemmer implementation in Matlab (http://tartarus.org/~martin/PorterStemmer/)
2) Latent Semantic Analysis (http://en.wikipedia.org/wiki/Latent_semantic_analysis)
3) Matlab implementation of L1 Logistic Regression (http://alliance.seas.upenn.edu/~cis520/fall09/demo_implementations.zip)