CIS520 Final Project Report

Fan Deng
Department of ESE
University of Pennsylvania
Philadelphia, PA 19104
fandeng@seas.upenn.edu

Libin Sun
Department of Computer Science
Swarthmore College
Swarthmore, PA 19081
lsun1@swarthmore.edu

1 Introduction

The task of gender and age classification based on texts and facial images has been studied extensively in the natural language processing community and the computer vision community respectively. Machine learning algorithms are particularly relevant in solving such classification problems. In the final project for *CIS520 Machine Learning*, we are presented with the task of gender and age classification in (1) blog postings, and (2) facial images. Our group conducted research on current literature and implemented prediction algorithms for the above tasks, using techniques such as *tf-idf, kNN, eigenfaces and SVM*. We achieved decent classification results and successfully beat all 4 baseline requirements. This report will describe in detail our methods and findings.

2 Blogs

The training data given for this task contains 1700 blog entries each with approximately 1000 words. The original data also comes with information for age and gender of the bloggers. The task is to train a model which can predict age and gender for a blogger given some unseen blog entry.

2.1 Steps of Text Categorization

Basically, the architecture of text categorization consists of following steps[9,8,13]:

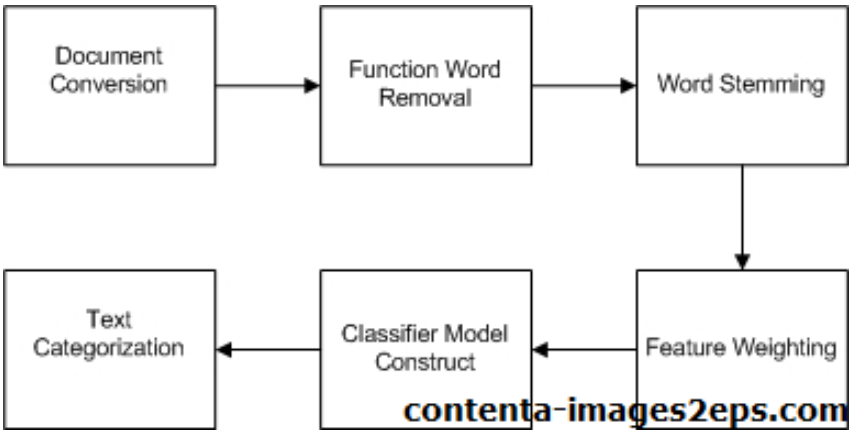

Figure 1: Overview of Text Categorization

1. Factors in Data Preprocessing[15,16]

   (a) Function Words(Stop Words) removal
   Where a list of function words which are topic neutral such as *a, an, and, I, in, at, the*
   In our project, this step is skipped as we noticed removing stop words did not boost performance. In some cases, the estimated test accuracy actually suffered from ignoring stop words [15].

   (b) Standardize words suffixs (Stemming)
   (e.g. introduction $\rightarrow$ introduce, running $\rightarrow$ run)
   In our project, this step is skipped since the original data has been transformed hence can not be processed with low cost. Additionally, the improvement in performance is not worth the computational overhead [15].

   (c) Feature selection
   Reducing feature dimensionality by removing less relevant words.
   In our project, we are using the whole dictionary as feature space without removing any entries. We noticed by both (1) Setting threshold and selecting features using information gain [12,15], (2) Setting threshold and selecting features using total frequency control the performance of the classifiers became worthwhile and calculation cost was also reduced. Since calculating efficiency is not our concern, we use the whole dictionary as feature space.

   (d) Feature Representation and Weighting
   Feature Representation and weighting give each word different weights and represent each training example with a combination of weights and features. In the starter-kit, the baseline method uses *Document Frequency* as feature expression and assigns each word identical weight, which is 1.
   We used the following feature expression and weighting schemes:
   TF-IDF(Term Frequency and Inverse Document Frequency with Normarlization):[10]

$$\mathbf{tf_v} = (tf_{v1}, tf_{v2}, ..tf_{vn}), w_i = \log \frac{N}{df_i}, \mathbf{w} = (w_1, w_2, ..w_n)$$

   tf is the term frequency in document v, N is total number of documents and df_i is the document frequency of ith term
   Hence document feature vector is defined as :

$$\mathbf{f_v} = \frac{\mathbf{tf_v} * \mathbf{w}}{||\mathbf{tf_v} * \mathbf{w}||_2}$$

   The words with most weight are:
   puuuurtty SOLD RACKFUL luuuuuuuurve urgly luuuuuurve XDD anyhooooooo thaaaank maaaaaan luuuuuuurve piccy ubercool aaaaaawesome TUBULAR riiiiiiiight meng wahooooooEE frogger
   logRelFreqL2Imp (logarithmic frequencies with redundancy normalized with respect to L2)[16]:

$$r_k = \log N + \sum_i^N \frac{f(w_k, d_i)}{f(w_k)} \log \frac{f(w_k, d_i)}{f(w_k)}, \mathbf{r} = (r_1, r_2, .., r_n)$$

$$\mathbf{l_i} = (\log(1 + f(w_1, d_i)), .., \log(1 + f(w_k, d_i)))$$

   r_k is the importance weight of term k and $\mathbf{l_i}$ is logarithmic frequency of document i
   Hence feature vector of document i can be defined as :

$$\mathbf{x_i} = \frac{\mathbf{l_i} * \mathbf{r}}{||\mathbf{l_i} * \mathbf{r}||_2}$$

   RelFreqL2Imp(Raw frequencies with redundancy normalized with respect to L2)[16]:
   Similar to **LogRelFreqL2Imp**, but we are using Raw term frequencies instead of logarithmic frequencies

$$\mathbf{f_i} = (f(w_1, d_i), .., f(w_n, d_i))$$

Feature vector of document i can be defined as:

$$\mathbf{x_i} = \frac{\mathbf{f_i} * \mathbf{r}}{||\mathbf{f_i} * \mathbf{r}||_2}$$

The words with most weight in this case are the same with previous one

2. Classification Methods[Dumais]
   SVM (Support Vector Machine) Using linear kernel since feature dimensionality is much larger than training size,the training set can definitely be linearly separated[11]
   Decision Tree-SVM Using Decision Tree -SVM combined method to deal with multi-class classification problem [14]
   KNN(K-nearest Neighbor) In our project, k is set to 25

2.2 Results

Table 1: Best Test Accuracy for Difference Feature Expression

	TF-IDF	logRelFreqL2Imp	RelFreqL2Imp
GENDER	75.3	74.5	74.2
AGE	70.9		68.5

Notice that logRelFreqL2Imp and RelFreqL2Imp weighting method gives more weight to some strange words or symbols,this may imply that why logRelFreqL2Imp and RelFreqL2Imp Feature Expression methods perform worse,after words refinement logRelFreqL2Imp and RelFreqL2Imp should theocratically perform better.

Table 2: Best Test Accuracy for Difference Classify Methods

	SVM	KNN	Decision-SVM
GENDER	75.3		
AGE	70.9	61.5	70.9

3 Facial Images

The training data given for this task consists of approximately 500 facial images of 120 people, namely, mulitple images for an individual are given, with varying lighting conditions, angles and backgrounds. The task is to predict gender ('male' or 'female') and age, which is divided into 3 age groups: 30, 35-50, 55. Our algorithm will be tested on a large test set of approximately $2/3$ the size of the training set, with roughly 4 images per person. It is assumed that no individual appears in both the training set and the test set. Our method will be described in the following sections.

3.1 Related Work

A highly related area research is face recognition, which has been studied extensively and offers a plethora of techniques. [1,2] presented a novel approach using eigenfaces for face detection, in which PCA is carried out to capture the variations among all the mean-subtracted faces. A face is then represented by mapping the mean-subtracted image into this new eigen space. [3] compared eigenfaces to fisherfaces for the task of face recognition and argued that fisherfaces delivered better accuracy over the Harvard and Yale datasets. [4] invented 'integral image' and discovered the efficiency of 'Haar-like' features for robust face recognition. In addition, their concept of a cascade of classifiers is also a major contribution. These ideas serve as platform for further investigation in facial images, such as gender and age classification. [6] applied eigenfaces to classify gender, age and ethnicity based on facial images and carried out analysis on the discriminative power of individual

components in the eigen space. In addition, many other approaches have also been developed for the same task, such as using Gabor features, face templates for localizing nose and eyes [5,7].

3.2 Methods

General Framework The starter kit supplied on the project webpage provides an efficient framework to work on the problem by training an SVM classifier and carrying out predictions. The main workflow can be described as follows:

1. load data, randomly split into training set and test set
2. feature extraction on training instances
3. perform cross-validation on training set to choose cost variable C
4. train and save SVM using the chosen cost variable
5. make predictions on the test set and analyze errors

In the above steps, (c) through (e) can be assumed to work well given a good set of features. In (a), a random split on the dataset is not optimal because we know that the final test set (used by the TA's) contains facial images from unknown individuals. As a result, a random split on the dataset will cause a high correlation between our own training set and test set, due to the presence of facial images of the same individuals in both sets. Hence, the classifier trained using a random split will perform well over the dataset we have, but face potential fiasco when making predictions on the final test set consisting of entirely unseen individuals. To be consistent with the final test set, we split the data based on names into approximately 300 training images and 188 test images such that no individual appears in both sets. In (b), feature extraction givens a representation of the image in a feature space. The baseline simply stacks all the pixels into a column vector to represent the image. This naive approach can be used for simpler task such as recognizing hand-written digits, but performs rather poorly when analyzing complex patterns such as faces. We examine possible features in the following discussions.

1. Low Level Features
   Image processing offers many techniques for content-based image analysis. It is natural to consider pixel-based low level features first. We considered color histograms in RGB and HSV color spaces, wavelet features (Haar wavelet) and Gabor features. Color spaces can be used to capture things such as make-up; wavelet features are expected to capture texture information such as wrinkles and hair. Color histograms obtained using n bins can be seen as an $n-$dimensional vector. Haar wavelet and Gabor transform both return a matrix of the same size as the input image, which can then be reshaped into a column vector. A final feature vector is then constructed by stacking all the features together. However, the features must first be standardized using the *Z-scoring* method below to account for the difference in variance:
   $$x_i^{'} = \frac{x_i - \textbf{Mean}(X_i)}{\textbf{Std}(X_i)}$$
   Due to time constraint, we did not test this method in a spatial pyramid setting. Instead, only global information is used. However, the spatial pyramid approach is unlike to deliver superior accuracy because the facial images contain many non-frontal views with varying lighting conditions, unlike the 'clean' images in the Harvard and Yale datasets used by many researchers. Such variations in orientation and lighting cannot be remedied by using frames or spatial pyramids.
2. Eigenfaces
   Eigenfaces are easy to implement and efficient for face recognition. It is argued that certain components in the eigenfaces contribute to different variations and thus carry the potential to distinguish gender and age [6]. The standard algorithm is given in [1,2], which we shall not belabor in this report. We examine several approaches:
   (a) Standard eigenface algorithm: choose k eigenvectors with the largest eigenvalues as basis. $k = 150$ gives consistently good performance.

(b) Gender-specific eigenfaces: compute eigenfaces using only male or female facial images so that 2 different sets of eigenspace can be formed. It is hoped that this formulation will pick up the subtleties for gender classification.

(c) Age-specific eigenfaces: compute eigenfaces using images from each age group, forming 3 different sets of eigenspace. It is hoped that this approach will be efficient for age classification.

(d) Age-gender-specific eigenfaces: a total of 6 eigenspaces is constructed for all 6 combinations of $\{\text{gender} \times \text{age}\}$.

3. Bootstrapping
   Since the given dataset is limited in terms of possible variations over lighting conditions and orientation, we wish to make the prediction invariant to such differences. The common practice is to bootstrap the training set by deliberately adding transformed instances to account for such variantions not intrinsic in the original training set. We consider 2 main transformations: (a) mirror image, (b) rotation. The original training set can be bootstrapped by adding (1) mirror images of all images, (2) images rotated by $\pm k$ degrees, (3) mirror images with rotation.

3.3 Experimental Setup

The training data given for this task consists of approximately 120 people, 4 images per person. The people are divided evenly among the age groups and genders: 50 people per age group, 25 of each gender in each age group. The test dataset consists of images of another 30 people, 4 images per person, also divided evenly among the classes. Images are of size 250x250x3 (third dimension is red-green-blue coloring). Before constructing eigenfaces, we first crop the images, only keeping the center portion containing the face with minimal hair, and resize that to 45x45 pixels. *imadjust* and *histeq* functions are used to normalize the intensities.
The dataset given to us is then split into a training set and a test set such that no individual person appears in both sets. This is consistent with the final test set our classifier will be tested against. In our tests, we keep the test set size around 180. This gives us 488-180=308 training images. However, the images often contain people in a non-front view, with inconsistent lighting and pose. Thus, we establish a blacklist to keep track of images where the cropped section contains non-face objects such as hand and microphone, or if the section does not fully capture the face, e.g. missing an eye. This list is subtracted from the training set, but retained if in the test set.
We tested the efficiency of using various number of eigenface components with the largest eigenvalue, and compared test accuracy among the methods discussed in the previous section.

3.4 Results

1. Low Level Features vs Standard Eigenface
   We first tested the efficiency of low level features against the standard eigenface approach. Here, low level features contain 10 bins for each component in RGB and HSV, along with 45x45 values after Haar wavelet transform, yielding a 2085 dimensional-vector for each image. For eigenface, we used the first 150 principal components, thus representing each image by a 150 dimensional vector. Table 1 shows the average test accuracy obtained on 5 runs using 180 test images each. As discussed, the test set consists of facial images from unseen individuals. The training set data does not contain any of the virtual examples discussed under *bootstrapping*. Clearly, eigenfaces offer much better performance for both gender and age classification.

Table 3: average test accuracy for feature comparison.

	RGB+HSV	Haar wavelets	RGB+HSV+Haar	eigenface
gender	57.45	65.83	70.38	77.42
age	29.71	31.55	37.49	45.35

2. Number of Eigenfaces as Basis
   In PCA, we often try to reduce dimenality by greedily choosing features to account for the largest variation in the data. In the case of eigenfaces, we do so by controlling the number of eigenfaces we keep to establish the new eigen space. Figure 1 and 2 illustrate how a reconstructed face looks like using various number of principal components. We see that approximately 100 eigenfaces will be sufficient for reconstructing the face of a stranger.

Figure 2: Face of an unseen person vs reconstructed faces using the first 5, 15, 25, 50 and 100 eigenfaces. Trained using 360 facial images plus their mirror images.

3. Variation of Eigenface
   As discussed in the previous section, we formulate several variations of eigenface. We conducted tests on these variations by training a separate SVM classifier for each variation, given the same split of dataset. This ensures that the test accuracy is consistent and reflects the true efficiency of each method. The results are shown in table 2 and 3. The gender-specific is expected to differentiate male and female better, but apparently by capturing only the variations among male faces, it is not guaranteed that such variations are absent in female. Same reasoning applies to the age-specific approach. These approaches considers only part of the data thus ignores useful variations in all the faces. Overall, the results show that by taking into account the variations in all images, better performance can be achieved for both gender and age classification.

Table 4: gender accuracy for variations of eigenface method

run #	1	2	3	4	avg
standard	78.32	76.84	75.49	79.57	77.55
gender-specific	74.53	75.21	72.06	75.96	74.44
gender+age specific	75.65	72.91	71.89	77.37	74.46

4. Power of Bootstrapping
   Last but not the least, we examine the power of bootstrapping. Let M be *adding all mirror images into the training set*, $R1$ be *adding clockwise rotation of 8 degrees for all images*, $R2$ be *adding anti-clockwise rotation of 8 degrees for all images*, $M + R1$ be *adding both*

Figure 3: Face of an unseen person vs reconstructed faces using the first 5, 15, 25, 50 and 100 eigenfaces. Trained using 260 facial images plus their mirror images.

transformations to bootstrap the dataset, $M * R1$ be *adding images obtained from flipping AND rotating into the dataset*. Table 4 shows the average test accuracy of various bootstrapping outcomes over 5 runs. Evidently, expanding the training set through bootstrapping can be an efficient way to boost test accuracy. However, as we add virtual examples to the training set, we are making the classification problem harder (more variations to account for) and also suffers from overfitting. In the final submission, due to space limitation, we choose to use only mirror images to expand the training set, which gives a good trade-off between accuracy and memory requirements.

4 Conclusions

In text categorization, data representation methods selection is vital to the final performance of classification. Without concern of calculation cost,it seems a relatively raw feature selection would be a good choice. TF-IDF seems to be a simple and effective feature representation, compare to

Table 5: age accuracy for variations of eigenface method

run #	1	2	3	4	avg
standard	45.83	49.31	47.62	43.79	46.64
age-specific	41.55	46.32	43.91	39.86	42.91
gender+age specific	44.21	47.39	42.48	41.75	43.96

Table 6: average test accuracy using bootstrapping

	M	M+R1+R2	M+R1+R2+M*R1+M*R2
gender	81.64	82.51	80.47
age	52.29	52.68	51.07

baseline method, it takes frequency and text length into consideration. SVM with linear kernel works pretty good since almost all text data are linear separable. Knn Classifier performs much worth than expected, also it consumes a lot of resources during classifying process. The reason why knn performs so bad might be TF-IDF in this case is not a good method. Feature dimensionality seems too large for a single document where most of features of a particular sample have value of 0. Eigenfaces provide a efficient and simple framework for analysis of facial images. In particular, decent accuracy can be achieved for gender classification by implementing a standard eigenface approach (80%). However, age classification requires further tweaking or even an entirely new approach.

Acknowledgments

We would like to thank Professor Ben Taskar, TA David Weiss and Jenny Gillenwater for their guidance and help.

References

[1] M. Turk and A. Pentland, "Eigenfaces for Recognition", *Journal of Cognitive Neural-science*, vol. 3, no. 1, March 1991.

[2] M. Turk and A. Pentland, "Face Recognition Using Eigenfaces", *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1991, pp. 586-591.

[3] P. N. Belhumeur, J. P. Hespanha and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, July 1997.

[4] P. Viola and M. Jones, "Robust Real-time Object Detection", *Second International Workshop on Statistical and Computational Theories of Vision*, July 2001.

[5] Y. H. Kwon and N. da Vitoria Loboy, "Age Classification from Facial Images", *Computer Vision and Image Understanding* Vol. 74, No. 1, April 1999, pp. 1-21.

[6] S. Buchala, N. Davey, T. M. Gale and R. J. Frank, "Principal Component Analysis of Gender, Ethnicity, Age, and Identity of Face Images", *IEEE ICMI*, 2005.

[7] H. Takimoto, Y. Mitsukura, M. Fukumi, N. Akamatsu, "A Design of Gender and Age Estimation System Based on Facial Knowledge", *SICE-ICASE International Joint Conference*, 2006

[8] Susan Dumais,John Platt "Inductive Learning Algorithms and Representations for Text Categorization" .

[9] Gonde Guo,Hui Wang,David Bell "An kNN Model-based Approach and Its Application in Text Categorization" 2004.

[10] Monica Rogati, Yiming Yang "High-Performing Feature Selection for Text Classification".

[11] A.Basu,C.Watters,M.Shepherd "Support Vector Machines for Text Categorization".

[12] Rahman Mukras, Nirmalie Wiratunga,Robert Lothian,Sutanu Chakraborti,David Harper "Information Gain Feature Selection for Ordinal Text Classification using Probability Re-distribution".

[13] Ciya Liao,Shamim Alpha,Paul Dixon "Feature Preparation in Text Categorization".

[14] Gjorgji Madzarov, Dejan Gjorgjevikj, Ivan chorbev "A Multi-class SVM Classifier Utilizing Binary Decision Tree" 2008.

[15] Fengxi Song,Shuhai Liu,Jingyu Yang "A comparative study on text representation schemes in text categorization" ,*Pattern Anal Applic(2005)8:199-209*,2005.

[16] Edda Leopold,Jorg Kindermann"Text Categorization with Support Vector Machines How to Represent Texts in Input Space?"*Machine Learning,46,423-444,2002*,2002.