CIS 520 Project Final Report: Twitter Gender Classification

Woodpecker (Xiang Deng, Yiren Lu, Dongni Wang)

Fall 2015

1 Project Overview

For the final project, we developed a system for twitter usersś gender prediction (male/female) from the language of their tweets and their profile images. We were given a training set of 4998 labeled training samples, each has 5000 word features, 7 pre-extracted image features and 30000 raw RGB image pixel features.The time constraint for the final model(s) initialization and prediction is 3 minutes and 10 minutes, respectively, for the 5,000 test samples. Also, The submission size is limited to 50 Mb in the final checkpoint/ competition. Our submitted model for the final competition achieved an accuracy of 91.04% on the validation set, ranked 6^{th} in a total of 50 2-3 people teams.

In our full system, we used seven classifiers on different feature sets and combined them using the stacking method. The seven classifier are: a logistic regression model on words features, an ensemble model consists of 300 decision stump trees using LogitBoost on selected words and image features, a SVM model with intersection kernel on selected words and image features, a SVM model with intersection kernel on selected and normalized words and image features, an ANN model with 2 hidden layers each with 100 and 50 nodes, a SVM model with RBF kernel on PCA-ed HOG features on face-detected images, and a SVM model with RBF kernel on PCA-ed LBP features on face-detected images. For the stacking method, we took the raw outputs (probabilities) from the seven basic models mentioned above trained with 80% of training samples and trained a logistic regression model using the other 20% of training sample. Our final full model achieved an overall accuracy of 92.42% on testing set.

In order to meet the time and space constraint for the competition, we dropped the SVM model on PCA-ed LBP features and replaced the SVM model on PCA-ed HOG features with one bagging of logistic regression classifiers on raw HOG features.

In the following sections, we present the cross-validation accuracies of each method we tried and discuss the rationale of our final model. We also provide some interesting visualization such as the most predictive words and eigenfaces..

2 Methods

In this section, we report the results of multiple methods we tried for feature extraction, dimension reduction, and classification.

2.1 Data Preprocessing

2.1.1 Stemming and Stop Words

Conclusion: Did not work!

2.2 Feature Selection

To extract features from the raw word and image features, we experimented with multiple feature selection methods, including Information Gain, BNS

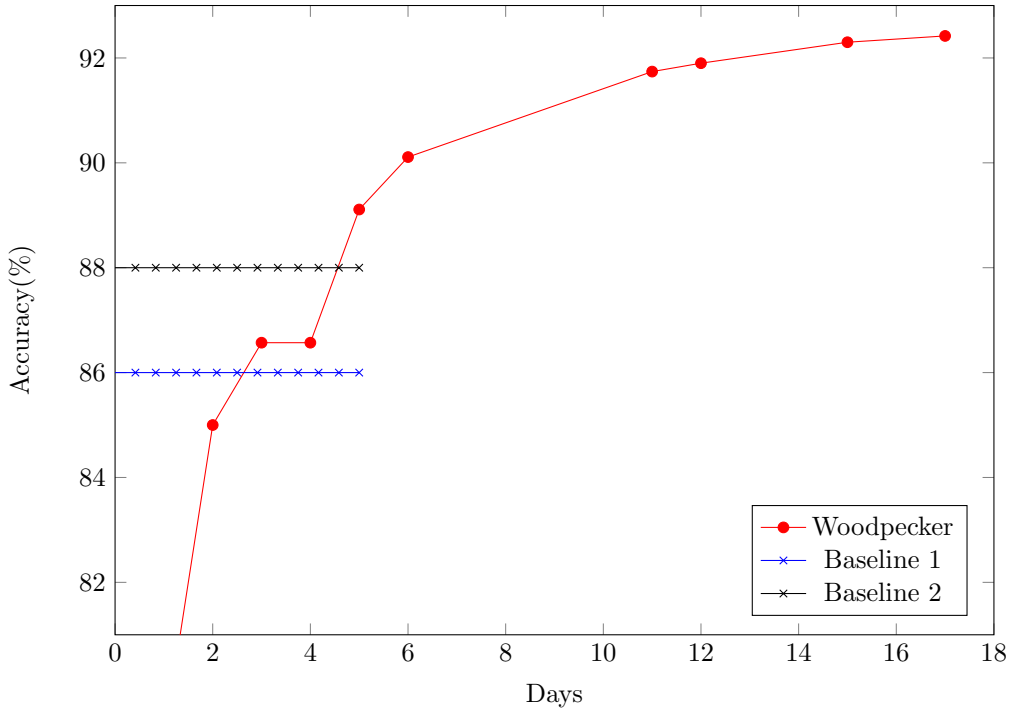


Figure 1: The progress plot for Woodpecker

2.2.1 Information Gain Over Words Features

words visualization

2.3 Dimension Reduction

2.3.1 PCA

2.4 Classification on Words Features and Extracted Image Features

2.4.1 Linear Regression (L2 Regularization) + Sigmoid

2.4.2 Logistic Regression + PCA

2.4.3 Logistic Regression on Raw Features

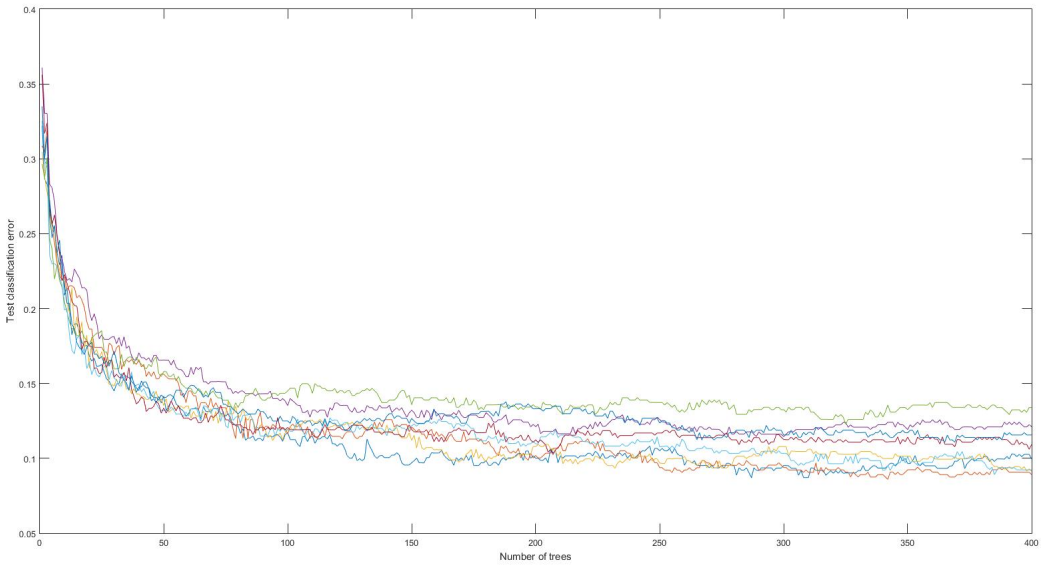
2.4.4 Naive Bayes

2.4.5 Neural Network

2.4.6 LogitBoost and Feature Selection Using Information Gain

We want to use both words features and image features in order to enhance our accuracy, but we don't know which feature is most informative; therefore, we ranks all words and image features together using information gain and select the top features (In fact, by computing the information gain of image features, we found image feature 1 2 5 6 and 7 have roughly 0 information gain, but of course this does not mean all image features are useless).
On the other hand, Logitboost can be seen as a convex optimization which combines Adaboost with the cost function of logistic regression. We experiments with Logitboost based on the following considerations.
1. Words and the seven image features have different scales, we need a model that is scale invariant; boosting with decision stump trees in this case is a great choice.

2. The cost function using logistic regression is suitable for binary classification, it is a convex problem and minimizes the binomial deviance and gives less weights to misclassified observations.
By selecting the top 1000 features ranked by information gain and using 330 decision stump trees, we achieved an accuracy of 89.11 percent on the testing set.
In our final submission, this model also serves as a major classifier that contributes to our ensemble method.
Below is the cross validation accuracy plots (over 8 folds):



2.4.7 Kernel SVM

2.5 Classification on Image Features

2.5.1 Grey-scale Images

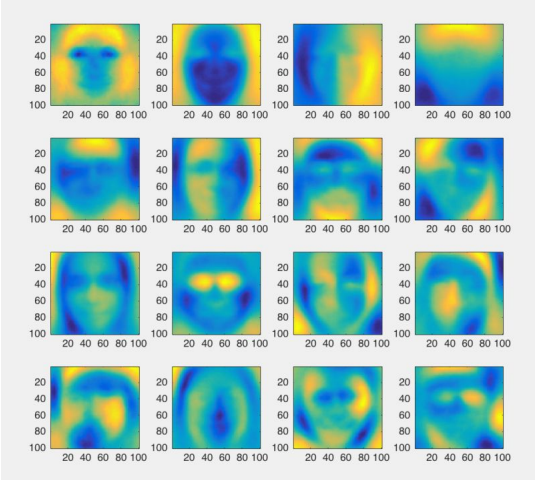
We first converted the profile images data into R, G, B and Grey-scale images.

2.5.2 Face Detection with Viola-Jones Algorithm

We extracted all the faces. We used matlab built-in Viola-Jones algorithm to do face detection on RGB images, which detected faces on 72% of the profile pictures. All the following classifications over images were based on the faces detected. Later when ensemble, we only used samples with detected faces. For those samples without faces in them, we just set the output of the classifier as 0 (the output of the classifier is either positive or negative for male or female).

2.5.3 Logistic Regression with Eigen Faces

After extracted faces, we did PCA over grey-scale images, the visualization of the top principal components are shown in figure below. We put top PCs into logistic regression classifier and it yielded 60+% accuracy.



2.5.4 HOG Features, Gaussian Pyramid, Eyes and Nose

We used Viola-Jones algorithm to detect and crop eyes and nose. After that, we extracted Gaussian Pyramid HOG features into 7020 feature vector. Classifying the features with logistic regression yielded 82% accuracy by itself on 72% detected faces. The ensemble with this classifier produced 91.9% over all accuracy on the test set.

2.5.5 SVM on PCA-ed HOG Features

We then replaced the logistic regression classifier with a more powerful RBF kernel SVM. The accuracy on detected faces was improved up to 83%. The ensemble with this classifier got 92.3% accuracy on the test set. Then we did PCA (1500 PCs) on HOG features to reduce the dimensionality and the size of SVM model. The accuracy of the single model with improved to 84%. However, the ensemble with this classifier dropped to 92.14% accuracy on the test set.

2.5.6 SVM on PCA-ed Dense LBP Features

LBP works complementary with HOG features, we extracted spatial pyramid LBP features (15871 features) on detected faces. And trained a SVM classifier over 2000 principal components of the LBP features. This model achieved 85% accuracy on detected faces. We integrated this model to the ensemble yielded our highest accuracy on the leaderboard of 92.42% on test set.

2.5.7 Bagging Logistic Regression on Raw HOG Features

To meet the time and space constraints of the competition, we dropped LBP model and and replaced the SVM model with bagging logistic regression. Because the basis of PCs took 130m, which was too large for 50m space constraint, we didn't do PCA in the final submission either. Instead we used a bagging logistic regression with 6 logistic regression models. The final model produced 91.04% accuracy on validation set.

2.5.8 Auto-encoder

2.6 Ensemble Methods

2.6.1 Stacking

We used stacking to ensemble our models to achieve better over all performance than any of the single models. First, we used 80% of the training data to train all the single models including logistic regression, neural network, logitboost with feature selection, kernel SVM with feature selection, kernel SVM with normalization, SVM over PCA-ed HOG, SVM over PCA-ed LBP. And then, use those models to generate predictions (scores) for the rest 20% data. Training a logistic regression model using the 20% scores with

labels yields a ensemble classifier. Finally, we use all the training data to re-train all the single models. Along with the ensemble classifier, we got our final model. Our full model yielded 92.42% prediction accuracy on the leaderboard over test set. For the final submission, due to the space and time constraints, we dropped the LBP model and replaced the SVM over PCA-ed HOG with bagging logistic regression over raw HOG features. This final submission yielded 91.04% prediction accuracy on validation set, ranked 6th out of 50 teams.

2.6.2 Normalization

When ensemble all the models, we notices that normalization actually works. For the images, we first trained logistic regression on raw gaussian pyramid HOG features (faces, eyes and nose), which yielded 82% accuracy on detected faces by itself. Later, we figured SVM works better by experiment (84%). However, when we ensemble the new model, it actually produced lower accuracy. We then found that the scores ranges produced by logistic regression and SVM were actually different. This may cause unbalanced weights across models. We then normalized all the scores with a sigmoid function with mean 0 and variance 2. This produced higher overall accuracy (92.3%). Tweaking the sigmoid function parameters for each model may produce a little bit higher accuracy but we didn't get enough time for that.

2.6.3 Cascaded Ensembling

3 Experiment Analysis

In this section, we analyze the results of our experiments of multiple methods for feature extraction, dimension reduction, and classification.

3.1 Feature Selection/Extraction

3.2 Dimension Reduction

3.3 Classification

The table of approaches and their associated 5-fold cross-validation classification accuracies are shown in the Table 1

4 Discussion

Working on this gender-classification project gave our team a chance reflected on what we have learned in class. Here is a short summary of things that have surprise us (or have taught us a lesson)

- With different feature sets (especially they have various ranges and dimensions), feature selection and normalization have played an important role in improving the performances of our model.
- Last but not least, be careful with required formats..

Feature	Approach		Accuracy (%)
	Dimension Reduction	Classifier	
Words + Image features	PCA(500)	Ridge Regression + Sigmoid	≈ 70%
Words + Image features	PCA(320)	Ridge Regression + Sigmoid	≈ 79%
Words + Image features	PCA(2000)	Logistic Regression	≈ 85%
Words	None	Logistic Regression	85.96%
Words	IG(1000)*	Logistic Regression	85.79%
Normalized-Words	None	Naive Bayes	72.25%
Words	IG(100)	multinomial Naive Bayes	77.95%
Words	None	Bernoulli Naive Bayes	79.59%
Words	IG(350)	Bernoulli Naive Bayes + EM	82.49%
Words	PCA(2000)	Artificial Neural Network	≈ 86%
Words	IG(76)	K-Nearest Neighbor (L2)	72.89%
Words	IG(84)	K-Nearest Neighbor (Minkowski)	71.43%
Words	IG(95)	Random Forest	83.32%
Words	None	K-means	≈ 60%
Words	IG(1000)	Decision stumps + LogitBoost	89.11%
Face-detected Image RGB	PCA(100)	Random Forest	≈ 69%
Face-detected Image RGB	Auto-encoder(100)	Logistic Regression	75.17%
Raw HOG features over Face-detected Image RGB	None	Logistic Regression	≈ 80%
Raw HOG features (Face/eyes/ nose-detected Image RGB)	None	Logistic Regression	≈ 81%
Raw HOG features (Face/eyes /nose-detected Image RGB) + Gaussian Pyramid	None	Logistic Regression	≈ 82%
Row HOG features (Face/eyes /nose-detected Image RGB) + Gaussian Pyramid	None	SVM (RBF kernel)	≈ 83%
HOG features (Face/eyes /nose-detected Image RGB) + Gaussian Pyramid PCA(1500)	None	SVM (RBF kernel)	≈ 84%
Dense LBP (Face-detected Image RGB)	None	SVM (RBF kernel)	≈ 85%
*IG represents Information Gain			

Table 1: Experimental results of single classifiers

Approach		
Preprocessing	Classifier	Accuracy (%)
Features: Words + Image features		
IG(1000) on words and image features	Logistic (W) + Neural Network (W) + Ensemble trees (W+I) + stacking	91.11%
IG(1000) on words and image features	Logistic (W) + Neural Network (W) + Ensemble trees (W+I) + cascading	89.04%
Features: Words + Image features + Image RGB		
IG(1000) on words and image features; Face-detected image RGB PCA(100)	Logistic (W) + Neural Network (W) + Ensemble trees (W+I) + Logistic (PCA-ed RGB) + stacking	90.37%
IG(1000) on words and image features; Face-detected image HOG features	Logistic (W) + Neural Network (W) + Ensemble trees (W+I) Logistic (HOG) + stacking	91.55%
IG(1000) on words and image features; Face/eyes/nose-detected image HOG features	Logistic (W) + Neural Network (W) + Ensemble trees (W+I)+ Logistic (HOG) + stacking	91.74%
IG(1000) on words and image features; HOG features (Face/eyes /nose-detected Image RGB) Gaussian Pyramid	Logistic (W) + Neural Network (W) + Ensemble trees (W+I) Logistic (HOG) + stacking	91.9%
IG(1000) on words and image features; HOG features (Face/eyes /nose-detected Image RGB) Gaussian Pyramid + PCA(1500) Normalization	Logistic (W) + Neural Network (W) + Ensemble trees (W+I) Logistic (HOG) + stacking (SVM)	92.3%
IG(1000) on words and image features; HOG features (Face/eyes /nose-detected Image RGB) Gaussian Pyramid + Dense LBP + Normalization	Logistic (W) + Neural Network (W) + Ensemble trees (W+I) Logistic (HOG) + stacking (SVM) PCA(1500)	92.42%
*For the classifiers: W: words; I: Image features		
*For the stacking method, we use logistic regression when not otherwise specified		

Table 2: Experimental results of ensemble classifiers