CS 613: Machine Learning

Group Competition Report

Ronak Nathani, Abhilash Iyer and Sundar Ram

Problem Statement

The given problem is a sentiment classification task, to classify online user reviews from amazon.com and tripadvisor.com into 5 classes, identifying the respective star ratings. The goal was to predict labels for test data to achieve as high accuracy as possible above the given baseline using provided vector representations of the 5-gram features.

Datasets

There are 3 datasets and each dataset has been split into 3 groups of training, validating and testing sets, resulting in a total of 9 data splits. All three datasets were obtained from user reviews organized into 6 categories: apparel, automotive, books, dvd, music and hotel reviews (from Trip Advisor). Each dataset contained 5000 validating samples and 10000 testing samples. Dataset 1 & 2 contained 10,900 training samples while dataset 3 had 8,200 training samples. All of these were preprocessed to extract 5-gram features. The labels in training data for all 3 datasets had the same distribution. Moreover, labels in validation and test sets also had identical distribution.

Data Representations

The raw text from the selected reviews were then converted to vector space representations. The following transformations were used to preprocess the raw text : TF-IDF , LSA and PCA. These transformations were estimated in a transductive setting. All reviews from the 9 splits in the 3 datasets(three *.sqw files per dataset) were concatenated into a single file dt_combined.sqw. Parameters of TF-IDF, LSA and PCA were then estimated using all samples in dt_combined.sqw. Please refer to get_tfidf.py for exact implementation of these transformations.

1. Term Frequency - Inverse Document Frequency (TF-IDF)

TF-IDF is the product of two statistics, term frequency and inverse document frequency. It is a numerical statistic that reflects how important a word is to a documents in a collection or corpus (collection of documents). It is often used as a weighting factor in information retrieval and text mining. The tf-idf value increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus, which helps to control for the fact that some words are generally more common than others. The Dataset 1 was provided in tf-idf format.

2. Principal Component Analysis (PCA)

PCA converts the data from high dimensional space to a lower dimensional space. It is a statistical method in which the observed raw data set which might be correlated in nature, is converted into linearly uncorrelated variables. This uncorrelated variables are principle components. The components are ordered in the decreasing order of their variance in this representation. PCA can be computed by mean centering the data and performing an SVD on it and taking the first few columns

(principal components) of the V matrix produced by SVD as the principal components.

$$X = U \Lambda V^T$$

where X is the data matrix, is the diagonal eigenvalue matrix and U and V are unitary matrices.

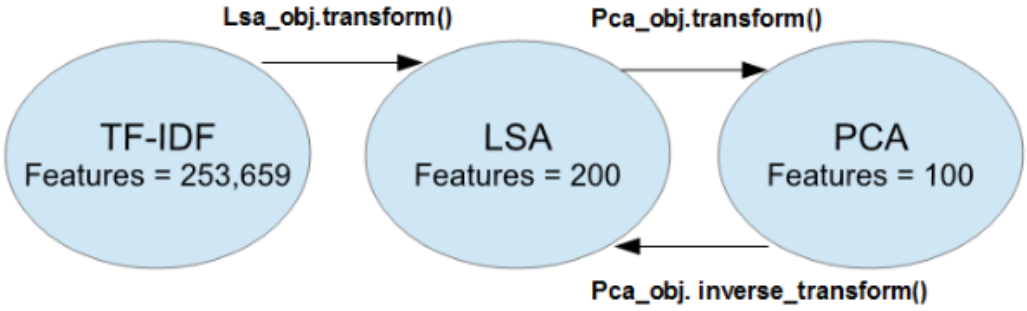
3. Latent Semantic Analysis (LSA)

Assume A be the [mxn] term-review matrix of a collection of reviews. Each column of A corresponds to a review. We can represent A in terms of

$$A = S_k \Sigma_k U_k^T$$

where, S_k is the matrix of eigenvectors of A^TA (review-review matrix) and U is the matrix of eigenvectors of AA^T (term-term matrix). In LSA, we take the k most important singular values and ignore the rest. The 'k' in our case is set to 200. That is why this method is also called truncated SVD. The terms in the new space are represented by the row vectors of the mxk matrix $S_k\sum_k$, whereas the reviews by the column vectors of the kxn matrix, $\sum_k U^T_k$.

Transform/Inverse Transform relations



Text data representations and their Transformations

These conversions led into the increase in the amount of training dataset in each of the representations.

Method used

Our first approach towards increasing the accuracy above the given baseline was to try different classifiers, namely SVM, Gaussian Naive Bayes and Random Forest, on the given vector representations for each dataset with the same training data. But more or less, all the classifiers gave

the same results, with little increment or decrement around the baseline accuracy. The significant improvement in accuracy was achieved after the hints E & F. We learnt from hints E & F about how the parameters were estimated and that the provided transformation objects could be used to invert the LSA and PCA transformations.

After learning from hints E & F, we used the provided PCA and LSA transformation objects to transform/inverse transform all the splits in each dataset, i.e. we had the training, validation and testing splits of all dataset in both 200 and 100 dimensional vector representation. For this purpose, we had to invert the PCA transformation for dataset 3, apply PCA transformation to dataset 2 and apply both LSA and PCA transformations to dataset 1.

The labels of training data for all three sets came from the same distribution, which meant that we could combine the training data of all three sets in a particular dimensional space (100 or 200). We combined the three training data splits in both 100 and 200 dimensional space for every dataset and used this combined training data to train our classifier. Next, we predicted the class labels of validation and test data splits, using this trained classifier, for each dataset using the vector representation in the corresponding dimensional space for validation and test sets as was for training set used to train the classifier. Now, with a large training set, we achieved significant increase in accuracies over the given baseline accuracy using the same classifiers and the same hyperparameters as were provided initially. From the results that we got, we decided to use 200 dimensional vector representation for dataset 1 & 2 and 100 dimensional vector representation for dataset 3. These results were submitted during the first round of final submission.

For the second round of final submission, we tried various classifiers for each dataset and compared results by changing the hyperparameter values. The table below describes all the different classifiers we used and results for the different hyperparameter values for a given classifier. Many of the hyperparameters setting which didn't give good result were not recorded and hence, they don't appear in the list below.

Prediction Table

Dataset	Representation	Accuracy on Validation set	Hyperparameters	Classifier
Dataset1	200 dimensional	55.36	C = 10	LinearSVC
Dataset1	200 dimensional	56.76	Number of trees = 1000	Random Forest
Dataset1	200 dimensional	57.68	C=100, gamma = 1	SVM - RBF kernel
Dataset1	200 dimensional	58.38	C = 550, gamma = 40	SVM - RBF kernel
Dataset1	200 dimensional	58.46	C = 500, gamma = 50	SVM - RBF kernel
Dataset1	200 dimensional	58.48	C = 750, gamma = 50	SVM - RBF kernel

Dataset1	200 dimensional	58.46	C = 1000, gamma = 50	SVM - RBF kernel
Dataset1	200 dimensional	58.7	C = 550 , gamma = 60	SVM - RBF kernel
Dataset1	200 dimensional	58.12	C = 550, gamma = 75	SVM - RBF kernel
Dataset 2	200 dimensional	58.34	C = 850, gamma = 47	SVM- RBF kernel
Dataset2	200 dimensional	58.16	C = 500, gamma = 50	SVM - RBF kernel
Dataset 3	100 dimensional	1.41935	Boosting stages=800	Gradient boosting regressor
Dataset 3	100 dimensional	1.8838	C=1000, penalty 'l2'	Logistic Regression
Dataset 3	100 dimensional	1.3378	C=500, g=50	SVM regression
Dataset 3	100 dimensional	1.3198	C=700, g=70	SVM regression
Dataset 3	100 dimensional	1.2534	C=1, gamma = 10	SVM regression

Final Prediction

Dataset	Classifier and Hyperparameters	Highest Accuracy on test data
Dataset 1	SVM with RBF kernel C = 550, gamma = 60	58.25
Dataset 2	SVM with RBF C=850, RBF=47	58.26
Dataset 3	SVM regression with RBF kernel C=1, gamma = 10	1.17296 (Least Square Error)

Conclusion

Our group won the first prize for datasets 1 & 2. We came third in dataset 3.