CS659: Assignment 3
Product Review Classification
Tema Name on *kaggle.com*: Assignment3

Muhammad Baqui and Irina Hashmi and Md. Alimoor Reza

March 28, 2013

0.1 Assignemnt Description

This assignment is competition based where we have to predict the ratings (1,2,4,5) for a given text review of a product like books and movies from Amazon's website. The goal of this competition is we are given a text review we would like to automatically predict it's 5-star rating. (5-highest and 1-lowest). We are given one training set with ratings, one test set for verification and one unlabeled set.

0.2 Methods

This section describes the steps in solving the assignment problem. The first section describes the initial challenges we faced to formulate the problem. The feature reduction scheme we have used is been discussed in the next section. The following section describes the classification technique we have used.

0.2.1 Initial Challengese:

Data representation: The very first challenge we face is how to parse the trianing and test set data and store them efficientlt for preprocessing. The total number of attriutes given in training set is 300045 having the total number of ratings 5744 while the test set has 403141 attributes with total data points of 5275. We first tried to parse the given training data and build a matrix of 300045 by 5744 to train the model which is expensive both in terms of accessing time as well as memory. Then we build a hashing scheme to represent this dataset and run PCA using MATLAB to project to the dataset to a smaller dimension. However, without feature reduction the computation of PCA is expensive. Thus the next step is to employ a feature reduction technique that can efficiently reduce the feature space.

0.2.2 Feature Reduction:

We have used frequency-based-feature-reduction technique. In this work we have tried two different approaches. The first approach we tried to remove some frequent terms that have no specific information about a particular class, for example, the word *'the'*, *'of'* are most frequent words in any english text which does not help discrimintaing a particular class. For this task we have implemented a python script that will prune all the attributes containing prepositions, articles and auxiliary verbs and any combinations of these types, since they do not carry any specific information. However, this approach did not help much reducing the total number of attributes. For example, out of 300045 attributes in the training set, this approach helps to reduce only 100 - 200 attributes from the set, still leaving a huge feature space. For this reason we tried our second frequency based approoach.

In this approach we have selected the attributes that is least common over the entire data set. For example: we calculate the frequency of each attriute a_i over entire data set. Let's say, if an attribute a_i is present in n out of N rating texts, then the frequency of attribute a_1, (lets denote it $freq_{a1}$) will be $(n/N) * 100$. If $freq_{a1}$ is less than a given threshold $freq_{th}$ then we discard a_i

Table 1: Comparative results among different kernel function

Type of kernel function	Parameters	Accuracy (%)
linear	-t(0)	0.445
polynomial	-t(1), -d(3)	xx
radial basis	-t(2), -g(0.1)	0.227
sigmoid	-t(3) -s(8)	0.255

from the trianing data set. After pruning this way, the rest of the attributes are preserved both in the testing and the training data set. This scheme helped us to reduce the feature space drastically. For example: for the given trianing the the total number of attributes before and after pruning is 300045 and 335. Therefore the dimension of the matrix is reduced to 5744 by 335. We have tested different values for $freq_{th}$ ranging from 5, 10 and 15% and finally we choose a conservative choice of 5% since the reduction does not seem to vary much for different values of $freq_{th}$. Here to mention, the training data set only contains the attributes present in the testing set, however, trainng set may have more attribtutes. This task is also implemented in python. Since the feature space reduced to a manageable space, next we directly applied the Support Vector Machine to classify the testing set.

0.2.3 Support Vector Machine (SVM):

We have used $SVM^{multiclass}$ package that uses the multi-class formulation. The reason for using this is it is capable of optimizing it with a technique that is fast in the linear case. For this assignment, we have tried different four different types of kernel function namely: linear, polynomial, sigmoid and radial basis function.

0.3 Results and Discussions:

Based on the above settings with the feature reduction we have tried four different settings from SVM multiclass package. Table 1 shows parameter settings and the accuracy obtained for each setting from *kaggle.com*. The linear kernel seems to perform better than other three based on kaggle's accuracy.