COSI 123 Statistical Maching Learning Term Project

Chuan Wang,* Chen Xing,* Long Sha,* Zewen Peng, and Hang Yang

E-mail: cwang24@brandeis.edu; xingchen0@gmail.com; longsha@brandeis.edu

Abstract

Our team tries to build a classifir which gives the best predictions for the test set given. Since the training data size is large and the data has very skewed distribution, we try different methods of sampling which aims to reduce the data size and also balance the data distribution. We also try some feature selection methods to validate the effectiveness of the feature extracted. At last, we train a variety of classifiers and estimate their weighted avg. F-measure performance. Overall we achieved the best weighted avg. F-score of 0.47 with 5-fold cross validation on the 10% subset of the training set by using the SVM with RBF kernel. However, the performance of logistic regression is competitive with the weighted avg. F-score 0.465. Also Logistic Regression takes much less time than the SVM with RBF Kernel to train the model. So we use the predictions from the Logistic Regression model trained on the whole training set for this assignment.

1. Data pre-processing

Data preprocessing is an important step in building machine learning systems. In this project, the training data set comes from the review. The features extracted are 291 dimension vectors with each elements representing TD-IDF value of a word and the review rating (1,2,3,4,5) serves as the class label. Since the data set is quite large (with 297066 instances) and suffers from unbalanced distribution(see Figure 1), we have tried downsample method to address these problems.

*To whom correspondence should be addressed

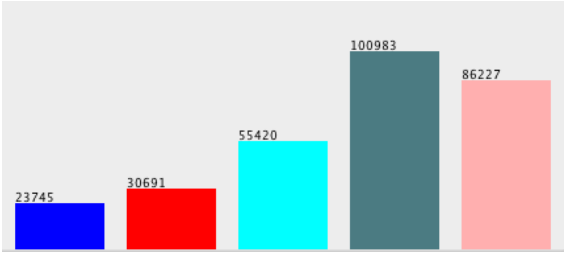

Figure 1: The training data distribution with 5 classes

1.1 Data sampling

Since the original data size is too large, it will take long time to do the cross-validation and parameter optimization. We propose the downsample 10% of the training data set as validation set which is used for choosing the classifiers and tuning the model.

Balancing and unbalancing

As we noticed that the class 4 & 5 had pike in the number of data in the training data set. So we try a balanced sampling method which resample 10% of the data samples by setting a uniform bias on the distribution. And the result is as follows:

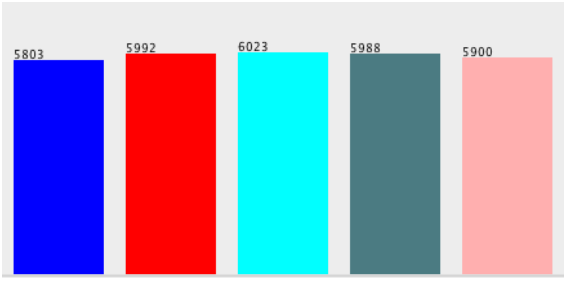

Figure 2: The 10% balanced training data distribution

Replacement and non-replacement

Also when sampling the validation data set we are curious about how the performance of the same method can be different when we use sampling with replacement and non-replacement. Since resample with replacement will add same instance many times in the result data set while no-

replacement will not, we have done several experiments to find out which method will be better at fitting the gold truth.

2. Evaluation method

The goal of this project is to predict what rating score should give according to the feature vector. It is a multi-class classification task. And we want to know the classifier's performance on each class. The F1 score, commonly used in information retrieval, measures accuracy using the statistical precision p and recall r.

Also, we mainly focus on the overall performance of the classifier. We may tolarance the low f-measure of the classes with fewer instances and gain more weights on the classes with more training instances, which is a fairer metric since the performance tends to be better with more training instances. So the evaluation metric we use for this project is weighted mean F1-Score, which is given by:

$$F_{weighted_avg.} = \sum_{i}^{C} \frac{\#\ of\ instances\ in\ class\ i}{total\ \#\ of\ instances} * F_i$$

where C is the number of classes and F_i is the f score of class i.

3. Feature Selection

Also, the 291 dimensional feature vector may contain some noise which will degrade the performance of the classifier. So we propose to select the top N ranked attributes to see whether it will improve the classifier's performance. And we choose two most effective feature selection methods,[1] Information gain and ChiSquared to filter out the less informative attributes.

3.1 Information Gain

The idea behind information gain is to select features that reveal the most information about the classes. Ideally such features are highly discriminative and occur in single class.

3.2 Chi-squared

This feature selection method tests for presence/absence of relation random variables. It bivariate analysis tests 2 random variables and can test strength of relationship.

4. Models and Classfiers

4.1 Naive Bayes Multinominal

Naive Bayes is the natural baseline for text classfication problem. Although based on the simple feature independence assumption, naive bayes works quite well on some of the text classification problems. However, it suffers from the drawbacks that it cannot capture the feature interaction and overlapping problem. So it may not be suitable for the task of which features are highly correlated.

4.2 Logistic Regression

Logistic Regression is one type of linear classifier, which can be used when the data set is linearly separable. Also, logistic regression is prefered when the dependent variable is categorical. In this data set, the labels to be predicted are review rating which are almost independent with each other. Although it is often used in binary classification problem, weka provides the implementation of multinominal version. So we try the multinomial version of logsitic regression in weka.

4.3 Support Vector Machine

SVM is one of the most powerful classifier under most of the cases. By applying the kernel function to map the instances into high-dimensional feature spaces, we can perform non-linear classifica-

tion. However, in practice it is very hard to decide whether the classes are linearly separable on the feature space. So we both try the linear kernel and RBF Kernel.

4.4 K-nearest neighbors

Amongst the simplest of all the machine learning algorithms, k-NN assigns the instances into the class which is most common among its k nearest neighbors. However, the main drawback of nearest neighbor is example of a more frequent class tend to dominate the prediction of the new example, which makes it not suitable for the situation where the class distribution is skewed. So k-NN may not perform well on this unbalanced data.

4.5 Artificial neural network

Artificial neural network (ANN) is a flexible mathematical structure which is capable of identifying complex nonlinear relationships between input and output data sets. ANN models have been found useful and efficient, particularly in problems for which the characteristics of the processes are difficult to describe using physical equations.[2] Here, we use the software *neuroshell2* to conduct the ANN experiment.

5. Experiments and discussions

5.1 Balanced Data vs Unbalanced Data

First, we do the experiment to compare the performance of balanced and unbalanced data. We separate the whole data set into two parts: 80% as training set, 20% as develop set and then train SMO linear kernel with balanced and unbalanced data set seperately. At last, test the model's performance on the develop set. From Figure 3, we can see that unbalanced data outperforms the balanced data in general. And the balanced sampling with no replacement gets closer to the unbalanced data performance when the training set increases. This is because the resampling with

no replacement will not generate the duplicate data instance. So as the data size increases, the distribution of unbalanced resampling with no replacement will be more and more similar to the unbalanced distribution.

So we can conclude that the balancing method does not help much. And the experiment follows will use unbalanced data.

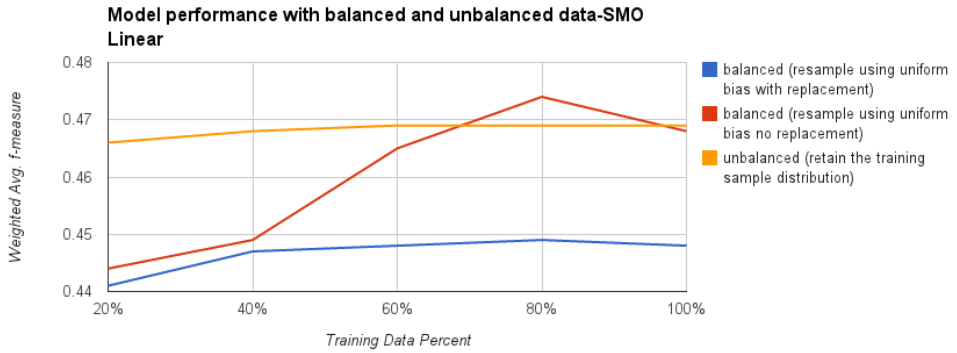

Figure 3: Model Performance with balanced and unbalanced training data set

5.2 Feature Selection

To check the effectiveness of feature selection, we choose logistic regression as the classifier and use 10% of the training set(resample with no replacement) to train the model. We separately choose top 30 and 100 ranked attributes by using information gain and Chi-squared evaluation method. Then we run 5-fold cross validation to estimate the performance of the classifier. Table 1 shows the result of feature selection.

Table 1: Feature Selection result

Model	**Overall Accuracy(%)**	**Weighted Avg. F-measure(%)**
Logistic_inforgain_top30	43.9	42.1
Logistic_inforgain_top100	47.4	46.1
Logistic_ChiSquared_top30	44.1	42.3
Logistic_ChiSquared_top100	46.8	45.5
Logistic_with_no_FS	**47.4383**	**46.5**

We can see from the table that the feature selection does not help so much. Feature selection

done by ChiSquared is even worse. But it may be too early to say that feature selection is useless. Since we haven't tried feature selection metric and we only choose top 30 and 100 features, further experiments should be done to provide more information.

5.3 Classifier Performance Comparison

Table 2: 5-fold cross validation results

Classifier	**Overall Accuracy(%)**	**Weighted Avg. F-measure(%)**
NaiveBayesMultinominal	43.025	36.4
Logistic Regression	47.4383	46.5
SVM Linear Kernel	47.8	46.3
SVM PolyKernel	47.6	46.32
SVM RBFKernel	**48.24**	**47**
KNN (K=3)	30.13	25.1
Artificial Neural Network	37.5	35.5

Table 2 shows the performance of the 7 classifers we have tried. All the models are trained on the 10% subset of the whole training set and estimated by 5-fold cross validation. We also provide the accuracy performance for comparison.

It turns out that the SVM with RBF Kernel outperforms all the other models with the highest accuracy 48.24% and weighted F-measure 47%, which makes sense because the number of instances is much bigger than the number of features (297066 and 291). And projecting the data instances makes it easier to separate the data points.[3] KNN is worsest classifier which indicates that the skewed data distribution makes it hard to do the classification based on "majority voting". Also, we notice that the performance of Logistic Regression is compatitive to the SVM with RBF Kernel and also the time cost of training Logistic Regression is much less than SVM, which makes it the practical model for application.

6. Conclusion

The series of experiments we conduct in this project suggests that SVM with RBF Kernel classifier performed the best on the downsample of the whole training set. Also, the performance of logistic regression is competitive with marginal drop on weighted avg. F-measure and also costs less time for training. Also, simply setting the bias to fit the data into uniform distribution cannot solve the unbalanced distribution problem and we may try some model weight tuning in the future.

7. Contribution

Chuan tried libsvm toolset and svm tools with different kernel in weka and took part in experiments about sampling methods and classifier comparison. Chuan also helped with the writing and formatted it into LaTeX.

Long took part in the experiment work load and helped to make the strategy. He also tried two features in the feature selection part and wrote parts of the write up.

Chen has try neural network in different software (*neuroshell2*),and figured out how to use the software to try different parameters,structures(neurons,levels). Chen also did the KNN experiment.

Hang took part in optimizing parameters for different classifiers and help with draw the diagrams.

Zewen helps to train the Naive Bayes classfier and also helps tuning the SVM models.

References

(1) Guyon, I.; Elisseeff, A. *J. Mach. Learn. Res.* **2003**, *3*, 1157–1182.

(2) Hsu, H. V. G., K.; Sorooshian, S. *Artificial Neural Network Modeling of the Rainfall-Runoff Process*; 1995.

(3) Hsu, C.-W.; Chang, C.-C.; Lin, C.-J. A practical guide to support vector classification. 2003.