Empirical Analysis of ML algorithms

Avideep Mukherjee
18111264

Soumya Banerjee
18111271

Nayan Das
18111044

Amit Chandak
18111262

Riya James
18111054

Sristi Jaiswal
18111074

Disclaimer

The work carried out in this project has not been re-used from any another course project at IITK or elsewhere, or any other project we might have done elsewhere.

1 Problem description

We have done empirical analysis of ML algorithms on 3 different subproblems-analysis of ML algorithms on text data, performing classification with the help of clustering, study of classification on imbalanced datasets and study of kernel approximation methods. The details of each of these subproblems is given separately in the report.

2 Tools/Softwares used

The codes that are written for generating the results are written in Python and for the ML Algorithms,scikit-learn is used. Other libraries include, numpy, pandas and scipy.

3 ML algorithms for text data

3.1 Sub-Problem Description

In this sub-problem we compare the performances of different ML Algorithms with deep learning classifiers on text data.

3.2 Literature Review

Joulin et. al [3] presented a comparative study of deep learning algorithms on three standard text datasets. Among classical ML algorithms they have only presented results for SVM. Due to lack of hardware resources, we could not run deep learning algorithms and have taken the results that they have shown.

3.3 Experimental Setup

3.3.1 Data Description and Pre-processing

We have taken the three standard text datasets used by Joulin et. al.[3], two review datasets from Yelp, one from 2013 and another from 2014. The third dataset is the review dataset from IMDB. The class labels of each of the three datasets are basically ratings. So, IMDB dataset has 10 classes from 1 to 10, since, they have a rating system like that. Similarly, Yelp datasets have 5 class labels. The datasets are pre-processed as follows.

For each review in the dataset :

- English stopwords are removed.
- Punctuation and other special characters (e.g. new line) are removed.
- Words are stemmed to filter the root words using Porter's Stemming Algorithm.

After the above pre-processing, TF-IDF scores of each words are generated from the whole dataset. This is how the final pre-processed feature matrix is generated. One exception that needs to be mentioned is that Yelp'14 had 183019 inputs both in the training as well as test data. Running classifiers on such a huge data would require considerable amount of time and hence we undersampled the training data only keeping in mind the class distributions. Every class has equal proportion of representative samples as was there in the original data. Undersampling was done to the level of the training input sizes of the other two datasets. Hence, 40% of the training data is used in Yelp'14. The overview of the datasets are given in Table 1.

Dataset	#reviews (Train)	#reviews (Test)	#words	#categories
Yelp'13	62522	8671	49251	5
Yelp'14	73206	183019	87637	5
IMDB	67426	9112	48883	10

Table 1: Overview of the text datasets used

3.3.2 Experimental Results

We have experimented the datasets with some classical and deep learning algorithms as shown in 2. We have also reported the performances of *fasttext* on the three datasets. The test accuracy that are obtained by using each of these classifiers are shown below in Table 2. The accuracy that are maximum among all the classifiers for a given dataset are marked as **bold**.

Algorithm	Yelp'13	Yelp'14	IMDB
SVM[3]	59.8	61.8	40.5
Logistic Regression	55.4	56.1	31.0
Random Forest	47.0	55.8	27.0
Naive Bayes	50.0	57.4	27.3
CNN[3]	59.7	61.0	37.5
Conv-GRNN[6]	63.7	65.5	42.5
LSTM-GRNN[6]	**65.1**	**67.1**	**45.3**
fasttext[3]	64.2	66.2	45.2

Table 2: Test Accuracy (%) of Different Classifiers (both classical and deep learning) over the three datasets

3.4 Discussion

From Table 2 it is clearly observed that Deep Learning Algorithms perform better than the classical ML Algorithms for these three datasets. The key observations that are noted from the different performances are listed as under :

- Performance of Random Forest on all the datasets is poorer than that of Naive Bayes
- Although Deep Learning Algorithm (LSTM-GRNN) outperformed all the other classifiers, it is interesting to note that the performance of SVM is always better than that of CNN, a deep learning algorithm.
- Since, the data distribution of Yelp'13 and Yelp'14 are similar, therefore the performance of the classifiers are also similar in both the cases. Only exceptions are Random Forest and Naive Bayes, both of them perform better in Yelp'14 than Yelp'13.

3.5 Future Works

Since we compared the performances of the classical Machine Learning algorithms with the results from Joulin et. al.[3], we had to compare with only test accuracy of the respective classifiers. As a future work, the performances should also be evaluated using other metrics such as Precision, Recall or F1-Score. Also more text datasets could be tested and experimented upon to solidify the observations. Lastly, other text-classification feature extraction techniques, like String Kernels could be used to compare and evaluate the performances.

4 Classification with the help of Clustering

4.1 Sub-Problem Description

There are many classification algorithm available like -Decision Tree,SVM,Logistic Regression which perform well on linearly separable low dimensional dataset.We can always take help of feature selection and dimensionality reduction technique like PCA,t-SNE algorithm to work on high dimensional dataset.We are exploring a technique where we will take help of clustering for classification of high dimensional non linear dataset.

4.2 Literature Review

There has been many works done in this area where the researcher uses different ensemble methods like bagging, boosting[2] to improve efficiency of classifier. They also used Mixture of expert model [4] where different expert is modeled on part of the training data and combine them using the gating function. Similarly This method improves familiar classification model accuracy with the help of clustering method[7] [1].

4.3 Experimental Setup

4.3.1 Data Description and Pre-processing

We have chosen two datasets- from UCI repository and handwritten letter data set from kaggle for our experimental analysis.

4.3.2 Experimental Results

Logistic Regression is train with multiclass setting with l1 norm.We have given result for best K (cluster) in the report

We set baseline by running individual classification algorithm on the dataset. Then,We have run clustering algorithm on each dataset and we ran classification on each cluster.

Data Set			
Data Set Name	Number of Instance	No of Class	Number of feature
MNIST Digit Recognizer	42000	10	784
A Z Handwritten Data	372450	26	784
Credit Card Data	30000	2	21

Table 3: MINIST Data Set

Type	Accuracy			F1-Score			Precision		
	Logistic	DTree	SVM	Logistic	DTree	SVM	Logistic	DTree	SVM
Baseline	91.70	86.08	85.70	91.68	86.06	85.68	91.68	86.06	85.68
Kmeans(10)	94.32	87.94	92.99	93.78	88.14	92.87	93.60	88.56	92.98
GMM(10)	94.33	88.97	93.33	93.58	88.95	93.28	93.45	89.21	93.59

Table 4: A Z Handwritten Letter Data

Type	Accuracy			F1-Score			Precision		
	Logistic	DTree	SVM	Logistic	DTree	SVM	Logistic	DTree	SVM
Baseline	91.09	95.53	82.75	91.06	95.53	82.79	91.06	95.55	85.15
Kmeans(26)	93.88	95.98	95.41	93.37	96.06	95.37	93.26	88.56	95.80
GMM(26)	93.56	96.06	95.63	92.98	96	95.59	92.82	96.21	95.80

Table 5: Credit Card Data

Type	Accuracy			F1-Score			Precision		
	Logistic	DTree	SVM	Logistic	DTree	SVM	Logistic	DTree	SVM
Baseline	78.54	72.88	78.50	69.53	73.05	69.65	79.89	73.23	76.87
Kmeans(2)	79.40	73.72	79.59	70.35	74.19	70.89	65.60	74.74	95.80
GMM(2)	78.55	73.02	78.63	70.35	73.41	69.90	67.63	73.86	70.68

4.4 Discussion

Prediction result is improving and able to handle non linear data.Classification perform well in similar structured data generated by clustering and identify distinct distributions of the data generated from.

4.5 Future Works

We can try Mixture Expert model or different feature selection menthod like PCA,t-SNE and try compare with our result.

5 ML Algorithms for imbalanced datasets

5.1 Sub-Problem Description

We have compared the performance of classification of various classification algorithms like Prototype based classification, Naive Bayes, Decision Tree, kNN, Logistic regression and SVM by using several techniques to handle imbalance datasets like random undersampling, random oversampling, clustering based undersampling, clustering based oversampling and smote. We have tried the algorithms on datasets with different imbalance ratios. In clusering based undersampling, we have clustered the majority class with no. of clusters equal to the no. of data in the minority class and then assumed the centres of clusters as the data for the majority class. In clustering based oversampling, we have clustered the data of both classes and oversampled each cluster, such that the no. of samples in majority class and minority class becomes equal. We have also classified by reweighting minority class and using decision tree, logistic regression and SVM. Apart from this, we have used bagging, random forest, gradient boosting and easyensemble. Easyensemble randomly samples as

many data points from the majority class as there are in the minority class and uses this as the input for each tree in the ensemble. A simple voting measure is used to aggregate the predicitions. We have also studied the change in recall with change in number of clusters of minority class.

5.2 Literature review

Imbalanced datasets pose a problem in classification and hence enormous amount of research has been done in this area. The various approaches can be divided into two broad categories- (i) modifying the dataset (ii) changing the learning algorithm.

For the first approach, there are various techniques that include changing the size of data of each class. For the second technique, the loss function can be changed. Apart from these, there are several other classification strategies that handle imbalanced datasets like ensemble methods. Moreover, the empirical analysis is done on different types of datasets with different imbalances.

5.3 Experimental Setup

5.3.1 Data Description

The dataset we used is a collection of 30000 legitimate and fraudulent transactions made over a two day period at an unspecified bank (to respect user privacy). The dataset was downsampled to make an imbalance ratio of 0.001, 0.01, 0.1. Most of the 30 features are PCA post-processed and are small-valued real numbers.

5.4 Experimental Results

Our F1 Scores on various algorithms on different techniques are listed in the following tables.

Method	Prototype	Naive Bayes	Decision tree	kNN	Logistic	SVM
Original	0.87	0.81	0.83	0.14	0.75	0.43
Undersampling	0.88	0.85	0.89	0.87	0.91	0.69
Oversampling	0.87	0.87	0.79	0.57	0.91	0.43
Cluster US	0.86	0.87	0.73	0.87	0.92	0.43
Cluster OS	0.87	0.87	0.81	0.59	0.90	0.44
Smote	0.88	0.88	0.72	0.64	0.91	0.67
Reweighted	-	-	0.83	-	0.87	0.67

Table 6: Dataset with imbalance ratio 0.001

Method	Prototype	Naive Bayes	Decision tree	kNN	Logistic	SVM
Original	0.84	0.76	0.91	0.57	0.88	0.50
Undersampling	0.84	0.84	0.87	0.90	0.95	0.53
Oversampling	0.84	0.84	0.87	0.69	0.94	0.50
Cluster US	0.87	0.90	0.73	0.87	0.94	0.56
Cluster OS	0.83	0.86	0.87	0.71	0.95	0.43
Smote	0.84	0.87	0.87	0.98	0.94	0.48
Reweighted	-	-	0.86	-	0.91	0.88

Table 7: Dataset with imbalance ratio 0.01

Method	Prototype	Naive Bayes	Decision tree	kNN	Logistic	SVM
Original	0.90	0.85	0.94	0.47	0.93	0.40
Undersampling	0.88	0.87	0.93	0.83	0.96	0.40
Oversampling	0.90	0.86	0.93	0.52	0.96	0.40
Cluster US	0.90	0.90	0.87	0.89	0.94	0.33
Cluster OS	0.90	0.88	0.91	0.69	0.96	0.58
Smote	0.90	0.87	0.91	0.69	0.95	0.40
Reweighted	-	-	0.91	-	0.96	0.58

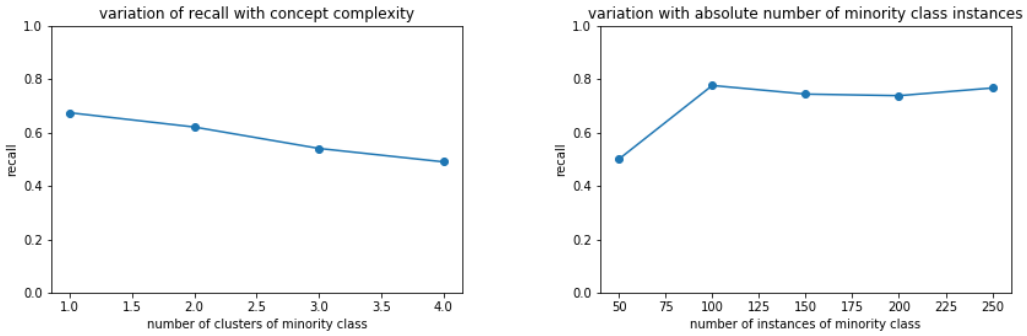
Table 8: Dataset with imbalance ratio 0.1

Ensemble-based methods:

Method	Bagging		Random forest		EasyEnsemble		Gradient Boosting	
Metric	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Value	0.87	0.75	0.93	0.75	0.84	0.71	0.84	0.72

Analysis on our Synthetically Generated Data

We had come across multiple research papers that suggested class imbalance isn't in itself the real issue when dealing with skewed data. But that the real issue was small disjuncts of the minority class in the dataspace that didn't allow for effective modeling. To test this, and see how much of an effect other factors have on prediction metrics, we generated our own 2D data (with relative imbalance = 0.01) and varied some of its properties. We ran it through a random forest (that keeps both features).



From the first figure, we can say that as the number of clusters of minority class increases, the recall rate decreases. The second figure shows that as the number of instances of minority class increases, the recall slightly increases even if the imbalance ratio is kept fixed.

5.5 Discussion

- The F1 score is mostly increasing with decrease in data imbalance.
- The F1 score of prototype based classification is nearly same on original and other data, probably because imbalance in the dataset was created by downsampling, hence, both the classes have nearly equal spread.
- Naive Bayes is improving with the various techniques applied as it can learn the class distribution well. However, this improvement is not significant when the classes are less imbalanced.
- Decision tree is working well even in the original dataset as it is not affected by class imbalances.
- In our dataset, logistic regression is giving very high score on various techniques. The score of SVM is lesser has we have run it for only 200 iterations as it was taking a long time.
- We can also observe that the score of oversampling and reweighting are mostly similar.
- Ensemble methods also help in handling imbalanced datasets, but they are a bit expensive.

5.6 Future Works

The performance can be measured using area under ROC curve etc. These algorithms can be tried on different types of datasets by changing their distribution, dimensions, attribute type to observe the change in performance.

6 Kernel Approximation Methods

6.1 Sub-Problem Description

This was an attempt to understand, implement and analyze popular kernel approximation methods. Kernel approximation methods help to speed-up kernel learning. Kernel learning is important for handling non-linear datasets.

6.2 Literature review

These papers [5] [8] were referred and an attempt was made to replicate the results by implementing the approach and measuring performance on different datasets. Random Fourier Features(RFF) and Orthogonal Random Features(ORF) methods have been implemented. The [8] claims that ORF's are better in terms of computation and storage cost while achieving same performance as RFF. Also tried kernel garbage collection approach where in dimensionality of randomly chosen projection vectors are reduced before projecting the data on these vectors for new representation.

6.3 Experimental Setup

The above approaches were implemented in python and sklearn's SVM was used for classification task. Dataset was randomly split in to train-test set in the ratio 80:20. Since this exercise is mostly to compare speed-up gained by using kernel approximation, deafult SVM parameters were used. Baseline numbers are SVM's with RBF kernel. Mean accuracy is measured since it's multi-class classification task. Note, that since CIFAR dataset is around 1.5GB , experiments were taking too long to complete and hence not including it's results.

6.3.1 Dataset Description

Data Set			
Data Set	Instances	Class	Dimensionality
MNIST	42K	10	784
USPS	10K	10	256
CIFAR	60K	10	3073

Table 9: MNIST DataSet

Details	Tr.Ti(s)	Pr.Ti/Ex.[μs]	Mean Accu.%
K-SVM	393.38	13351.69	75.27
W=1200,(RFF, ORF)	(229.7, 168.6)	(69.6, 46.1)	(96.85, 96.5)
W=784,(RFF, ORF)	(167.2, 160.0)	(46.0, 46.0)	(96.39, 96.46)
W=256,(RFF, ORF)	(96.3, 95.5)	(17.7, 18.3)	(94.15, 94.58)
W=128,(RFF, ORF)	(71.37, 67.6)	(9.6, 10.0)	(92.5, 92.8)
W=64,(RFF, ORF)	(43.79, 54.4)	(4.38, 4.36)	(88.49, 88.49)
W=1200 followed by PCA			
W-Red= 784,(RFF, ORF)	(227.7, 160.78)	(69.5,46.0)	(96.85, 96.49)
W-Red= 256,(RFF, ORF)	(231.4, 160.9)	(83.3,61.4)	(96.85, 96.49)

6.4 Discussion

Though the paper [8] claims ORF is better than RFF , I couldn't replicate the same numbers. There might be some minor bug in the code or I am missing something. In most of the cases RFF is

Table 10: USPS DataSet

Details	Tr.Ti(s)	Pr.Ti/Ex.[μs]	Mean Accu.%
W=1200,(RFF, ORF)	(21.5, 4.87)	(55.6, 13.6)	(93.82, 92.83)
W=784,(RFF, ORF)	(14.6, 4.9)	(36.5, 13.2)	(93.37, 92.83)
W=256,(RFF, ORF)	(5.3, 4.9)	(13.5, 13.2)	(93.22, 92.97)
W=64,(RFF, ORF)	(2.23, 2.17)	(4.2, 3.5)	(90.6, 89.89)
W=1200 followed by PCA			
W-Red= 256,(RFF, ORF)	(21.3,4.7)	(54.0,13.4)	(93.82, 92.83)
W-Red= 64,(RFF, ORF)	(21.6, 4.8)	(54.9,13.8)	(93.82, 92.83)

performing better in terms of accuracy. As expected, kernel approximation methods reduce training time and significantly reduce prediction time. Also the number of components to sample is important and can be chosen based on specific accuracy requirements. Dimensionality reduction on sampled 'W' is giving better performance in all cases suggesting that since 'W' is sampled randomly there is a high probability of repetated vectors being chosen.

6.5 Future Work

Complete the study and understand the unexpected results of ORF. Also compare with other kernel-approximation techniques like Fast-Food, memory efficient kernel approximation, etc..

References

[1] J. Ghosh A. Acharya, E. R. Hruschka and S. Acharyya. c3e: A framework for combining ensembles of classifiers and clusterers,. *MCS. Berlin, Heidelberg: Springer-Verlag, pp. 269278*, 2011.

[2] T. G. Dietterich. Ensemble methods in machine learning. *International workshop on Multiple Classifier Systems. Kittler J., and Roli., F. (Eds.), Lecture Notes in Computer Science, New York, Springer Varlag*, 2000.

[3] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

[4] Jong-Min Park and Yu Hen Hu. Estimation of correctness region using clustering in mixture of experts.

[5] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1177–1184. Curran Associates, Inc., 2008.

[6] Duyu Tang, Bing Qin, and Ting Liu. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1422–1432, 2015.

[7] Y. Zhang K. Huang X. Zhang, P. Yang and C. Liuich. Combination of classification and clustering results with label propagation. *IEEE Signal Process. Lett., vol. 21, no. 5, pp. 610614*, 2014.

[8] Felix X. Yu, Ananda Theertha Suresh, Krzysztof Choromanski, Daniel N. Holtmann-Rice, and Sanjiv Kumar. Orthogonal random features. *CoRR*, abs/1610.09072, 2016.