CS229: Machine Learning Project Report
An Ensemble Classifier for Rectifying Classification Error

Cheuk Ting LI ctli@stanford.edu

December 14, 2013

1 Introduction

In the field of classification, apart from building a single powerful classifier, efforts have been made on ensemble methods which combine several classifiers to give results better than any one of them. Some notable examples are boosting (Freund and Schapire 1997) and stacking (Wolpert 1992). In this project, we propose a new ensemble method in which each constituent classifier would focus on correcting errors made by the previous ones.

To explain this method, first focus on the case where there are two constituent classifiers. The first classifier, called the *assorter*, would perform classification on the training data to produce a ranking of the classes for each training instance (rank 1 is the class which is the most likely, rank 2 is the second likely, etc). For each instance, we find the rank of the correct class (rank is 1 if the prediction of the assorter is correct; rank is 2 if the prediction is wrong, but the assorter decides that the correct class is the second most likely, etc). Then the second classifier, called the *rectifier*, would use the rank as the class variable (and discard the original class variable) to perform classification. Intuitively, the rectifier would decide whether we should accept the prediction made by the assorter, or pick from the classes which are determined as less likely by the assorter. To classify a test instance, run the assorter to obtain a ranking, and pick the position given by the rectifier. This method can be generalized to more than two constituent classifiers, where each subsequent classifier would try to rectify the results made by the previous classifiers.

Experiments were conducted to investigate the classification accuracy of the assorter-rectifier method. The results suggest that the assorter-rectifier method, with suitable choices of assorter and rectifier, outperforms its constituent classifiers, and many other state-of-the-art classifiers.

2 Assorter-Rectifier Method

In this section, we would describe the assorter and rectifier, and how the outputs of the two classifiers are combined.

2.1 Assorter

The assorter serves as the "base" classifier, which gives the initial prediction of the class. The assorter has to be able to produce a distribution (or at least a ranking) of the classes for a test instance. Simple parametric classifiers are more suitable choices for the assorter.

2.2 Rectifier

The rectifier would try to correct the prediction made by the assorter. The rectifier does not need to produce a distribution of the classes. It only needs to pick one class as the prediction. Local non-parametric classifiers are more suitable choices for the rectifier, since the rectifier has to capture small clusters near the decision boundary of the assorter which are incorrectly classified by the assorter.

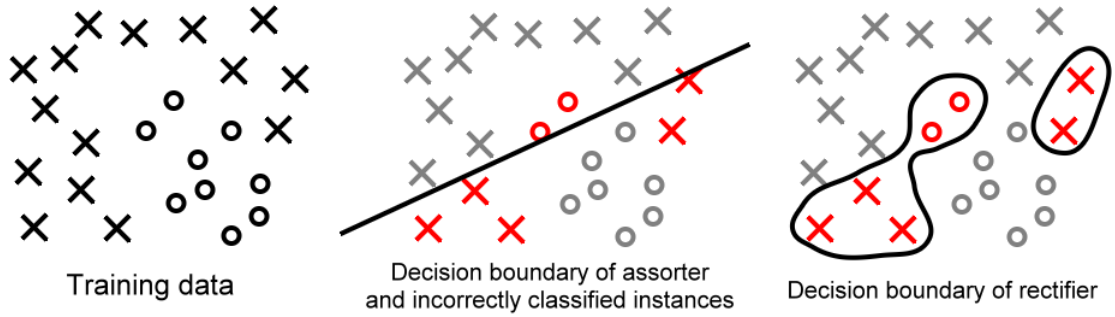

Figure 1: Illustration of assorter and rectifier on a hypothetical data set

2.3 Combining the Results

Suppose there are k different classes labeled $1, ..., k$. Let $\vec{h}_a(x) \in \mathbb{R}^k$ denotes the distribution of classes predicted by the assorter for instance x (i.e., $\left(\vec{h}_a(x)\right)_i = \mathbb{P}\{y = i \,|\, x\}$). Let $h_r(x) \in \{1, ..., k\}$ denotes the rank of the correct class guessed by the rectifier. Define the function $\mathrm{r}(i, \vec{h})$ to be the index of the element in $h_1, ..., h_k$ ranked the i-th ($\mathrm{r}(1, \vec{h}) = y$ if h_y is the largest among $h_1, ..., h_k$, etc). Define the function $\mathrm{r}^{-1}(y, \vec{h})$ to be the rank of h_y among $h_1, ..., h_k$ ($\mathrm{r}^{-1}(y, \vec{h}) = 1$ if h_y is the largest, etc).

To train the classifier, we first train the assorter with the original training data $\left\{\left(x^{(i)}, y^{(i)}\right)\right\}_{i=1,...,m}$. Then we would replace the class variables $y^{(i)}$ by the rank of the correct class $r^{-1}(y^{(i)}, \vec{h}_a(x^{(i)}))$, and train the rectifier on the modified data $\left\{\left(x^{(i)},\, r^{-1}(y^{(i)}, \vec{h}_a(x^{(i)}))\right)\right\}_{i=1,...,m}$.

To obtain the prediction for a test instance x, we first use the assorter to obtain a distribution $\vec{h}_a(x)$, and then output the class corresponding to the rank guessed by the rectifier, i.e., the predicted class is $\mathrm{r}(h_r(x), \vec{h}_a(x))$. Figure 1 illustrates the idea of assorter and rectifier.

Note that if the rectifier just outputs the class which has the highest frequency in the training data for any test instance, then the assorter-rectifier method would reduce to using the assorter alone (assume the assorter has >50% accuracy). Intuitively, if the rectifier would not perform worse than just choosing the most frequent class regardless of the test instance, then the assorter-rectifier method would not be worse than using the assorter alone.

2.4 Appending Assorter Output to Feature Vector

In the current method, to train the classifier, the output of the assorter is only used in finding the rank of the correct class. The rectifier does not know the prediction made by the assorter. We can supply the prediction by the assorter to the rectifier as an additional feature (i.e., append it to the vector $x^{(i)}$), so that the rectifier can use the prediction made by the assorter.

3 Experiments

We have performed experiments on different combinations of assorter and rectifier. The choices of assorter are:

Naive Bayes Naive Bayes is a simple parametric classifier with low variance, a suitable choice for assorter.

Tree Augmented Naive Bayes (TAN) (Friedman and Goldszmidt 1996) Tree augmented naive Bayes is a Bayesian network searching algorithm, which finds the best tree-shaped Bayesian network on the attributes (conditioned on the class). Compared to naive Bayes, it has higher variance but lower bias. It usually outperforms naive Bayes for moderately-sized datasets.

Averaged One-Dependence Estimator (AODE) (Webb et al. 2005) The averaged one-dependence estimator is a method which aggregates several Bayesian networks, where each network is a tree where one attribute

is taken as the root, and all other attributes are children of the root (conditioned on the class). Compared to naive Bayes, it has higher variance but lower bias.

Logistic Regression Logistic Regression is a simple discriminative parametric classifier (all the classifiers above are generative) with low variance, a suitable choice for assorter.

The choices of rectifier are:

k-Nearest Neighbors A simple non-parametric classifier which takes the average of k nearest training instances of the test instance in order to produce the prediction. We would use $k = 1$ and 5.

Decision Tree We used the C4.5 decision tree learning algorithm (Quinlan 1993).

Support Vector Machine Support vector machine, when used with a high-dimensional kernel, can be used separate small clusters, and thus is a suitable choice for rectifier.

We have implemented the method in Java. The method was tested using Weka (Witten et al. 2011). We used the 36 datasets recommended by Weka (except the datasets "letter", "mushroom" and "waveform" due to size constraint). These datasets were taken from the UCI repository (Frank and Asuncion 2010), and were downloaded at the website of Weka. Ten-fold cross validation was performed.

The average classification accuracy for each combination of assorter and rectifier, together with the average accuracy when using each assorter/rectifier alone, is given in Table 1. We found out that appending the assorter output to the feature vector always improves the accuracy (except for AODE + SVM, where the accuracies are nearly the same). The combination of AODE as assorter and decision tree as rectifier gives the best average accuracy.

We then tested the assorter-rectifier method with AODE and decision tree against NB, TAN, AODE, Logistic Regression, nearest neighbor, 5-nearest neighbors, decision tree and SVM. The classification accuracy for each data set is given in Table 2. Paired T-tests were performed for each dataset. We can see that the average accuracy of assorter-rectifier is the highest, and is significantly more accurate for many datasets.

Table 1: Average accuracy for each combination of assorter and rectifier. The numbers next to the assorters/rectifiers are the accuracies when using those assorters/rectifiers alone. The first number in each cell is the accuracy without appending the assorter output to the feature vector, and the second number is that with appending.

Accuracy (without append, with append)			Rectifiers			
			NN	5-NN	Decision Tree	SVM
			80.62	81.58	82.42	84.45
Assorters	NB	82.46	80.43, 81.24	82.71, 82.95	82.95, 83.42	82.57, 82.65
	TAN	83.82	81.72, 82.22	83.57, 83.66	83.96, 84.04	83.80, 83.81
	AODE	84.62	82.02, 82.53	84.48, 84.61	84.77, 84.89	84.52, 84.52
	Logistic Regression	82.15	80.62, 80.95	82.01, 82.08	82.15, 82.16	82.01, 82.03

4 Conclusion

In this project, we have proposed a new ensemble classifier, called the assorter-rectifier method, in which the first classifier (the assorter) would give an initial prediction of the class, and the second classifier (the rectifier) would

focus on correcting the mis-classified instances of the assorter. Experiment results suggest that the assorter-rectifier method outperforms its constituent classifiers, and many other state-of-the-art classifiers.

Several variants of the method may be investigated in the future. For example, we may add more rectifiers to the method. Each subsequent rectifier would try to correct the mis-classified instances of the previous assorter and rectifiers. The rectifiers should be ordered from higher bias to lower bias.

Another possible improvement is to break down the classes supplied to the rectifier. Instead of only using the rank of the correct class as the class variable for the rectifier, we can also retain the original class variable and add it to the new class variable. For example, if there are two classes A and B, then there will be three classes for the rectifier $\{1, 2A, 2B\}$, where 1 stands for the case where the prediction of the assorter is correct, $2A$ stands for the case where the prediction of the assorter is incorrect and A is the correct class, and $2B$ stands for the case where the prediction of the assorter is incorrect and B is the correct class. Although this will produce extra classes, hopefully the instances within the same classes will be more similar to each other.

References

Frank, A. and Asuncion, A.: 2010, UCI machine learning repository.
URL: *http://archive.ics.uci.edu/ml*

Freund, Y. and Schapire, R. E.: 1997, A decision-theoretic generalization of on-line learning and an application to boosting, *Journal of Computer and System Sciences* **55**(1), 119 – 139.

Friedman, N. and Goldszmidt, M.: 1996, Building classifiers using bayesian networks, *Proceedings of the Thirteen National Conference on Artificial Intelligence*, pp. 1277–1284.

Quinlan, J. R.: 1993, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Webb, G. I., Boughton, J. R. and Wang, Z.: 2005, Not so naive bayes: Aggregating one-dependence estimators, *Machine Learning* **58**(1), 5–24.

Witten, I. H., Frank, E. and Hall, M. A.: 2011, *Data Mining: Practical Machine Learning Tools and Techniques*, 3 edn, Kaufmann, Burlington.

Wolpert, D. H.: 1992, Stacked generalization, *Neural Networks* **5**(2), 241 – 259.

Table 2: Accuracy for each dataset

Dataset	AR	NB	TAN	AODE	Logistic	NN	5-NN	C4.5	SVM
anneal.ORIG	88.87	87.53	90.98	88.87	90.98	88.42	87.31	90.09	91.42
anneal	97.44	94.32 •	98.11	96.77	99.56	97.77	96.88	98.66	99.44 ○
audiology	72.53	71.23	71.58	71.66	79.62	74.74	60.57 •	78.32	81.78 ○
autos	76.50	64.83 •	79.93	76.50	75.14	82.33 ○	66.29 •	82.86	83.86 ○
balance-scale	89.76	91.36	86.72 •	89.76	98.56 ○	66.72 •	83.84 •	64.48 •	90.24
breast-cancer	71.33	72.06	69.63	71.33	68.95	65.04 •	73.78	75.54 ○	69.63
wisconsin-breast-cancer	96.85	97.28	94.70 •	96.85	92.85 •	95.99	94.99 •	93.56	95.85
horse-colic.ORIG	75.81	75.26	75.81	75.81	64.92 •	72.80	70.63	81.52	76.91
horse-colic	80.45	78.81	79.89	80.45	73.11 •	75.84	80.68	83.95	81.24
credit-rating	84.78	84.78	83.62	85.22	83.33	78.99	85.07	85.36	85.51
german-credit	76.90	76.30	76.10	76.90	73.70 •	69.10 •	71.50 •	72.80 •	75.50
pima-diabetes	76.70	75.40	74.48	76.70	75.27	67.06 •	69.14 •	73.83	73.70
Glass	62.68	60.32	59.83	62.68	54.70	63.51	58.92	57.92	65.37
cleveland	82.80	84.14	83.80	82.80	77.85	78.81	81.41	78.18	83.47
hungarian	84.72	84.05	82.34	84.72	77.22 •	79.59 •	81.36	80.02	82.03
heart-statlog	82.96	83.70	77.41	82.96	78.52	77.04	80.74	80.00	81.11
hepatitis	83.79	83.79	81.33	83.79	74.92	77.42	84.46	81.25	78.83
hypothyroid	93.56	92.79 •	93.16	93.56	93.45	89.82 •	93.03	93.27	93.53
ionosphere	91.73	90.89	91.19	91.73	86.32 •	90.32	89.44	86.62 •	88.60
iris	94.00	94.67	90.67	94.00	90.00	92.67	93.33	96.00	96.67
kr-vs-kp	98.31	87.89 •	92.05 •	91.24 •	97.56	89.96 •	96.03 •	99.44 ○	95.43 •
labor	91.67	93.33	88.00	91.67	91.33	86.33	91.67	82.33	84.67
lymphography	85.67	85.67	82.33	85.67	79.76	77.62	82.33	79.71	80.43
primary-tumor	47.49	46.89	45.12	47.49	43.98	35.64 •	41.26	40.11 •	46.90
segment	92.60	88.92 •	94.33 ○	92.60	93.77	93.59	90.74 •	93.20	94.50 ○
sick	97.91	96.74 •	97.61	97.48 •	97.61	97.51	97.51	98.25	97.59
sonar	81.31	77.50	75.57	81.31	76.48	79.31	80.79	70.69 •	76.00
soybean	93.40	92.08	95.75	93.40	93.99	91.64	90.76	92.39	93.85
splice	96.21	95.36 •	95.49	96.21	91.03 •	75.92 •	79.81 •	94.36 •	93.42 •
vehicle	72.58	61.82 •	72.81	72.58	65.48	66.91 •	70.57	71.17	70.33
vote	94.73	90.14 •	94.49	94.50	95.86	92.43	94.03	96.33	95.87
vowel	90.10	67.07 •	93.94 ○	90.10	80.91 •	93.54 ○	81.31 •	75.45 •	87.17
zoo	95.09	94.18	97.18	95.09	94.18	96.09	92.09	92.18	96.09
Average	84.89	82.46	83.82	84.62	82.15	80.62	81.58	82.42	84.45

○, • statistically significant improvement or degradation