CS 229 Final report
A Study Of Ensemble Methods In Machine Learning

Kwhangho Kim, Jeha Yang

Abstract

The idea of ensemble methodology is to build a predictive model by integrating multiple models. It is well-known that ensemble methods can be used for improving prediction performance. In this project we provide novel methods of combining multiple learners in classification tasks. We also present an empirical study with wine quality score data set and demonstrate superior predictive power of the proposed ensemble methods over simple voting or single model methods.

1 Introduction

An ensemble of classifiers is a set of classifiers whose individual decisions are combined in some way, typically by weighted or unweighted voting, to classify new examples with the goal of improving accuracy and reliability. As combining diverse, independent opinions in human decision-making to pursue a more protective mechanism, the Ensemble model provides an efficient way to improve accuracy and robustness over single model methods. Ensemble methods provide a huge practical usefulness in that it has the promise of reducing and perhaps even eliminating some key shortcomings of standard learning algorithms.

In this project, we focus on classifier ensembles and propose novel algorithms of combining method. Our methods provide very simple yet effective way to determine optimal weight of each classifier in a sense that it seeks not only empirical risk minimization but diversification of classifiers.

For application, we analyze Napa Valley Wine Quality data and show that all of our ensemble algorithms produce way better predictive performance than any single method. Our methods also show superior performance than simple voting method.

2 Related Work

In the recent years, experimental studies conducted by the machine-learning community show that combining the outputs of multiple classifiers reduces the generalization error (Quinlan, 1996, Opitz and Maclin, 1999, Kuncheva et al., 2004, Rokach, 2006). Ensemble methods are very effective, mainly due to the phenomenon that various types of classifiers have different inductive biases (Geman et al., 1995, Mitchell, 1997). Indeed, ensemble methods can effectively make use of such diversity to reduce the variance-error (Tumer and Ghosh, 1999, Ali and Pazzani, 1996) without increasing the bias-error. In certain situations, an ensemble can also reduce bias-error, as shown by the theory of large margin classifiers (Bartlett and Shawe-Taylor, 1998).

Given the potential usefulness of ensemble methods, it is not surprising that a vast number of methods is now available to researchers and practitioners. There are several factors that differentiate between the various ensembles methods. Rokach (2010) summarizes four factors as below:

1. Inter-classifiers relationship - How does each classifier affect the other classifiers
2. Combining method - The strategy of combining the classifiers
3. Diversity generator - How should we produce some sort of diversity between the classifiers
4. Ensemble size - The number of classifiers in the ensemble

In this project, we particularly focus on the combining method. Ali and Pazzani (1996) have compared several combination methods. More theoretical analysis has been developed for estimating the classification improvement by Tumer and Ghosh (1999).

3 Application : Napa Valley Wine Quality Score data

To test whether our proposed ensemble algorithms really work and to see how much improvement can be made, we conduct an empirical study. For this empirical study we use Napa Valley Wine Quality Score data [1].

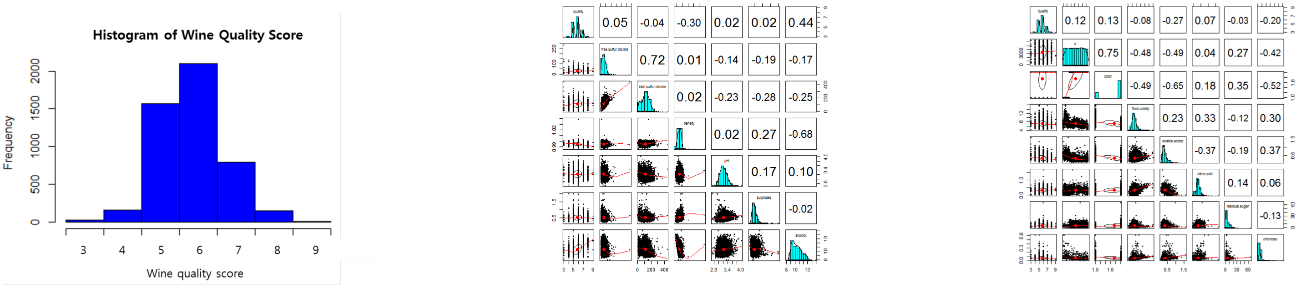
3.1 Purpose

A wine rating is a score assigned by one or more professional wine critics to a wine tasted as a summary of that critic's evaluation of that wine. This wine quality score is assigned after the wines have been released to the market. Since it directly affects the price and demand of the wine, for the wine companies predicting the wine quality score is very important.
We want to predict the wine quality score with given soil conditions of the year in which the wine was produced.

3.2 Data Description

- We have 4800 (complete) data points.
- Wine Quality Score is integer-valued, and ranges from 3 to 9.
- There are 13 predictors : X (labeling information), color, fixed.acidity, volatile.acidity, citric.acid, residual.sugar, chlorides, free.sulfur.dioxide, total.sulfur.dioxide, density, pH, sulphates, alcohol
- The training data is **imbalanced** ; that is, there are major (5,6,7) labels that appear most of the time and minor (3,4,8,9) labels.

	X	Color	Fixed acidity	Volatile acidity	Citric acid	Residual sugar	Cl^-	Free SO_2	Total SO_2	Density	PH	$SO_4{}^{2-}$	Alcohol
Min.	2	Red:	4.2	0.08	0	0.6	0.01	1	6	0.987	2.72	0.22	8.0
Median	3234	1189	7.0	0.29	0.31	3.0	0.05	29	118	0.995	3.20	0.50	10.3
Mean	3241	While:	7.2	0.34	0.32	5.4	0.06	31	115	0.994	3.22	0.53	10.5
Max.	6497	3611	15.9	1.58	1.66	65.8	0.61	289	440	1.039	4.01	2.00	14.9



(a) Histogram of wine quality score

(b) Correlation matrix

(c) Correlation matrix

Figure 1: Basic data description

4 Ensemble Methodology

4.1 Single Model Methods and Basic Analysis

We fit the following prediction models to our training data set :
1. Ordinary Least Square (OLS) Regression
2. Robust Linear Regression - M-estimation(RLM), Least Trimmed Squares (LTS)
3. Weighted Least Square (WLS) Regression - weights : reciprocal of the square root of the size of each score
4. L1 regression : backfitting + WLS (L1R)

[1] This dataset is given by Prof. T. Hastie and originally used in STATS 315A class. In this simulation, to keep confidentiality of the original data set, we only use a part of training set of the original dataset.

5. Variable Selection - Forward Selection (FWD), Backward Selection (BWD), All-subset Selection (ALL)
6. Penalized Least Squares - Ridge, Lasso, Elastic Net
7. Principle Component Regression (PCR)
8. Partial Least Squares (PLS) Regression
9. Basis Expansions and Regularization - Natural Cubic Splines with Lasso (NCS)
10. Local Polynomial Regression (LPR) with selected variables from 5
11. Support Vector Machine (SVM) - Linear Kernel, Polynomial Kernel, Radial Kernel
12. Linear Discriminant Analysis (LDA)
13. Logistic Regression (LOG)
14. K-nearest neighbor (K-nn)

Note the following details of these methods :

- Regression models, from 1 to 10, give real-valued scores, which are then rounded off to be integers.
- In the WLS Regression model, weights are chosen to avoid the phenomenon that a model mostly predicts major scores due to not predictors but its criterion.
- SVM with k labels is the one-against-one-approach, in which k(k-1)/2 binary classifiers are trained by SVM with 2 labels and majority vote determines the label of a data point.

We conducted 5-fold CVs for different learning methods to estimate the generalization errors, using 4500 data points randomly chosen from the training data set(the remaining 300 data points is set to be the test set).

Table 1: RMSEs for different learning methods

	L1R	ElaNet	PCR	NCS	LPR	SVM	LDA	LOG
CV	0.731	0.731	0.769	0.792	0.724	0.717	0.726	1.501
TEST	0.802	0.792	0.856	0.936	0.806	0.723	0.781	1.343

Note that 8 methods were chosen to be representatives of redundant methods and independent of each other. Also, SVM denotes SVM with the radial kernel from now on.

From Table 1, we can see that all methods show mediocre performance, although classification models are slightly better than regressions. If we can pick up models which are good to predict the major labels and the others which is good to predict the minor labels and ensemble them, it may contribute to the overall performance improvement.

4.2 Ensemble Algorithms

First we construct a simple heuristic ensemble classifier using the idea described above. Let's look at the per class(score) misclassification rates for typical methodologies selected above.

Table 2: Per class misclassification rates

Score	L1R	ElaNet	PCR	NCS	LPR	SVM	LDA	LOG
3	100	100	100	95.5	100	100	**77.3**	100
4	98.8	98.8	98.8	100	98.2	94.5	**87.7**	98.8
5	46.8	47.6	47.9	43.7	44.6	**42.2**	**41.5**	48.7
6	25.0	23.6	23.7	28.1	26.1	**20.3**	31.2	35.1
7	74.3	75.2	75.3	75.5	75.5	**55.0**	69.7	78.3
8	100	100	100	100	100	**73.1**	99.3	91.4
9	100	100	100	100	100	100	100	100

From this table, we can observe that

- SVM outperforms the others for major classes, and it is the only classifier which successfully distinguishes some of the class 8 data points.
- LDA classifier performs much better job for class 3 and 4 than regressions and SVMs did.

Based on these observations, we decided to combine three classifiers SVM, LDA, and elastic-net (which is included because it may have predictive power related to regression settings, although being dominated by SVM) in the way that strengthens the strengths and makes up for the weaknesses ; that is, put the weights decreasing in the order of SVM, LDA, and elastic-net. When choosing weights under this scheme, we did several experiments with different combinations of the weights and empirically chose the best model among them without using any formal optimization over the training data. The prediction rule obtained from our heuristics is as follows :

Algorithm 1 : Ensemble Method based on simple heuristics (Heuristic)

$$\textit{Prediction = 0.7 SVM + 0.25 LDA + 0.05 ElaNet}$$

Next, we discuss a reasonable process to find the optimal weight vector more systematically. Suppose that we have a finite set of classes $\{1, \cdots, C\}$ and learning methods $L_1, \cdots, L_M$ for classifying these classes (using the given predictors and data set), and use K-fold CV. Let N denote the number of observations so with the K-fold CV it follows that $N = \sum_{k=1}^{K} n_k$, where in our data $n_k = 900$ for all $k = 1, ..., K$. For each (k, m), let $y^{(k)}$ and $\hat{y}^{(-k,m)}$ denote the realized value of classes for k^{th} CV fold and the predicted value of classes for k^{th} CV fold based on the learning method L_m trained on the rest of the data, i.e. trained on the data points taken out for the k^{th} CV fold, respectively. Finally let the matrix $X^{(k)}$ denote $\left[\hat{y}^{(-k,1)}\,\hat{y}^{(-k,2)}\,\cdots\,\hat{y}^{(-k,M)}\right]$. Then we obtain a $n_k \times 1$ vector $y^{(k)}$, and $n_k \times M$ matrix $X^{(k)}$ for our analysis. Now consider the following algorithm.

Algorithm 2 : Ensemble Method based on l_2 Minimization of CV Errors (L2CV)

With given constructions $\{X^{(k)}\}_k$ and $\{y^{(k)}\}_k$, we find the $M \times 1$ optimal ensemble weight vector $\boldsymbol{\omega}^*$ as

$$\underset{\boldsymbol{\omega}\in\mathbb{R}^M}{\operatorname{argmin}} \sum_{k=1}^{K} ||y^{(k)} - X^{(k)}\boldsymbol{\omega}||_2^2 = \underset{\boldsymbol{\omega}\in\mathbb{R}^M}{\operatorname{argmin}}\ \boldsymbol{\omega}^T \sum_{k=1}^{K} {X^{(k)}}^T X^{(k)}\boldsymbol{\omega} - 2\sum_{k=1}^{K} {y^{(k)}}^T X^{(k)}\boldsymbol{\omega}\ s.t.\ \mathbf{1^T}\boldsymbol{\omega} = 1, \boldsymbol{\omega} \geq 0$$

It can easily be shown that **Algorithm 2** has lower CV RMSE (without rounding predictions, which is necessary for the comparison and expected to have a little effect on CV RMSE) than any of single learning methods, by letting $\boldsymbol{\omega}$ be columns of the $M \times M$ identity matrix. From this algorithm, we have the following ensemble:

$$\text{Prediction} = 0.092\ \text{NS} + 0.170\ \text{LPR} + 0.507\ \text{SVM} + 0.231\ \text{LDA}$$

Now recall that we used misclassification rates per class to choose learning methods and give them weights. Similarly, we could instead use Squared Errors Per Class(SSEPC), to be consistent with RMSE. Here we propose 2 automatic ensemble methods based **only** on SSEPC. The first one is a modification of Algorithm 2.

Algorithm 3 : Simple SSEPC Ensemble (SSEPC)

For each (k, m, c), let $s_{cm}^{(k)} := \sqrt{\sum_{i:y_i^{(-k)}=c}(\hat{y}_i^{(-k,m)} - y_i^{(-k)})^2}$ and $S^{(k)} := (s_{cm}^{(k)})_{1\leq c\leq C, 1\leq m\leq M} \in \mathbb{R}^{C\times M}$. Simple SSEPC Ensemble is defined by $\sum_{m=1}^{M} \alpha_m^* M_m$, where α^* is the solution of the quadratic programming problem

$$\min_{\alpha\in\mathbb{R}^M} \alpha^T \sum_{k=1}^{K} S^{(k)T} S^{(k)}\alpha\ \ s.t.\ \ \alpha \geq 0, 1^T\alpha = 1$$

This optimizes an upper bound of the CV RMSE(without rounding), and has the same property as Algorithm 2 :

The CV RMSE of the simple SSEPC ensemble is smaller than those of $L_1, \cdots, L_M$,

To see these, let's consider the k^{th} CV fold, and omit superscripts k and $-k$ from y's for simplicity. We can then obtain an upper bound of SSE (from the k^{th} CV fold) of an ensemble method $\sum_{m=1}^{M} \alpha_m L_m$ as follows :

$$\begin{aligned} SSE_k &:= \sum_i (\sum_{m=1}^{M} \alpha_m \hat{y}_i^{(m)} - y_i)^2 = \sum_i [\sum_{m=1}^{M} \alpha_m^2 (\hat{y}_i^{(m)} - y_i)^2 + 2 \sum_{1\leq l<m\leq M} \alpha_l \alpha_m (\hat{y}_i^{(l)} - y_i)(\hat{y}_i^{(m)} - y_i)] \\ &= \sum_{c=1}^{C} [\sum_{m=1}^{M} \alpha_m^2 \sum_{i:y_i=c} (\hat{y}_i^{(m)} - y_i)^2 + 2 \sum_{1\leq l<m\leq M} \alpha_l \alpha_m \sum_{i:y_i=c} (\hat{y}_i^{(l)} - y_i)(\hat{y}_i^{(m)} - y_i)] \\ &\leq \sum_{c=1}^{C} [\sum_{m=1}^{M} \alpha_m^2 {s_{cm}^{(k)}}^2 + 2 \sum_{1\leq l<m\leq M} \alpha_l \alpha_m s_{cl}^{(k)} s_{cm}^{(k)}] = \sum_{c=1}^{C} [\sum_{m=1}^{M} \alpha_m s_{cm}^{(k)}]^2 = \|S^{(k)}\alpha\|_2^2 \end{aligned}$$

where the inequality is due to the Cauchy-Schwarz inequality, which makes the upper bound a function of $S^{(k)}$. Summing up these upper bounds for $k = 1, \cdots, K$ gives $SSE := \sum_{k=1}^{K} SSE_k \leq \sum_{k=1}^{K} \|S^{(k)}\alpha\|_2^2 = \alpha^T \sum_{k=1}^{K} S^{(k)T} S^{(k)} \alpha \ \cdots(1)$. Furthurmore, note that $\sum_{k=1}^{K} \|S^{(k)}\alpha\|_2^2 = \sum_{k=1}^{K} \sum_{c=1}^{C} {s_{cm}^{(k)}}^2$ is the SSE of L_m when $\alpha_m = 1, \alpha_l = 0 \ \forall l \neq m$, i.e., $\alpha^{*T} \sum_{k=1}^{K} S^{(k)T} S^{(k)} \alpha^* \leq (SSE\text{'s of } L_1, \cdots, L_M) \ \cdots(2)$. Combining (1) and (2) results in the claim above. Applying **Algorithm 3** to our data resulted in the following sparse ensemble

$$\text{Prediction} = 0.050 \text{ LPR} + 0.950 \text{ SVM}$$

Finally, we revisited the first heuristic idea and came up with the following algorithm.

Algorithm 4 : Forward Stepwise SSEPC Ensemble (FWD SSEPC)

Let R be the $C \times M$ matrix of CV SSE per class (hence its rows are classes).

(1) Define the best single model as the current model, and update R by removing the corresponding column.

(2) For each column of R, divide each element by the corresponding CV SSE per class of the current model, to obtain the matrix of those ratios.

(3) If all of those ratios are larger than 1, stop ; otherwise, find the column with the smallest sum of ratios, include the corresponding single model to the current set of models, and update the model by using Algorithm 2 of the new set of models and R by removing the corresponding column.

(4) Repeat 2 and 3 until convergence (in terms of the supremum of the change in weight over all single models)

Let's discuss the **ideas** of step 2 and 3. First of all, the smaller the ratio of the SSE of a class between a single model and the current model is, the better the single model works in the corresponding class. Therefore, summing up those ratios over all classes could be considered as a measure of how much a single model complements the current ensemble model, treating all classes equally. Furthermore, if those ratios are greater than 1 for all classes, than we don't need to include the corresponding single model to the ensemble.

Implementing this algorithm gave the ensemble

$$\text{Prediction} = 0.520 \text{ SVM} + 0.373 \text{ LDA} + 0.107 \text{ L1R}$$

5 Results and Discussion

Below are the experiment results of 4 algorithms we discussed so far.

Table 3: RMSEs for Ensemble Methods

	Heuristic	L2CV	SSEPC	FWD SSEPC
CV Set	0.679	0.669	0.709	0.673
Test Set	0.683	0.687	0.718	0.687

By comparing Table 3 with Table 1, it is found that

1. All four ensemble methods perform better than any of single classifiers in terms of RMSE.
2. Ensemble methods show their strengths particularly when they combine the complementary predictive powers of multiple models.
3. In particular, we can observe that the background ideas of FWD SSEPC worked in the sense that LDA was picked following SVM as done in the heuristic ensemble, although the next step actually increased the RMSE from 0.688 to 0.695.

6 Future Work

- From the results, L2CV and FWD SSEPC showed similar performances. Therefore, we can further compare them via experiments or theoretical analysis. Also, it can be examined if simple SSEPC always guarantees sparse ensemble models, as shown in our experiment.
- To improve FWD SSEPC, we can try to develop a measure for the third step as a function of the ratios of SSEs, other than the one we used before.

7 Reference

1. Ali K. M., Pazzani M. J., Error Reduction through Learning Multiple Descriptions, Machine Learning, 24: 3, 173-202, 1996.
2. Bartlett P. and Shawe-Taylor J., Generalization Performance of Support Vector Machines.
and Other Pattern Classifiers, In Advances in Kernel Methods, Support Vector Learning, MIT Press, Cambridge, USA, 1998.
3. Opitz, D. and Maclin, R., Popular Ensemble Methods: An Empirical Study, Journal of Artificial Research, 11: 169-198, 1999.
4. Mitchell, T., Machine Learning, McGraw-Hill, 1997.
5. Dietterich T., Ensemble methods in machine learning. In J. Kittler and F. Roll, editors, First International-Workshop on Multiple Classifier Systems, Lecture Notes in Computer Science, pages 1-15. Springer-Verlag, 2000
6. Dietterich, T.G., Machine learning research: Four current directions. AI Magazine 18(4) (1997) 97136.
7. Rokach, L. and Maimon, O., Clustering methods, Data Mining and Knowledge Discovery Handbook, pp. 321352, 2005, Springer.
8. Rokach, L., Decomposition methodology for classification tasks: a meta decomposer framework, Pattern Analysis and Applications, 9(2006):257271.
9. Rokach, L., Ensemble Methods in Supervised Learning. Springer (2010).
10. Tumer. K. and Ghosh. J., Error correlation and error reduction in ensemble classifiers. Connection Science (1997), 8(3-4), 385-404.
11. Bennett, K. P., Demiriz, A., Maclin, R., Exploiting unlabeled data in ensemble methods, KDD '02 Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, 289-296.
12. Rokach L., Genetic algorithm-based feature set partitioning for classification problems, Pattern Recognition, 41(5):16761700, 2008.
13. Geman S., Bienenstock, E., and Doursat, R., Neural networks and the bias variance dilemma. Neural Computation, 4:1-58, 1995.
14. Tan A. C., Gilbert D., Deville Y., Multi-class Protein Fold Classification using a New Ensemble Machine Learning Approach. Genome Informatics, 14:206217, 2003.
15. Quinlan, J. R., C4.5: Programs for Machine Learning, Morgan Kaufmann, Los Altos, 1993.
16. Quinlan, J. R., Bagging, Boosting, and C4.5. In Proceedings of the Thirteenth National Conference on Artificial Intelligence, pages 725-730, 1996.
17. Sohn S. Y., Choi, H., Ensemble based on Data Envelopment Analysis, ECML Meta Learning workshop, Sep. 4, 2001.