MSBD 6000B Individual Project 1

KWAN Hok Ming 20386486

Introduction

There are three files for this project: traindata, testdata and trainlabel. The training data contains 3220 rows with 56 quantitative features without null while the testing data contains 1380 rows without null also. The trainlabel contains labels 1 and 0 for corresponding rows of training data.

Task

The classification models are utilized to predict the labels and the accuracies of models are tried to improve.

Data Processing

The training data is normalized before training the models.

Models

- **K-nearest neighbors algorithm** (**k-NN**)
  K-nearest neighbors is a non-parametric method used for classification and regression [1], so there are no assumptions about the shape of the decision boundary. The K training observations that are closest to x are identified to make a prediction for an observation X = x. Then X is assigned to the class to which the majority of these observations belong.

- **Random Forest**
  Random forest is an ensemble of classification trees for classification [2]. Each of the decision trees is built using a bootstrap sample of the data, and at each split the candidate set of variables is a random subset of the variables. Random forest uses both bootstrap aggregation for combining unstable learners [3], and random variable selection for tree building. Moreover, random forests improve the overfitting problem of decision tree to the training set.
  For example, given a training set $X = x_1, x_2 \ldots x_n$ with responses $Y = y, y_2 \ldots y_n$, bagging repeatedly (N times) selects a random sample with replacement of the training set. After training, predictions for unseen samples x' can be made by averaging the predictions from all the individual regression trees on x': $\hat{f} = \frac{1}{N}\sum_{n=1}^{N} f_n(x')$
  Advantage of random forest is that it can handle large number of training examples without features deletion and maintain good accuracy when handling very high dimensional spaces. However, it may make the algorithm slow, if there is a large number of features.

- **Naive Bayes Classifier (Gaussian)**
  Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes’ theorem with the assumption of independence between every pair of features.

Moreover, naive Bayes classifiers have worked quite well in many real-world situations, famously document classification and spam filtering.
It is assumed that the continuous values associated with each class are distributed according to a Gaussian distribution while dealing with continuous data.
The probability distribution of v given a class c can be calculated as below:

$$p(x = v|c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} e - \frac{(v - \mu_c)^2}{2\sigma_c^2}$$

Advantage of Naive Bayes Classifier is that if the NB conditional independence assumption holds, then it will converge quick. Therefore, less training data will be used. However, if any two features are not independent, the result can be potentially not very accurate.

- **Adaptive Boosting (AdaBoost)**
  The AdaBoost is to fit a sequence of weak learners, for example: small decision trees, on repeatedly modified versions of the data. The predictions from all weak learners are then combined by a weighted vote to produce the final prediction. The AdaBoost training process selects those features which could improve the predictive power of the model. Then it could reduce dimensionality and improve execution time as irrelevant features do not need to be computed.

- **Extra Trees Classifier**
  The Extra-Trees classifier is similar with the random forest, but it splits nodes by choosing cut-points fully at random and that it uses the whole learning sample (rather than a bootstrap replica) to grow the trees [4].

- **Gradient Boosting**
  Similar with other boosting methods, gradient boosting combines weak learners into a single strong learner in an iterative fashion, but it allows optimization of an arbitrary differentiable loss function.

- **Extreme Gradient Boosting**
  Extreme gradient boosting is built on the principles of gradient boosting framework, but it uses a more regularized model formalization to control over-fitting, which gives it better performance according to the author of the algorithm, Tianqi Chen [5]. Also, it also leverages the structure of your hardware to speed up computing times and facilitate memory usage.

- **Support Vector Machine**
  Support Vector Machines (SVM) are popular classifiers in many areas. SVM (with linear kernel, as we used for this project) try to find an optimal separating hyperplane between the classes. When the classes are linearly separable, the hyperplane is located so that it has maximal margin which should lead to better performance on data.
  For example, the hyperplanes can be described by the equations: $w \cdot x - b = 1$ and $w \cdot x - b = -1$. We try to maximize the distance between the plances, $\frac{2}{\|w\|}$, so we need to minimize $\|w\|$. Therefore, it is an optimiztion problem as followings:
  Minimize$\|w\|$ subject to $y_i(w \cdot x - b) \geq 1$ for i =1,…,n.

One of the advantages of SVM is that it is defined by a convex optimization problem as shown above (no local minima) for which there is an efficient method. However, the disadvantage is that the theory only really covers the determination of the parameters for a given value of the regularization and kernel parameters and choice of kernel. Kernel models can be quite sensitive to over-fitting the model selection criterion [6].

- **Linear discriminant analysis**
  Linear discriminant analysis (LDA) is a generalization of Fisher's linear discriminant, a method used in statistics, pattern recognition and machine learning to find a linear combination of features that characterizes or separates two or more classes of objects or events. The resulting combination may be used as a linear classifier, or, more commonly, for dimensionality reduction before later classification.

- **Quadratic Discriminant Analysis**
  QDA is preferred if decision boundary between two classes is assumed to be a quadratic decision boundary and can accurately model a wider range of problems than can the linear methods. Moreover, QDA assumes that each class has its own covariance matrix [8]. Due to the assumptions, QDA can perform better if the training set is very large, so that the variance of the classifier is not a major concern.

- **Logistic Regression**
  Statistician David Cox developed logistic regression in 1958[7]. Logistic regression can be treated as a special case of the generalized linear model. The binary logistic model is used to estimate the probability of a binary response based on one or more independent variables. The logistic regression can be realized simply as finding the β parameters that best fit:

  $$y = \begin{cases} 1 \; \beta_0 + \beta_1 + \epsilon > 0 \\ 0 \; else \end{cases}$$

  where ϵ is an error distributed by the standard logistic distribution.
  Logistic Regression does not assume a linear relationship between the features and dependent variable and the dependent variables need not be normally distributed.

Features Selection

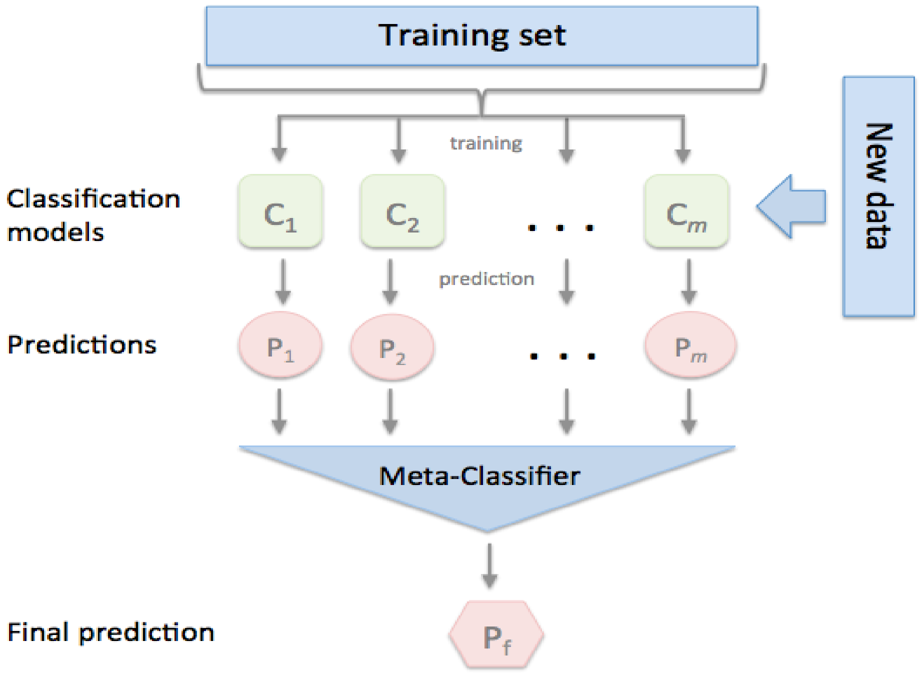
Having too many either irrelevant or redundant features in training data can decrease the accuracy of the models. Feature selection which is a process to select those features in the training data that contribute most to the prediction reduces overfitting [9], improve accuracy and shorter training time.
Models that use ensembles of decision trees, for example, Random Forest or Extra Trees, can compute the relative importance of each feature. Hence, for those ensemble models, the features importance is calculated and features are selected if the importance is larger than 0.02. For other models, whole training data is used for training.

Hyperparameter Tuning

The tuning set is separated from training set. 30% of training data is used for tunning the hyperparameters.

Meta Classifier

Stacking is used to improve the prediciton because stacking typically yields performance better than any single one of the trained models [10]. Stacking combines the predictions of several other learning algorithms. First, all of the other algorithms are trained using the training data, then a combiner algorithm is trained to make a final prediction using all the predictions of the other algorithms with new training data as additional inputs. The meta classifier of this project is logistic regression. Stratified k-fold cross-validation is also used. This means that each fold contains roughly the same proportions of the two types of class labels and the mean response value is approximately equal in all the folds.



Meta Model Validation

Area under the ROC curve is used for optimizing our result. The best result of the tuning set is 0.978624208304.

Prediction

After tuning the meta model, the metal model is used to predict the testing data. The result is saved as a csv file.

Conclusion

Combining predictions from multiple uncorrelated classification models via a meta-classifier can outperform other individual classification models and increase accuracy, even though each individual classifier has over 0.9 accuracy.

References:

[1] An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression
N. S. Altman, pp. 175-185 | Received 01 Feb 1990, Published online: 27 Feb 2012

[2] Breiman L, Friedman J, Olshen R, Stone C: Classification and regression trees. New York: Chapman & Hall; 1984.

[3] Hastie T, Tibshirani R, Friedman J: The elements of statistical learning. New York: Springer; 2001.

[4] Pierre Geurts, Damien Ernst, Louis Wehenkel: Extremely randomized trees, 2006. http://orbi.ulg.ac.be/bitstream/2268/9357/1/geurts-mlj-advance.pdf

[5] https://www.quora.com/What-is-the-difference-between-the-R-gbm-gradient-boosting-machine-and-xgboost-extreme-gradient-boosting

[6] G. C. Cawley and N. L. C. Talbot, Over-fitting in model selection and subsequent selection bias in performance evaluation, Journal of Machine Learning Research, 2010. Research, vol. 11, pp. 2079-2107, July 2010.

[7] Quadratic Discriminant Analysis Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani, "An Introduction to Statistical Learning with Applications in R" Pages 149- 150, 2013.

[8] Cox, DR (1958). "The regression analysis of binary sequences (with discussion)". J Roy Stat Soc B. 20: 215–242.

[9] Bermingham, Mairead L.; Pong-Wong, Ricardo; Spiliopoulou, Athina; Hayward, Caroline; Rudan, Igor; Campbell, Harry; Wright, Alan F.; Wilson, James F.; Agakov, Felix; Navarro, Pau; Haley, Chris S. (2015). "Application of high-dimensional feature selection: evaluation for genomic prediction in man"

[10] Rokach, L. (2010). "Ensemble-based classifiers". Artificial Intelligence Review. 33 (1-2): 1–39.