Use of classifier stacking to predict a biological response

Cheng-Bin Ren

Zhejiang University, Hangzhou 310027, China

**Abstract.** In the age of Big Data, the usage of data is more and more often in all kinds of industries.

1. Introduction

Almost two decades ago, the information flow in the pharmaceutical industry was relatively simple and the application of technology was limited. However, as we progress into a more integrated world where technology has become an integral part of the business processes, the process of transfer of information has become more complicated. Today increasingly technology is being used to help the pharmaceutical firms manage their inventories and to develop new product and services.

Data mining fondly called patterns analysis on large sets of data uses tools like association, clustering, segmentation and classification for helping better manipulation of the data help the pharma firms compete on lower costs while improving the quality of drug discovery and delivery methods. A deep understanding of the knowledge hidden in the Pharma data is vital to a firm’s competitive position and organizational decision-making.

Pharma companies think that drugs might have some yielded benefits. The drug undergoes testing in animals and human tissue to observe effect and determines how much drug to consume for desired effect or how dangerous is the drug. The Data mining techniques can be here used is classification and neural networks. The goal here is to predict if treatment will aid patients. Because if drug will not aid patients, what purpose does drug serve. Predicting the drug behavior is essential when we have data supporting use of drug and also have training data that shows effects of drug (positive or negative). The test should be able to predict which patients will benefit and which treatment help sickle cell anemia patients. The information like gender, body weight, disease state, etc will play crucial role. This crucial data should be fed into neural network and predict whether patient will benefit from drug. Only one of two classifications yes/no will be available on training data. Network is trained for the yes classifications and a snapshot is taken of the neural network. Then network is trained for the no classifications and another snapshot is taken. The output is yes or no, depending on whether the inputs are more similar to the yes or the no training data.

1. Related Work
   1. SVM

In machine learning, support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. The basic SVM takes a set of input data and predicts, for each given input, which of two possible classes forms the output, making it a non-probabilistic binary linear classifier. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

* 1. Random Forrest

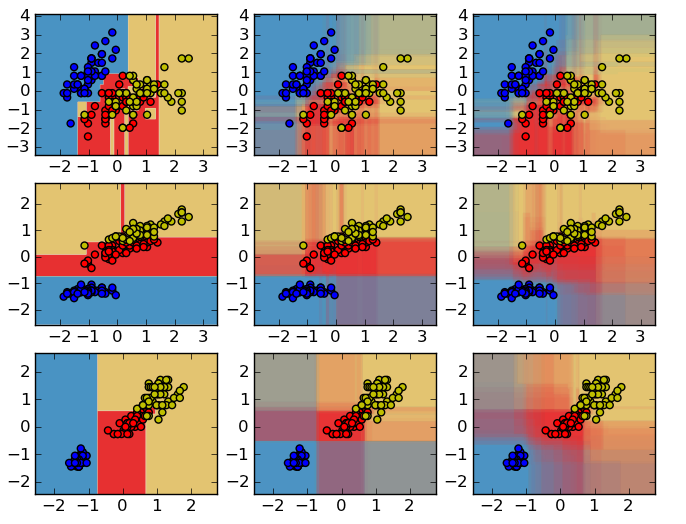
Random Forrest has been proposed by Breiman (2001) as an enhancement of Tree Bagging. To build a tree it uses a bootstrap replica of the learning sample, and the CART algorithm (without pruning) together with the modification used in the Random Subspace method. At each test node the optimal split is derived by searching a random subset of size K of candidate attributes (selected without replacement from the candidate attributes). Empirical studies have shown that Random Forests significantly outperform Tree Bagging and other random tree ensemble methods in terms of accuracy. In terms of degree of randomization, this algorithm is stronger than Tree Bagging, specially, if K is small compared to the number of attributes, n. It is also stronger than Random Subspace since it combines this method with bootstrap sampling. Randomization is both implicit (attribute and cut-point) and explicit (attribute). We use the notation RFK, with K = d for the default setting, and K =∗ for the best result over the range K = 1, … , n.

In Random Forrest, each tree in the ensemble is built from a sample drawn with replacement (i.e., a bootstrap sample) from the training set. In addition, when splitting a node during the construction of the tree, the split that is chosen is no longer the best split among all features. Instead, the split that is picked is the best split among a random subset of the features. As a result of this randomness, the bias of the forest usually slightly increases (with respect to the bias of a single non-random tree) but, due to averaging, its variance also decreases, usually more than compensating for the increase in bias, hence yielding an overall better model.

* 1. Extremely Randomized Trees

Extremely Randomized Tress is a tree-based ensemble method for supervised classification and regression problems. It essentially consists of randomizing strongly both attribute and cut-point choice while splitting a tree node. In the extreme case, it builds totally randomized trees whose structures are independent of the output values of the learning sample. The strength of the randomization can be tuned to problem specifics by the appropriate choice of a parameter.

In extremely randomized trees, randomness goes one step further in the way splits are computed. As in random forests, a random subset of candidate features is used, but instead of looking for the most discriminative thresholds, thresholds are drawn at random for each candidate feature and the best of these randomly-generated thresholds is picked as the splitting rule. This usually allows to reduce the variance of the model a bit more, at the expense of a slightly greater increase in bias.



**Fig. 1.** Decision surfaces of a decision tree, of a random forest and of an extra-trees classifier

* 1. Gradient Tree Boosting

Gradient Tree Boosting or Gradient Boosted Regression Trees (GBRT) is a generalization of boosting to arbitrary differentiable loss functions. GBRT is an accurate and effective off-the-shelf procedure that can be used for both regression and classification problems. Gradient Tree Boosting models are used in a variety of areas including Web search ranking and ecology.

The advantages of GBRT are:

* Natural handling of data of mixed type (= heterogeneous features)
* Predictive power
* Robustness to outliers in input space (via robust loss functions)

The disadvantages of GBRT are:

* Scalability, due to the sequential nature of boosting it can hardly be parallelized.
  1. Classifier Stacking

In statistics and machine learning, ensemble methods use multiple models to obtain better predictive performance than could be obtained from any of the constituent models. Unlike a statistical ensemble in statistical mechanics, which is usually infinite, a machine learning ensemble refers only to a concrete finite set of alternative models.

The crucial prior belief underlying the scientific method is that one can judge among a set of models by comparing them on data that was not used to create any of them. This same prior belief underlies the use in machine learning of bake-off contests to judge which of a set of competitor learning algorithms actually the best fit in selected domains is.

This prior belief can also be used by a single practitioner, to choose among a set of models based on a single data set. This is done by partitioning the data set into a held-in data set and a held-out data set; training the models on the held-in data; and then choosing whichever of those trained models performs best on the held-out data. This is the cross-validation technique, mentioned above.

Stacking (sometimes called stacked generalization) exploits this prior belief further. It does this by using performance on the held-out data to combine the models rather than choose among them, thereby typically getting performance better than any single one of the trained models. It has been successfully used on both supervised learning tasks (regression) and unsupervised learning (density estimation). It has also been used to estimate bagging's error rate.

Because the prior belief concerning held-out data is so powerful, stacking often out-performs Bayesian model-averaging. Indeed, renamed blending, stacking was extensively used in the two top performers in the recent Netflix competition.

1. Methodology
   1. Datasets

The data is provided by Boehringer Ingelheim for research. The data is in the comma separated values (CSV) format. Each row in this data set represents a molecule. The first column contains experimental data describing an actual biological response; the molecule was seen to elicit this response (1), or not (0). The remaining columns represent molecular descriptors (d1 through d1776), these are calculated properties that can capture some of the characteristics of the molecule - for example size, shape, or elemental constitution. The descriptor matrix has been normalized.

* 1. Choice of Stacking Classifiers

### **.** Random Forrest

In contrast to the original publication, in this research, the implementation combines classifiers by averaging their probabilistic prediction, instead of letting each classifier vote for a single class.

* 1. Process of Classifier Stacking

The process for Classifier Stacking in this paper is:

1. Choose multi classifiers
2. Take each of them and perform Cross-Validation on a training set
3. For each classifier, collect predictions from each folder of Cross-Validation into a blender training set: Blend\_train
4. Train each classifier on a full training set and collect predictions into a blender test set: Blend\_test
5. Train a blender on Blend\_train (Linear model)
6. Get predictions for Blend\_test.
   1. Evaluation Method

Predicted probabilities that a molecule elicits a response are evaluated using the log loss metric.

Log loss is defined as:

,

where N is the number of samples, log is the natural logarithm, is the posterior probability that the sample elicited a response, and is the ground truth ( means the molecule elicited a response, means that it did not).

* 1. Experiments

1. Results

**References**