Ensemble Learning and the Heritage Health Prize

Jonathan Stroud, Igii Enverga, Tiffany Silverstein,
Brian Song, and Taylor Rogers
Advisors: Dr. Max Welling, Dr. Alexander Ihler, Sungjin Ahn, and Qiang Liu
University of California, Irvine

August 17, 2012

Abstract

The Heritage Health Prize is awarded to the team that develops the most accurate model for predicting how long a person will spend in the hospital during the next year, given historical medical data. A successful prediction algorithm could revolutionize the health care industry, as it could preemptively identify patients with a high risk of hospital admission and reduce the number of unnecessary hospitalizations. In this work, we examine the applications of commonly used machine learning algorithms to this problem and compare their individual accuracy to that of an ensembling algorithm that combines all models. We conclude that an ensemble is significantly more effective than any individual model, but is ultimately not accurate enough to win the competition.

1 Introduction

The Heritage Health Prize is a $3 million reward for the team which can best "identify patients who will be admitted to a hospital within the next year, using historical claims data." [1] The purpose of this competition is apparent when considering that over $30 billion was spent on unnecessary hospital admissions alone in 2006. An accurate model would allow health care providers to administer more personalized care, thereby decreasing both these unnecessary hospital admissions and medical spending as a whole.

The prize is hosted by Kaggle, a website where teams of researchers tackle machine learning problems in a competitive environment. Kaggle provides relevant datasets as well as quantitative feedback for predictions made. The team that produces the most accurate model within the time frame wins the competition and is typically compensated in return for a description of their method.

Previous research into the Heritage Health Prize data have highlighted a number of predictors, each with varying levels of accuracy. The predictors that we will focus on in this paper include:

- K-Nearest Neighbours
- Logistic Regression
- Support Vector Regression

- Random Forests
- Gradient Boosting Machines
- Neural Networks

Similar problems, such as the Netflix Prize, have been solved through a technique called blending, in which several algorithms are combined to create a superior predictor. We apply this approach to the Heritage Health Prize problem, ensembling the predictors listed above, which we will detail in later sections of this paper.

2 Problem Statement

The Heritage Health Prize provides several tables of anonymized patient information. The data spans three years of hospital records. However, the number of days spent in the hospital by each patient is only provided for two of the three years. The job of the predictor is to determine how long the patients spent in the hospital in the third year.

The tables provided by the Heritage Health Prize include data regarding insurance claims, prescriptions, lab results, primary conditions, and other relevant information. In order to facilitate the prediction models, it is necessary to reduce this data into one consistent dataset. We use the method provided by the team, Market Makers, in their milestone one paper in order to accomplish this task. [5] More about this method will be explained in later sections.

Predictions are evaluated using root mean squared logarithmic error, referred to henceforth as RMSLE.

$$\varepsilon = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(log(p_i+1) - log(a_i+1)^2)} \tag{1}$$

Where:

- i is a patient's unique MemberID
- n is the total number of patients
- p_i is the prediction made for patient i
- a_i is the actual number of days spent in the hospital by patient i

These ratings are made public and used to place teams on a global leaderboard. When Kaggle calculates the leaderboard positions between milestones, only 30 percent of the test data is used. The RMSLE calculated on the entire test set is not revealed until the end of the competition.

A maximum accuracy threshold has been set at $\varepsilon = 0.4$. In order for a team to win the grand prize, their predictor must have an RMSLE below this threshold. So far, no predictor has obtained a score this low. In fact, the predictor with the lowest RMSLE at the time of writing obtained a score of greater than 0.45, and the majority of leaderboard scores range between this and 0.48 [2].

The competition began on April 4th, 2011, and ends on April 3, 2013, with several milestone prizes awarded between. Teams are limited to one submission per day, thus preventing abuse of the feedback system in order to climb the leaderboard.

3 Idea

In 2009, online DVD rental company Netflix offered a $1 Million prize for the first team that improved the accuracy of their movie recommendation system by 10%. The competition was quite successful and inspired many other data science competitions, such as those featured on Kaggle, including the Heritage Health Prize. The winner of the Netflix competition used an ensemble of several methods, which individually were not as successful as the ensemble as a whole.

The fundamental idea behind this technique is that the collective knowledge of a group is greater than that of any individual member. This is commonly referred to as "Wisdom of the Crowd." This type of predictor has been studied for over a century. Its effectiveness was observed in 1906 when Francis Galton noticed at a fair that the mean prediction from a guess-the-weight competition was more accurate than any one individual's prediction [3].

While simply predicting the mean of all predictions may be quite effective, we can use the feedback obtained from Kaggle to improve our blending algorithm significantly. The RMSLE indicates how accurate a predictor is, which makes it possible for us to assign weights to each predictor. This should give us an even more accurate ensembled predictor.

Using weighted averages to combine predictors is not new. One example of this approach being used is in prediction markets such as Intrade, in which anyone can 'bet' on an event occurring by buying or selling virtual shares of stock in that event. If a stock is valued very highly, many people believe the event will occur. This has been shown to be very accurate in predicting the results of elections [6].

Because of the accessibility of feedback on predictors from Kaggle and the proven success of ensembling methods, we proposed the idea of applying the same ensembling method used in the Netflix Prize to the Heritage Health Prize. This method involves finding an optimal weight vector for the set of classifiers through optimization.

4 Details

4.1 Data Pre-processing

Although Kaggle supplies its contestants with several tables of medical records, the prediction models we use require the data to be condensed into a single uniform data table, where each person is represented as a feature vector. Team Market Makers released their solution for this problem as a part of the milestone one prize [5]. We use their method to pre-process the data tables provided by Kaggle and reduce them to two consistent matrices. The first matrix consists of training data, for which the number of days spent in the hospital is known. There are 147,473 members in this set. The second matrix consists of 70,942 members, for which the number of days in hospital are not known. These are the members on which our prediction is scored. This method reduces each person's attributes to 139 unique features. Once this method for representing each person as a vector was established, it then became possible to generate prediction models for the test set.

4.2 K-Nearest Neighbours

The K-Nearest Neighbour predictor finds members in the training set that are most similar to members of the test set within the feature space, and uses a combination of the known number of days spent in the hospital to make predictions for the unknown values. In order to calculate a correlation value between two users, we use the following equation:

$$C_{ij} = \frac{1}{N}\sum_{i=1}^{N}(X_{in} - \bar{X}_i)(X_{jn} - \bar{X}_j)^T \tag{2}$$

where C_{ij} is the correlation coefficient, N is the number of features, $X_{in} - \bar{X}_i$ is the centralized training set, and the $X_{jn} - \bar{X}_j$ is the centralized test set. We use this to figure out which members of the training set are most closely related to the target member of the test set. This method of comparing users is, however, very costly. In order to find the closest matches between members of the test set and training set, each user in the test set must be compared the each member of the training set. In addition, many features are highly correlated to others and, as a result, can cause the correlation algorithm to place too much weight on certain features. We solve this problem using eigenvalue decomposition. Using this method, we reduce the number of features and spherize the data. We combine and decompose the training and test set using the following equation:

$$X = U\lambda U^T \tag{3}$$

We find U, the matrix of eigenvectors, and λ, the diagonal matrix of eigenvalues. We remove all but the n highest-variance eigenvalues from U and λ to create U_n and λ_n and spherize the data using the following equation:

$$\tilde{X}_n = \lambda_n^{-\frac{1}{2}} U_n^T (X - \bar{X}) \tag{4}$$

$\tilde{X}_n$ is a combination of training and test set, with only k features for each member. We use this matrix to calculate the correlation coefficients for each person. We then find the k highest-correlated users and use the correlation coefficients to create a weighted average prediction for that user.

We used this method with $k = 1000$ and $n = 6$ in Matlab and obtained an RMSLE of 0.475197, which achieved a leaderboard position of 603rd place out of a total of over 1,250 teams.

4.3 Logistic Regression

Logistic regression fits a logistic curve onto the training set and then uses this curve as a model to make predictions on a test set. Our prediction σ is given by the following:

$$\sigma(X; w) = \frac{1}{1 + e^{-w^T X - w_o}} \tag{5}$$

Where X is the test set and w is a vector for which we optimize. σ makes predictions in the range [0, 1], but predictions for the Heritage Health Prize range from 0 to 15, so the final prediction we use is $15\sigma(X;w)$. To optimize w, we first examine the cost function, which is the sum squared error (SSE):

$$C(w) = \sum_{i=1}^{N}(y_i - 15\sigma(X;w))^2 \tag{6}$$

We then find the derivative of the cost function and use gradient descent to find an optimal value of w. Because the dataset contains so many members, it is impractical to calculate the gradient for all members at each step. Therefore, we randomly select a member at each step and calculate the gradient for that member only. We call the prediction σ_i for user i. The gradient for that member in terms of w is as follows:

$$g(w) = \frac{\partial C_i}{\partial w} = 15\sum_{i=1}^{N}[(\sigma_i(1-\sigma_i))(15\sigma_i - y_i)X_i] \tag{7}$$

After calculating the cost gradient for member i, we update w by subtracting the gradient multiplied by a step size constant η, which is obtained through cross-validation.

$$w^{t+1} = w^t - \eta g(w^t) \tag{8}$$

In this equation, w^t is the current iteration of w and w^{t+1} is the updated w vector. We repeat this process over a large number of steps until the cost gradient converges on zero, meaning no further steps are necessary.

After optimizing for $\eta = 0.001$ and the number of steps $T = 100,000$, our Matlab implementation of this method achieved an RMSLE of 0.466726 and position number 378 on the Kaggle leaderboard.

4.4 Support Vector Regression

Support Vector Regression is a linear regression method, similar in many ways to logistic regression. What sets it apart is that, in order to prevent over-fitting and reduce the computation time, all data points less than a certain distance from the best-fit line are ignored when calculating the cost gradient. Through cross-validation, we determined that the optimal acceptable distance from this line is $\varepsilon = .02$. Using this value of ε and Liblinear's L2-regularized support vector regression, we obtained an RMSLE of 0.467152, which earned spot 393 on the leaderboard.

4.5 Random Forests

A random forest prediction model utilizes an ensemble of randomly generated decision trees in order to produce its predictions. In essence, it creates multiple decision trees in hopes that the individual errors of each decision tree will cancel each other out when combined in an entire forest. However, in order for this prediction model to work, an element of randomness must be maintained throughout the entire generation of the forest. In our implementation, this element of randomness is introduced by feature selection at the different nodes.

In addition to this, the decision trees were bagged. This allows the different decision trees to be based upon different subsets of the training set, thus helping to prevent over-fitting. Our Matlab implementation of this prediction model used 500 trees, each with a maximum depth of 15, and achieved an RMSLE of 0.464918. This is the second best individual model behind gradient boosting machines and placed us in position 321 on the Kaggle leaderboard.

4.6 Gradient Boosting Machines

Gradient boosting machines produce a prediction model in the form of an ensemble of weaker prediction models, which in our case are decision trees. They build the model in a stage-wise fashion wherein each stage a cost function is optimized, much like other boosting methods do. By giving one our training set which consists of the features of X patients and their corresponding Y days spent in the hospital, it can find an approximation $\hat{F}(x)$ to a function $F^*(x)$ that minimizes the error of our cost function, $C(y, F(x))$.

$$F^* = \arg\min_F E_{x,y} C(y, F(x)) \tag{9}$$

Its parameters include the number of trees to ensemble together, the size of these trees, shrinkage, and the minimum number of observations in the leaves. Through cross-validation, we have found that increasing the number of trees, keeping the feature interaction within six to eight features, incrementally shrinking the updates, and increasing the minimum number of observations all help toward finding the optimal predictions while simultaneously preventing over-fitting. Currently, our best implementation of this model in R uses 8000 trees, a shrinkage of 0.002, a depth of 7, and 100 minimum observations for each leaf, in order to obtain an RMSLE of 0.462998, making this our most accurate individual predictor in position 203 on the leaderboard.

4.7 Neural Networks

Artificial Neural Networking is a computational model which is inspired by the biological workings of neurons. The model is composed of layers of nodes, the artificial equivalent to a neuron. These nodes cascade information to each other through an abundance of weighted lines. The model used in our research is composed of three fully connected layers. The first layer inputs vectored member features. The second layer of nodes combines the input data with different weights. The third combines the nodes again to produce a prediction. [4]

We use the technique of back propagation to modify the feature weights. The process found the error between the model with the true data. This result was then backtracked through the pathways to find which weights were most responsible for the error. These weights were then changed in the direction of negative gradient of the cost function. In our implementation of this predictor, we used 7 neurons in the hidden layer, and completed 3000 cycles. Our most successful neural network predictor, which we implemented with Netlab, obtained an RMSLE of 0.465705, which earned spot number 345 on the leaderboard.

4.8 Summary of Individual Predictors

Now that we have discussed the theoretical basis behind each prediction model, we summarize our various predictors below in order of RMSLE and leaderboard score.

• Gradient Boosting Machines	0.462998 (203rd place)
• Random Forests	0.464918 (321st place)
• Neural Networks	0.465705 (345th place)
• Logistic Regression	0.466726 (378th place)
• Support Vector Regression	0.467152 (393rd place)
• K-Nearest Neighbors	0.475197 (603rd place)

4.9 Blending

Once we have a sufficient number of predictions and an RMSLE for each of them, we are then able to ensemble the various predictors. We approach this through a method inspired by the Netflix prize winning team “BellKor’s Pragmatic Chaos” [7].

The ensembled predictor is produced using the following equation:

$$\tilde{X} = Xw \tag{10}$$

X is a matrix of all predictions made by our various models, and w is a vector used to weight the predictors. $\tilde{X}$ is the final ensembled prediction. We begin the process of finding an optimal weight vector w by first defining this equation for a single member i and for k predictors:

$$\tilde{X}_i = \sum_{c=1}^{k} w_c X_{ic} \tag{11}$$

In order to optimize the weight vector, we utilize a cost function. Since Kaggle scores predictions using RMSLE, we convert all test and training data to a logarithmic scale and instead use root mean squared error (RMSE) in order to simplify the cost function.

$$C = \frac{1}{n} \sum_{i=1}^{n} (Y_i - \tilde{X}_i)^2 \tag{12}$$

We then substitute for $\tilde{X}_i$.

$$C = \frac{1}{n} \sum_{i=1}^{n} (Y_i - \sum_{c=1}^{k} w_c X_{ic})^2 \tag{13}$$

We then derive the cost in terms of w.

$$\frac{\partial C}{\partial w} = \sum_{i=1}^{n} (Y_i - \sum_{c=1}^{k} w_c X_{ic})(-X_{ic}) \tag{14}$$

Because we intend to find the minimum cost, we solve for when the derivative equals zero. This results in the equation:

$$\sum_i Y_i X_{ic} = \sum_i \sum_c w_c X_{ic} X_{ic} \tag{15}$$

We can then rewrite this equation using matrix multiplication.

$$Y^T X = w_c^T X_c^T X_c \tag{16}$$

We proceed by isolating w_c.

$$w_c = (Y^T X)(X^T X)^{-1} \tag{17}$$

At this point, we have a formula for w which incurs the lowest possible cost. However, Y is unknown to us, so we return to the cost function to find an approximation of $Y^T X$.

$$\varepsilon = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i - Y_i)^2} \tag{18}$$

We first manipulate the cost function by assuming that all predictions made by a predictor were zero. Thus, $X_i = 0$. We call the cost of submitting all zeros ε_0. We can substitute this into the cost function and obtain the following:

$$\varepsilon = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(0 - Y_i)^2)} \tag{19}$$

If we simplify this equation and isolate the sum, we find the following equation:

$$n\varepsilon_0^2 = \sum_{i=1}^{n} Y_i^2 \tag{20}$$

This equation will later become relevant. We can also determine ε_c, the cost for an individual classifier c.

$$n\varepsilon_c^2 = \sum_{i=1}^{n}(Y_i - X_{ic})^2 \tag{21}$$

Both ε_0 and ε_c are obtained from Kaggle. We see where equations 20 and 21 become relevant when we expand the term $Y^T X$.

$$Y^T X = \sum_i Y_i X_{ic} = \frac{1}{2}[\sum_i X_{ic}^2 + \sum_i Y_i^2 - \sum_i (Y_i - X_{ic})^2] \tag{22}$$

If we substitute equations 20 and 21 for the second two summation terms, we obtain the following equation in which all terms are known:

$$Y^T X = \frac{1}{2}(\sum_i X_{ic}^2 + n\varepsilon_0^2 - n\varepsilon_c^2) \tag{23}$$

Now that all terms in the equation for w_c are known, we calculate w and obtain an ensembled predictor using equation 10. Our implementation of this method, using several predictions from each model, obtained an RMSLE of 0.461432, which placed us in position 98 on the leaderboard. This placed our team well within the top 10% of submissions, and marks a significant improvement over our most successful individual prediction model.

5 Related Work

One especially helpful feature of the Heritage Health Prize is the requirement that the top two teams must publish their methods to win prize money each time a milestone passes, which usually occurs every six months. At the time of writing, four milestone papers had been published. For our research, the most helpful was the "Market Makers Milestone 1" description [5]. The writers utilized gradient boosting machines and neural networks, predictors that we also employed. The paper also discussed blending several algorithms, and how an ensemble proved to be a far better solution than any one specific model.

The idea of ensembling several predictors to create one which is more accurate than any individual was explored prior to the Heritage Health Prize, in the Netflix Prize. The Netflix Prize was a competition to improve the algorithm that Netflix, a movie rental company, uses to determine what movies users will like, based off of their earlier movie ratings. Like the Heritage Health Prize, the Netflix Prize was a long-term competition based on predicting information from a massive dataset. The winning team's paper, "The BigChaos Solution to the Netflix Grand Prize" [7] details the technique of blending, as well as the utility of k-nearest neighbors, neural networking, and gradient boosting machine predictors within an ensemble. The paper also warns against overfitting based on feedback from test data.

6 Conclusions & Future Work

By optimizing our predictions through blending, we have been able to consistently minimize our error on the leaderboard. This reaffirms the notion that ensembling multiple predictors together produces better results than any individual one can. Although this affirmation is an achievement in and of itself, there is much more work that can be done to improve our ensemble.

We could explore optimizing our blending equation even further through the use of a regularization constant. This may be necessary because the basis of our ensembling method rests on the feedback that Kaggle gives us. Since this feedback is only based upon a subset of the test data, we run the risk of over-fitting to it. This may be counterproductive because, at the end of the competition, we will be scored on a different subset of the test data altogether. Nevertheless, optimal regularization can be determined through cross-validation and can be implemented when finding weights for the ensemble using the following equation:

$$w_c = (Y^T X)(X^T X + \lambda I)^{-1} \tag{24}$$

Further improvements could be made in the area of feature design. For the most part, we have been using the data that the Market Makers group developed as part of their milestone one predictor. However, it may be a good idea to add or remove some features. For example, we might add linear or quadratic combinations of different features that we observe might have interesting relationships with one another. On the opposite side of the spectrum, we could also remove some features that may just act as noise for some of our prediction models. In both cases, we would have to run extensive tests and experiments to ensure that the changes that we make on our features are in fact enabling us to produce better results.

Lastly, we could optimize our prediction models even further and add new ones that might work well on our data set. The top competitors right now have chosen to add Additive Groves and Multivariate Adaptive Regression Splines prediction models to their ensembles, making those good places to start. All in all, there is a plethora of things that we could look into in order to improve our ensemble and climb the leaderboard even further.

References

[1] Heritage provider network health prize description, 2012. http://www.heritagehealthprize.com/c/hhp.

[2] Heritage provider network health prize round 2 milestone leaderboard, 2012. http://www.heritagehealthprize.com/c/hhp/leaderboard.

[3] Francis Galton. Vox populi. *Nature*, 75:445–50, 1907.

[4] James M. DeLeo Judith E Dayhoff. Artificial neural networks: Opening the black box. *Cancer*, 91:1615–1635, April 2001.

[5] David Vogel Phil Brierley and Randy Axelrod. Market makers - milestone 1 description. September 2011.

[6] Ian Saxon. *Intrade Prediction Market Accuracy and Efficiency: An Analysis of the 2004 and 2008 Democratic Presidential Nomination Contests*. PhD thesis, University of Nottingham, September 2010.

[7] Andreas Töscher and Michael Jahrer. The bigchaos solution to the netflix grand prize. September 2009.