Anti-matter detection: Particle Physics Model for KDD Cup 2004

David S. Vogel
MEDai / AI Insight
University of Central Florida
Orlando, FL
dvogel1@cfl.rr.com

Eric Gottschalk
MEDai / AI Insight
Orlando, FL
egottschalk@MEDai.com

Morgan C. Wang
Department of Statistics & Actuarial Science
University of Central Florida
Orlando, FL
cwang@mail.ucf.edu

ABSTRACT

What is the difference between matter and anti-matter? A.I. Insight's winning solution on the Particle Physics Task for the 2004 KDD Cup demonstrates how an accurate predictive model can be formulated without knowledge of the content of the data. Information on the data was not available for the modeling, including a description on the outcome to be predicted. In other words, an 80 x 150,000 grid of numbers with the header "Particle Physics" was all that was given to the 500+ registrants of this competition. Key steps in creating the winning model were interactive analysis of the variables, detection of interactions, a powerful self-organizing neural network, and customization of the 4 different error criteria.

Keywords

KDD Cup, particle physics, data mining, variable transformation, neural network, classification, support vector machine, logistic regression, performance measure, MITCH, NICA.

1. INTRODUCTION

Immediately after the big bang, physicists theorize that there were equal amounts of matter and anti-matter. Fast forward about 1 nanosecond, the universe became to be dominated by matter, just as we know it today [3]. The particle physics task of the 2004 KDD Cup dealt with differentiating between matter and anti-matter. More technically, this classification problem involved distinguishing between two sub atomic particles, B and B-Bar, where B-Bar is the antimatter counterpart of the B meson. The data set for this problem was created at SLAC (Stanford Linear Accelerator Center) [1]. Physicists at SLAC and other institutions would like to understand why the universe is dominated by matter, and predictive models discerning B and B-Bar particles assist in their experiments to answer this fundamental question.

For this task, we were given 78 predictor variables for the binary classification problem. The 78 predictor variables contained mostly real values and described the trajectory of the mesons during high energy experiments. The organizers of the competition made available 50,000 observations to create a predictive model which would be evaluated against different performance measures. A data set with 100,000 observations was used to evaluate the model with the four different criteria. Prior to any data analysis, we randomly removed approximately 10,000 observations from the training set to be used for our own internal validation. There were multiple steps in the model building process: data exploration, transformations, variable creation, modeling techniques, and customization of the predictions for the four different evaluation criteria.

2. DATA PREPARATION

The training set contained one dependent binary variable. The binary variable, took on the value of 0 or 1 depending on whether or not the particle was B or B-Bar. This classification problem was a balanced one, in that the number of particles in B and B-Bar were equal in the training set.

The predictor variables for this data set remain a mystery. We were not told any prior information about the predictors or what they represent. Even if we knew what the predictors represented, it would not have been helpful as the authors of this paper have no background in particle physics.

2.1 Missing Values

Several of the predictor variables in this data set contained missing values. Predictors numbered 20-22 and 44-46 were assigned a value of "999" to denote missing values while the predictor variables numbered 29 and 55 use "9999" to denote missing values. It was important to create two missing value indicator variables, one for predictors 20-22 and a second for predictors 44-46. For the initial 6 variables, the missing values were then replaced with the median of the available values. We considered it inappropriate to try imputation techniques in this situation because our analysis of the variables led us to believe that the value "missing" had its own specific meaning. If this were true, simply blending the variable in with the available values would be a loss of information.

2.2 Variable Categories

Based on an initial analysis of the variables, we categorized the predictors into 4 main groups:

- **Group 1**: 8 variables with values {-1,0,1}. Interactive and symmetric.
- **Group 2**: A single key nominal variable (Predictor #63)
- **Group 3**: 6 individually predictive variables.
- **Group 4**: All others variables, having no correlation to dependent variable.

2.3 Completely Predictive Categories

One of the categorical predictors and it had a very interesting property: some of its categories were 100% predictive. Table 1 demonstrates how predictor #63 allows one to predict with 100% confidence the value of the target class membership.

Values of Variable #63	N	# Class 0	# Class 1
{-8,-2,1,14}	2350 (4.7%)	0	2350
{8,2,-1,-14}	2294 (4.6%)	2294	0

Table 1: The 100% predictive categories of Variable #63 are shown below. The completely predictive nature of these categories accounts for nearly one tenth of the population in the training set.

2.4 Variable Interactions

Using the tool developed at AI Insight, NICA (**N**umerical **I**nteraction **CA**librator), we were able to detect and analyze interactions that occurred within the data. For example, the variables #1 and #4 are barely predictive when used individually. However, we can combine the two to create a very predictive variable. Figures 1-3 illustrate the process on a single interaction.

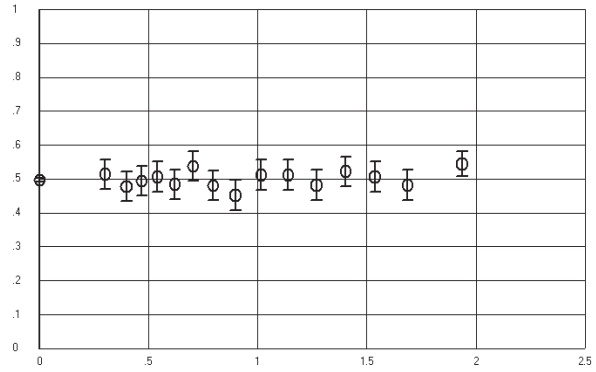

Figure 1: A graph of variable #1, which has a correlation of .006 with the target. The x axis represents the support of variable #1 and the y axis represents the probability of belonging to the positive target class.

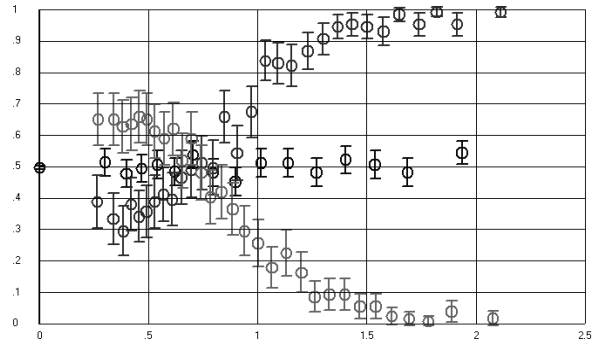

Figure 2: The plot of variable #1 when variable #4 is equal to -1 and 1 is also now included with the original plot of variable #1 from Figure 1.

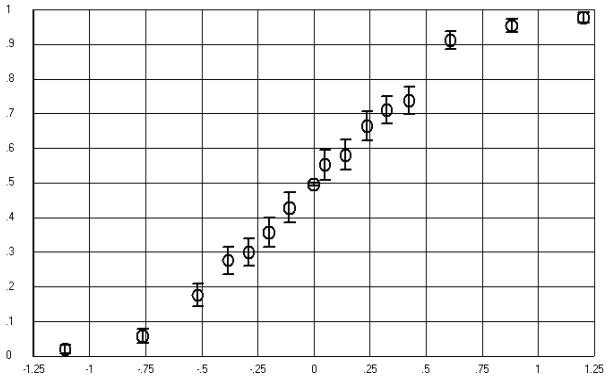

Figure 3: Plot of the new interaction variable defined as *Interaction_V04V01* = V04 * (V01 - .75). This newly generated variable has a correlation of 0.24 with the target class.

We detected thousands of statistically significant interactions from combining different predictors in a similar manner. However, most were not likely to contribute enough to the overall model to justify the increase in model complexity. In total, we generated 72 first order interaction variables, 185 second order interaction variables, and 301 third order interaction variables.

3. MODEL BUILDING

Combining the original variables, the missing value indicators, the key categories, and the 1st, 2nd, and 3rd order interaction variables, we obtained a grand total of 639 predictors in the data set. Variable selection techniques were attempted to lower this level of complexity, but all attempts to reduce the number of interactions proved to lower the validation set accuracy. With all of the 639 predictors as input into MITCH neural networks, we were able to achieve the best validation set results in all 4 categories. We also attempted models using logistic regression, standard "off-the-shelf" neural networks, SVM's (support vector machines), nearest-neighbor techniques, decision trees, and tree-based boosting methods. Because the predictors were continuous, tree based techniques generally did not perform as well. A table of results was considered, but not included because a fair comparison of the various techniques would require significantly more analysis. Some techniques did better than others without interactions. Different techniques benefited more than others from using the interactions. But it was universally true that the use of MITCH neural networks combined with interaction terms performed significantly better than all other methods and combinations.

4. PERFORMANCE CRITERIA

There were four criteria that our predictions were evaluated against. The organizers of the KDD cup made the program "PERF" available to participants in this task. This allowed us to

evaluate our test set with the same software that would be used to decide the winner of the task. Two of the performance criteria, ROC area and accuracy were not directly optimized, but scored the best by minimizing the Bernoulli error function during the neural network training. We paid closer attention to the two measures: Cross Entropy and Q-Score.

4.1 Cross Entropy

Cross Entropy is a measure that requires close attention because the penalties for mis-classifications with extreme probabilities can be quite severe. In fact, a mis-classified probability of 0 or 1 would result in an infinite error. The evaluation software modifies such extreme errors to be a "googol" (well, almost a googol: 1E+99). Therefore, even the 100% predictive categories were modified to have probabilities of 0.005 or 0.995. All other predictions were truncated at 0.01 and 0.99. We considered this tactic "playing it safe." 14 teams obtained a cross-entropy error of a "googol" somewhere in their solution, which was apparent by their extremely high error for this category. Several others obtained a cross-entropy worse than that of simply predicting 0.5 for every record; evidence that care must be taken for extreme probabilities for this category. We obtained the best result by setting the MITCH Neural Network to directly minimize the Bernoulli error function, which is equivalent to minimizing Cross Entropy.

4.2 SLAC Q Score

The SLAC Q-Score is a measure specific to the field of particle physics, and is of high importance as it is proportional to how frequently the particle accelerator must be run. Traditionally, neural networks are used to minimize the more traditional error functions: Bernoulli and Sum of the squared errors. For optimizing Q-Score, we obtained a more accurate solution using a more creative error function: (Predicted – Actual) ^ 6. This is the Minkowski-R error function with the value of R equal to 6 [2]. We then re-calibrate the predictions using the formula:

$$\text{New Prediction} = \frac{\left(\frac{p}{1-p}\right)^{1/5}}{1+\left(\frac{p}{1-p}\right)^{1/5}}$$

This kind of recalibration is necessary for many non-standard error functions because they tend to cause the predictions to accumulate around certain probabilities. More specifically, high order polynomial error functions cause the probabilities to move closer to 0.5. The recalibration formula causes the predictions to be more evenly distributed, leading to a better Q-Score.

5. Discussion

It was an unexpected result that a model could generalize so well with 639 predictors on only 40,000 records. The optimal strategy for predictive modeling is to examine an abundance of potential predictors, and then use variable selection techniques to pinpoint the subset of predictors that obtains the best result on the validation data set. Given the time constraints of the competition, we had time for one shot at the creation of variables. The fact that we could not trim down the 639 predictors is indicative that a larger set of predictors should have been generated for an even better result.

We only investigated multiplicative interactions in the data, as they are the fastest to run and least time-consuming to integrate into a model. For a data set like this with such a high number of interactions, it is likely that many influential non-multiplicative interactions exist that we simply did not look for.

Predictors were used in their raw form (with the exception of null values), but some had extremely non-normal distributions. Improvements would be likely if some time were spent on variable transformations.

The methodology discussed in this paper is extremely scalable, and could be taken a step further using more predictors and more data (if available) to create a model that is significantly more accurate than the one submitted for this contest.

It should be noted that although most of the patterns in the variables and interactions appeared to be perfectly symmetric, there were slight aberrations from the symmetry. These aberrations were very small, yet statistically significant. Perhaps these patterns should be more closely examined by physicists to determine if they are linked to the puzzle of why this universe is not symmetric with respect to quantities of matter versus anti-matter.

6. ACKNOWLEDGMENTS

Our thanks to KDD Cup 2004 organizers Rich Caruana and Thorsten Joachims at Cornell University for the interesting and well run competition. We would also like to thank Charles Young at SLAC for the donation of the data for this task. Finally, we acknowledge A.I. Insight, Inc. and MEDai, Inc. for the use of their predictive modeling technology MITCH (Multiple Intelligent Tasking Computer Heuristics).

7. REFERENCES

[1] ACM *SIGKDD KDD Cup* 2004 homepage http://kodiak.cs.cornell.edu/kddcup.

[2] Bishop, C. M. Neural Networks for Pattern Recognition. Oxford University Press, Oxford UK, 1995.

[3] Salisbury, D. and Riordan, M. Antimatter: Not just for sci-fi anymore. Stanford Online Report (August 9, 2000).

About the authors:

David Vogel is the Senior Scientist at A.I. Insight, where he has spent over 6 years leading the development of their modeling tool MITCH (Multiple Intelligent Tasking Computer Heuristics). David has been the winner of the KDD Cup in 3 different topics as well as honorable mentions on 3 additional topics, attesting to the versatility of his approach to predictive modeling. David's research interests include development of innovative modeling techniques, scalable algorithms, and new applications for predictive models.

Eric Gottschalk has recently joined A.I. Insight as a Research Associate, where he assists in the development of MITCH technology. Eric's research interests are in wavelets and their applications to data mining, and also non-linear time series analysis. Eric studied applied math and computer science at the University of Colorado, and earned a MS in Data Mining from the University of Central Florida.

Morgan C. Wang is Professor in the Department of Statistics and Actuarial Science of the University of Central Florida. He and his colleague established a SAS Data Mining Certificate Program in the Fall of 2000 and started a Master's degree in Data Mining in the Fall of 2001. Every graduate of the program has served an internship with an eminent Orlando business partner in the aerospace, entertainment, hospitality or automobile service industry. Moreover, all faculty have established consulting relationships with industrial clients inspiring relevant research directions, student employment opportunities and enhanced curriculum case studies.