KDD-Cup 2004: Results and Analysis

Rich Caruana
Cornell University
Dept. of Computer Science
Ithaca, NY, US
caruana@cs.cornell.edu

Thorsten Joachims
Cornell University
Dept. of Computer Science
Ithaca, NY, US
tj@cs.cornell.edu

Lars Backstrom
Cornell University
Dept. of Computer Science
Ithaca, NY, US
lb87@cornell.edu

ABSTRACT

This paper summarizes and analyzes the results of the 2004 KDD-Cup. The competition consisted of two tasks from the areas of particle physics and protein homology detection. It focused on the problem of optimizing supervised learning to different performance measures (accuracy, cross-entropy, ROC area, SLAC-Q, squared error, average precision, top 1, and rank of last). A total of 102 groups participated in the competition, 6 of which received awards or honorable mentions. Their approaches are described in other papers in this issue of SIGKDD Explorations. In this paper we do not analyze any particular approach, but give insight into the performance of the field of competitors as a whole. We study what fraction of the participants found good solutions, how well participants were able to optimize to different performance measures, how homogeneous their submitted predictions are, and if the best submissions represent the maximal performances that could reasonably be achieved. We are keeping the KDD-Cup 2004 WWW site open and have added an automatic scoring system for new submissions in order to encourage further research in this area.

1. INTRODUCTION

Real-world applications of data mining typically require optimizing to non-standard performance measures that depend on the specific application. For example, in direct marketing, accuracy is a poor indicator of performance, since there is a strong imbalance in cost between missing a customer and making the advertisement effort a little too broad. Even for the same dataset, we often want to have different classification rules that optimize different criteria. For example, in information retrieval, we sometimes want to optimize precision, at other times want to optimize recall, and at other times need to optimize to a measure that balances both precision and recall (e.g. F-Score). The need for data mining methods that allow optimizing to different performance measures inspired the tasks of this year's KDD-Cup. In particular, this year's KDD-Cup focused on optimizing predictions to a variety of performance measures in supervised classification.

The 2004 KDD-Cup competition included two datasets — a binary classification task for a quantum physics problem, and a protein homology prediction task. We provided a supervised training set and an unlabeled test set for each task. Participants were asked to submit 4 sets of predictions for each test set, each set of predictions maximizing performance according to a particular measure. The four performance metrics for the physics problem were accuracy, cross-entropy, ROC area, and SLAC Q-Score. The four metrics for the protein problem were squared error, average precision, top1, and rank of last. These eight metrics are described in Sections 2.1 and 2.2. We provided software[1] to the participants to standardize how the performance measures were computed. The same software was used to determine the winners of the competition.

In this report we describe the two KDD-Cup 2004 tasks in detail[2], provide participation statistics, and announce the winners and how they were determined. We also analyze the results of the competition. The submissions of more than 100 participants are an interesting dataset to mine for patterns of how the field as a whole performed on the two data-mining tasks. For example, what percentage of the field provided reasonable solutions? Did the winners perform significantly better than the rest of the field? Did the participants improve their performance by optimizing to particular performance measures? Did all groups that performed well find essentially the same solutions? These are some of the questions we address in this report.

2. DATASETS AND TASKS

2.1 Particle Physics Task

The first of the two KDD-Cup 2004 tasks is a particle physics classification problem. At the Stanford Linear Accelerator Center (SLAC), high energy particle beams are collided to generate subatomic particles. A major challenge in some of these experiments is to correctly classify the particle tracks. In the Physics Task, the goal is to learn a classification rule that differentiates between two types of particles generated in collider experiments based on 78 attributes. The training set has 50,000 examples, and the test set has 100,000 examples. The data set was contributed by Charles Young et al. from SLAC. The source of the data set, the identity of the two particles, and the definitions of the 78 attributes were hidden from participants to prevent competitors from trying to gain an advantage by studying the physics of the problem.[3]

[1] `http://kodiak.cs.cornell.edu/kddcup/software.html`
[2] Additional information can be found on the KDD-Cup 2004 WWW-site at `http://kodiak.cs.cornell.edu/kddcup`
[3] One might argue that in data mining one should take advantage of background knowledge and any other domain-specific information that is available, which argues against

In the competition, we measure performance on the particle physics problem using four metrics:

ACC (maximize) We use the usual definition of accuracy – the number of cases predicted correctly, divided by the total number of cases. Predictions must be made for all cases. The predictions submitted for accuracy were allowed to be boolean or continuous. For this metric, participants were required to submit a prediction threshold: predictions above or equal to this threshold were treated as class 1, predictions below threshold as class 0. The goal, of course, is to maximize accuracy on the test set. An accuracy of 1.00 is perfect prediction. Accuracy near 0.00 is poor. Because the Physics Task is a balanced problem, baseline accuracy is 0.50.

AUC (maximize) We use the usual definition for area under the ROC curve. An ROC plot shows the true positive rate vs. false positive rate as the prediction threshold sweeps through all the possible values. This is the same as plotting sensitivity vs. 1-specificity as the threshold is swept. AUC is the area under this curve. An AUC of 1 indicates perfect predictions – all positive cases sorted above all negative cases. AUC of 0.5 is random prediction – there is no relationship between the predicted values and truth. AUC below 0.5 indicates there is a relationship between predicted values and truth, but the model is backwards, i.e. tends to predict smaller values for positive cases! An alternate way to think of AUC is to imagine sorting the data by predicted values, and counting the number of swaps needed to properly order the data by class:

$$AUC = 1.0 - \frac{\#\ swaps}{(\#\ positives) \times (\#\ negatives)}$$

CXE (minimize) We use the usual definition for cross-entropy, but protect the cross-entropy from becoming infinite. Cross-entropy, like squared error, measures how close predicted values are to target values. Unlike squared error, cross-entropy assumes the predicted values are probabilities on the interval [0,1] that indicate the probability that the case is class 1.

$$CXE = -\sum_{cases} [(T) \times \log(P) + (1-T) \times \log(1-P)]$$

where T is the Target Class (0 or 1) and P is the predicted probability that the case is class 1. Note that the terms (T) and $(1-T)$ are alternately 0 or 1 so $log(P)$ is added to the sum for positive cases and $log(1-P)$ is added for negative cases. Note that cross entropy is infinite if $T = 0$ but $P = 1$, or if $T = 1$ but $P = 0$. In the code provided to calculate cross-entropy we protect against this by returning a very, very large number instead of infinity. This helps minimize platform dependence. To make cross-entropy independent of data set size, we use the mean cross-entropy, i.e., the sum of the cross-entropy for each case divided by the total number of cases.

SLQ (maximize) The Slac Q-Score (SLQ) is a domain-specific performance metric devised by researchers at the Stanford Linear Accelerator (SLAC) to measure the quality of predictions made for certain kinds of particle physics problems. SLQ works with models that make continuous predictions on the interval [0-1]. It breaks this interval into a series of bins. For the KDD-CUP we used 100 equally sized bins: 0.00-0.01, 0.01-0.02, ..., 0.98-0.99, 0.99-1.00. SLQ places predictions into the bins based on their predicted values. In each bin SLQ keeps track of the number of true positives and true negatives. SLQ is maximized if bins have high purity, e.g. if all bins contain all 0's or all 1's. This is unlikely, so SLQ computes a score based on how pure the bins are:

$$\text{SLQ} = \sum_{bins} \varepsilon(1-2w)^2$$

where ε is the percent of events accepted for prediction, and w is the probability of misclassification. Note that SLQ only cares about the purity in each node. A model would have poor accuracy, and AUC below 0.5, if you switch the labels used for positive and negatives after training, but SLQ is insensitive to this.

Collaborators at SLAC tell us that SLQ is an important quantity custom designed for this particle physics problem that estimates the statistical power of the learned model. Increasing SLQ by 5% is equivalent to having 5% more data, which potentially saves hundreds of thousands of dollars or more in accelerator time.

On both the Physics and Protein tasks contestants are allowed to optimize their learning methods for each metric and submit different predictions for the test set for each of the four metrics on each task.

2.2 Protein Homology Task

Unlike the Physics problem where each training or test case is independent, this task has more complex structure. The goal in this task, contributed by Ron Elber, is to predict which proteins are homologous to a native sequence. The data is grouped into blocks around each native sequence. We provided 153 native sequences as the training set, and 150 native sequences as the test set. For each native sequence, there is a set of approximately 1000 protein sequences for which homology predictions are needed. Homologous sequences are marked as positive examples, non-homologous sequences (also called "decoys") are the negative examples. On average, each native sequence contains over 100 decoys per homologous sequence, making this a very unbalanced problem. The goal is to predict which of the 1000 proteins are homologous to the native protein based on 74 attributes. These 74 attributes are a variety of scores that describe the match between two proteins. These scores include, for example, the length of the longest local alignment, the per-

us hiding the identity of the problem. However, we decided to not provide participants with the identity of the task and the contributor because we thought few participants would be able to spend the time to learn enough particle physics to understand the problem, and because we did not want to burden our collaborators at SLAC with requests from participants for additional information.

centage sequence identity after alignment, or the z-score of the global alignment[4].

Evaluation measures are applied to each block corresponding to a native sequence, and then averaged over all blocks. Most of the measures are rank-based and assume that model predictions provide a ranking within each block from "most likely homologous" to "least likely homologous". The task is to provide 4 sets of predictions for the test set, each of which optimizes one of the following 4 performance measures.

TOP1 (maximize) This measure is defined as the fraction of blocks with a homologous sequence ranked highest. TOP1 is calculated conservatively when there are ties. If multiple sequences are tied for rank 1, all of them must be homologous. If any of the sequences tied for rank 1 are not homologous, the TOP1 score is 0 for that block. (This means it is never beneficial to have ties.) TOP1 captures how well the model predicts on the most confident cases. If search engines such as Google could achieve perfect TOP1, instead of returning a list of potential hits they could just jump directly to the correct hit.

RKL (minimize) This score is the average over the blocks of the rank of the lowest ranked homologous sequence. Again ties are treated conservatively: if multiple sequences tie, the lowest element of the tie determines the rank, so ties are not beneficial. RKL complements TOP1 and measures the accuracy of predictions on the least confident homologs. If a search engine has better RKL, users do not have to search as far down the list of potential hits to find the correct hit.

RMS (minimize) RMS measures the root mean squared error with 0/1 targets. This is one way of evaluating how well the predicted values approximate probabilities. For the competition we calculate the RMSE for each block, and then take the average RMSE across the blocks.

APR (maximize) This score is defined as the average of the average precision of each block. Average precision is a measure that is widely used to evaluate rankings in information retrieval. It can be thought of as the area under the precision/recall curve [1] and provides an overall evaluation of ranking quality. There are a variety of methods for calculating average precision. The methods differ in how they handle ties and in how they accumulate area under the precision/recall plot. After careful consideration, we decided to define average precision as the average of the precisions at each of the recall values for which precision is well defined. If there are multiple precision values for the same recall value, we used only the highest one in our average. To resolve situations where multiple cases are predicted with the same value (ties), we consider all possible orderings of the tied cases, and take the expected precision over these orderings.

Note that three of the measures (TOP1, RLK, and APR) depend only on the relative ordering of the matches within each block, not on the predicted values themselves. Only RMS measures the accuracy of the predicted values beyond the ordering that they induce.

[4]To conceal the identity of the proteins and prevent participants from looking up known homologs in public databases, no description of the proteins and features was given to the participants during the competition.

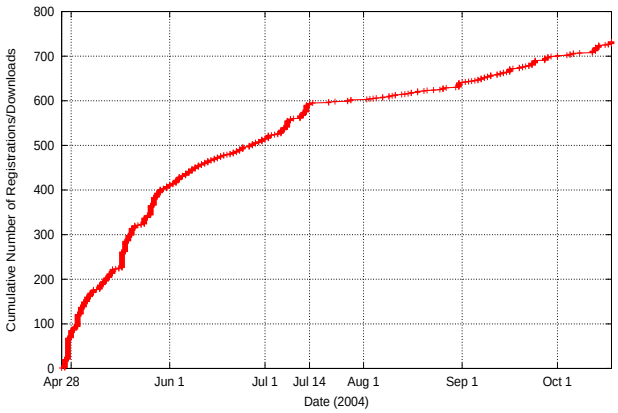

Figure 1: Number of Downloads

3. COMPETITION RULES & SCHEDULE

The competition started on April 28^{th} with the publication of the datasets and the task descriptions on the WWW. The datasets each included a labeled training set as well as a test set from which we withheld the labels. To download the data, groups had to register and were assigned an anonymous ID under which they could later submit their results. The anonymous ID was also used to publish results, so that groups which did not wish to be identified could remain anonymous. Groups that did want to be identified were allowed to replace their anonymous ID with a group name.

The contest was open to any party planning to attend the SIGKDD 2004 Conference. Since the two tasks were evaluated separately, each group could enter in both tasks, in only one task, or in only one particular performance measure on one task. Each person was allowed to participate in only one competing group per task. Each group was allowed to make multiple submissions for each task and performance measure. We did not provide any feedback about performance when submissions were made. Only the last submission before the deadline was evaluated for the competition. All previous submissions from the same group were discarded.

Submissions were made via a WWW interface. The web interface performed tests on submissions to make sure that the formatting of the submission was correct and that predictions were submitted for all test cases. To help detect formatting problems in the submissions, each test example was assigned a unique identifier. Groups had to submit their predictions including those case identifiers. This allowed the submission interface to give immediate feedback on the integrity of the predictions (e.g. wrong number of lines, duplicate or missing example id's, etc.).

By the deadline for the submission of predictions on July 14^{th}, more than 500 groups had registered to download the data. The cumulative number of downloads over time is depicted in Figure 1. We suspect that some of the last-minute registrations just before the July 14 deadline are

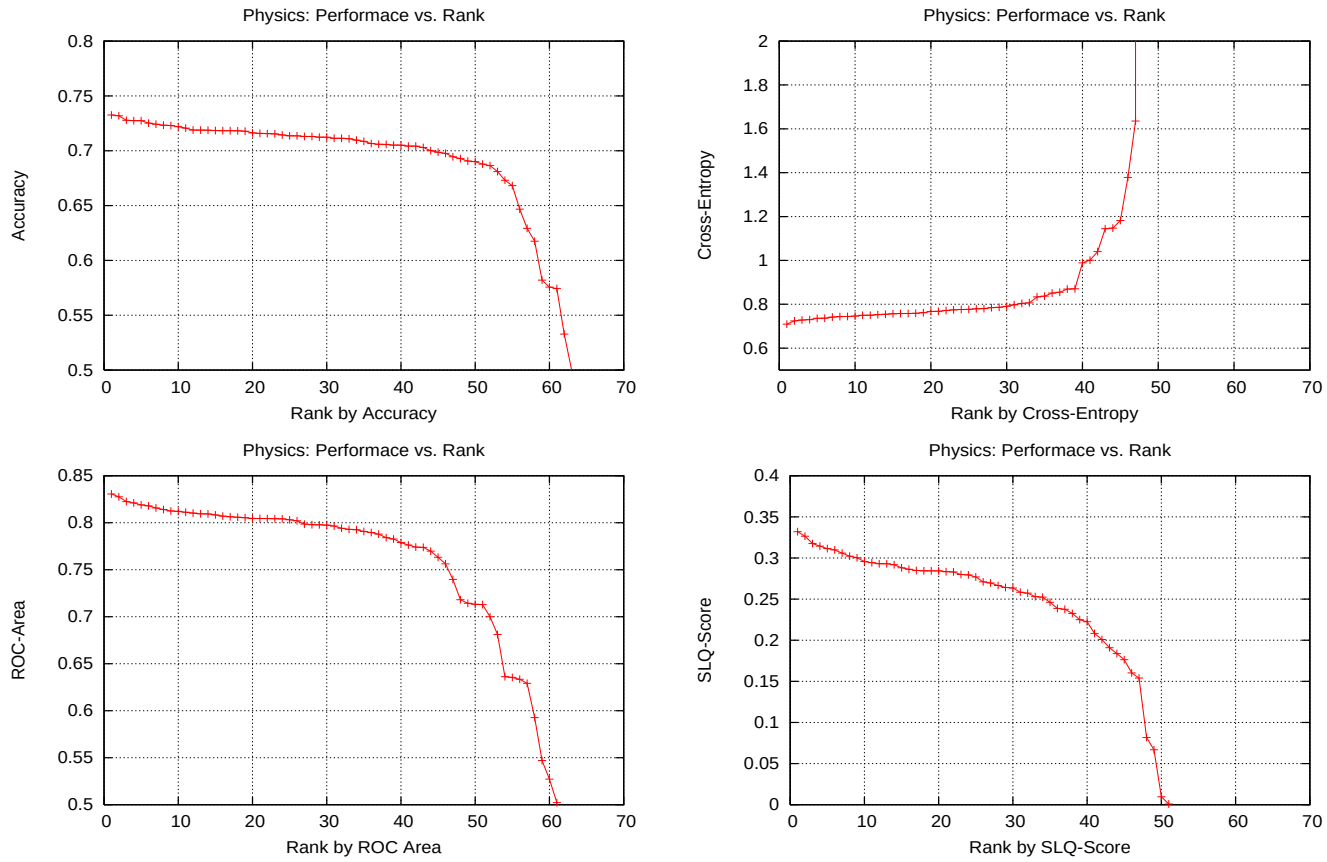

Figure 2: Performance of group vs. rank of group on Physics task.

from groups that registered a second time after forgetting their assigned login information. The registration interface remained open after the submission deadline, so that it is still possible to download the data. Since the end of the competition, more than 100 new users have registered.

By the July 14^{th} deadline, 102 groups had submitted predictions. Of these, 65 groups participated in the Physics task and 59 groups participated in the Protein task. 22 groups participated in both tasks. An analysis of the email addresses revealed a broad international participation. We received submission from 49 country suffixes (including .com, .edu, etc.). The broad international participation is also reflected in the distribution of winners. As described below, the winners and honorable mentions went to groups from China, Germany, India, New Zealand, and the USA. Roughly half of the winners are primarily affiliated with commercial companies, the other half are from academia.

At the opening day of the SIGKDD Conference, the winners were officially announced. There were two main prizes: "Overall Winner of the Quantum Physics Task" and "Overall Winner of the Protein Homology Prediction Task". These overall winners were determined according to the following method. All participants were ranked according to their performance on the test set for each task and performance measure (four measures per task). Not submitting predictions for a performance measure resulted in being ranked last for that measure. The overall winner of a task was the participant who had the best average rank over the four performance measures for that task.

In addition to the overall winners, we also awarded eight honorable mentions, one for each metric on each task. Honorable mentions were given to the group who ranked highest for that particular task and performance measure.

4. RESULTS AND WINNERS

Tables 1 and 2 show the test-set performance of the submissions made by each group on each task and metric. The tables also include the rank for each of these scores. In addition, the first column in each table is the overall rank of each group on the whole task. This rank is based on the average rank of each group across the four tasks (last column), as well as on a statistical analysis of the ranks that is described later in this section. Entries in bold in the tables placed either 1^{st}, 2^{nd}, or 3^{rd} overall (based on average rank), or received an honorable mention for one or more metrics.

While the rules outlined in the previous section provide clear guidelines for determining winners, we also were interested in the significance of the performance differences between the top competitors. Let's begin with a qualitative overview of the performances before jumping into a detailed statistical analysis of the observed differences in performance. Figures 2 and 3 show the performances for each task and performance measure plotted in increasing order, so that the leftmost point of each graph represents the group with the best performance.

Interestingly, most plots have roughly the same shape. The

Table 1: Quantum Physics Results Table.

Rank	Group	Accuracy		AUCA		Cross Entropy		SLQ Score		Avg Rank
		Rank	Score	Rank	Score	Rank	Score	Rank	Score	
1	**MEDai/AI Insight**	2.0	0.73187	**1.0**	**0.83054**	**1.0**	**0.70949**	**1.0**	**0.33280**	**1.333**
2	**Inductis**	**1.0**	**0.73255**	2.0	0.82754	2.0	0.72456	2.0	0.32648	1.667
3	**Golden Helix**	3.0	0.72775	3.0	0.82250	4.0	0.73001	3.0	0.31749	3.333
4	Ahmad abdulKader	5.0	0.72744	6.0	0.81791	3.0	0.72798	6.0	0.30982	4.667
5	Salford Systems	4.0	0.72745	4.0	0.82109	8.0	0.74371	4.0	0.31447	5.333
6	Probing - JL&BZ	6.0	0.72522	5.0	0.81906	6.0	0.73634	5.0	0.31142	5.667
7	Andre and Tiny	7.0	0.72424	8.0	0.81412	7.0	0.74137	9.0	0.30217	7.333
8	FEG, Japan	11.0	0.72060	7.0	0.81572	5.0	0.73583	7.0	0.30591	7.667
9	191	9.0	0.72304	9.0	0.81252	10.0	0.74632	8.0	0.30410	9.333
10	14	8.0	0.72332	10.0	0.81208	14.0	0.75432	-	-	10.667
11	Tiberius	12.0	0.71884	11.0	0.81116	13.0	0.75301	10.0	0.29569	12.000
12	CoSCo	15.0	0.71836	13.0	0.80952	11.0	0.74999	12.0	0.29307	13.000
13	UIUCSFP	13.0	0.71883	12.0	0.81028	15.0	0.75644	11.0	0.29441	13.333
14	159	14.0	0.71870	15.0	0.80829	12.0	0.75015	14.0	0.29176	13.667
15	408	21.0	0.71584	16.0	0.80699	18.0	0.75878	15.0	0.28814	18.333
16	64	16.0	0.71833	25.0	0.80310	16.0	0.75789	-	-	19.000
17	415	19.0	0.71790	21.0	0.80456	20.0	0.76745	24.0	0.27987	20.000
18	584	20.0	0.71633	24.0	0.80419	22.0	0.77181	-	-	22.000
19	Weka	18.0	0.71824	20.0	0.80458	29.0	0.78607	23.0	0.28297	22.333
20	Rueping	26.0	0.71357	23.0	0.80428	19.0	0.76196	19.0	0.28440	22.667
21	7	33.0	0.71096	31.0	0.79646	9.0	0.74423	18.0	0.28441	24.333
22	167	25.0	0.71359	22.0	0.80445	27.0	0.78032	22.0	0.28332	24.667
23	347	10.0	0.72196	14.0	0.80934	52.0	1.742E73	13.0	0.29295	25.333
24	362	30.0	0.71234	26.0	0.80211	24.0	0.77623	21.0	0.28360	26.667
25	182	27.0	0.71311	17.0	0.80642	38.0	0.86928	17.0	0.28478	27.333
26	433	31.0	0.71141	27.0	0.79860	26.0	0.77967	-	-	28.000
27	27	32.0	0.71139	29.0	0.79779	28.0	0.78481	27.0	0.26959	29.667
28	585	34.0	0.70972	30.0	0.79748	25.0	0.77690	26.0	0.27092	29.667
29	agileumbrella	45.0	0.69863	28.0	0.79796	17.0	0.75798	25.0	0.27694	30.000
30	382	36.0	0.70663	32.0	0.79420	23.0	0.77459	28.0	0.26659	30.333
31	66	37.0	0.70588	35.0	0.79074	21.0	0.76813	32.0	0.25839	31.000
32	3	22.0	0.71560	19.0	0.80535	60.0	8.100E73	20.0	0.28437	33.667
33	586	23.0	0.71544	18.0	0.80582	61.0	9.000E73	16.0	0.28626	34.000
34	UIUCstat	39.0	0.70511	37.0	0.78783	31.0	0.79766	35.0	0.25245	35.667
35	8	41.0	0.70428	36.0	0.78959	30.0	0.79063	33.0	0.25722	35.667
36	117	40.0	0.70508	34.0	0.79257	34.0	0.83395	29.0	0.26410	36.000
37	Claudio Favre	29.0	0.71238	33.0	0.79303	50.0	1.449E73	30.0	0.26351	37.333
38	500	43.0	0.70299	38.0	0.78413	32.0	0.80406	37.0	0.24607	37.667
39	60	38.0	0.70577	40.0	0.77886	37.0	0.85489	39.0	0.23758	38.333
40	Jylin	17.0	0.71832	48.0	0.71813	53.0	2.535E73	46.0	0.19088	39.333
41	138	46.0	0.69766	41.0	0.77626	35.0	0.83694	40.0	0.23265	40.667
42	Monash SBS	51.0	0.68778	39.0	0.78269	33.0	0.80625	38.0	0.23876	41.000
43	PG445 UniDo	42.0	0.70426	43.0	0.77386	40.0	0.98868	41.0	0.22493	41.667
44	26	24.0	0.71438	49.0	0.71429	54.0	2.571E73	47.0	0.18385	42.333
45	276	49.0	0.69057	44.0	0.76975	39.0	0.87045	42.0	0.22243	44.000
46	13	28.0	0.71304	51.0	0.71284	55.0	2.583E73	-	-	44.667
47	42	55.0	0.66814	47.0	0.73973	36.0	0.85084	45.0	0.19766	46.000
48	Orrego-WVU	52.0	0.68649	46.0	0.75629	43.0	1.14441	44.0	0.20091	47.000
49	jacek	50.0	0.69006	45.0	0.76322	49.0	1.317E73	43.0	0.20832	48.000
50	219	48.0	0.69284	54.0	0.63626	47.0	1.63563	34.0	0.25298	49.667
51	409	54.0	0.67311	50.0	0.71333	45.0	1.18190	48.0	0.17636	49.667
52	153	44.0	0.70010	52.0	0.69997	56.0	2.699E73	49.0	0.16020	50.667
53	WizSoft	53.0	0.68107	53.0	0.68107	51.0	1.656E73	50.0	0.15401	52.333
54	518	61.0	0.57431	55.0	0.63547	44.0	1.14744	-	-	53.333
55	148	62.0	0.53276	59.0	0.54706	42.0	1.04035	54.0	0.00964	54.333
56	154	59.0	0.58207	58.0	0.59276	46.0	1.37861	52.0	0.08170	54.333
57	352	64.0	0.29094	62.0	0.31402	41.0	1.00119	-	-	55.667
58	187	63.0	0.49942	61.0	0.50224	48.0	1.170E73	36.0	0.24698	57.333
59	HKNN	57.0	0.62920	57.0	0.62918	58.0	3.337E73	53.0	0.06680	57.333
60	142	58.0	0.61752	56.0	0.63344	59.0	3.714E73	-	-	57.667
61	264	60.0	0.57565	60.0	0.52715	57.0	3.264E73	51.0	0.09919	59.000
62	206	-	-	-	-	-	-	31.0	0.26032	-
63	318	35.0	0.70858	-	-	-	-	-	-	-
64	385	47.0	0.69455	42.0	0.77415	-	-	-	-	-
65	568	56.0	0.64672	-	-	-	-	-	-	-

Table 2: Protein Homology Results Table.

Rank	Group	Top 1		RMSE		RKL		APR		Avg Rank
		Rank	Score	Rank	Score	Rank	Score	Rank	Score	
1	**Weka**	3.5	0.90667	8.0	0.03833	3.0	52.44667	2.0	0.84091	**4.125**
1	**ICT.AC.CN**	2.0	0.91333	**1.0**	**0.03501**	14.0	54.08667	**1.0**	**0.84118**	4.500
1	**MEDai/AI Insight**	**1.0**	**0.92000**	5.0	0.03805	13.0	53.96000	3.0	0.83802	5.500
4	Mario Ziller	3.5	0.90667	4.0	0.03766	15.0	55.00667	6.0	0.82422	7.125
5	Rong Pan	11.0	0.88667	2.0	0.03541	16.0	58.85333	5.0	0.82459	8.500
6	Probing - JL&BZ	6.5	0.89333	18.0	0.03952	2.0	52.42000	10.0	0.81931	9.125
7	560	16.5	0.88000	6.0	0.03826	9.0	53.24000	13.0	0.81344	11.125
8	285	11.0	0.88667	16.0	0.03923	11.0	53.30000	8.0	0.82066	11.500
9	**PG445 UniDo**	28.0	0.86667	14.0	0.03878	**1.0**	**45.62000**	4.0	0.82995	11.750
10	587	11.0	0.88667	3.0	0.03692	24.0	64.58667	9.0	0.82006	11.750
11	206	5.0	0.90000	17.0	0.03941	18.0	59.11333	12.0	0.81883	13.000
12	591	22.0	0.87333	7.0	0.03830	4.0	52.84667	23.0	0.79938	14.000
13	575	22.0	0.87333	9.0	0.03838	6.5	53.06667	21.0	0.80187	14.625
14	584	11.0	0.88667	21.0	0.04097	23.0	61.71333	7.0	0.82420	15.500
15	513	22.0	0.87333	11.0	0.03848	12.0	53.37333	19.0	0.80298	16.000
16	276	6.5	0.89333	22.0	0.04135	21.0	59.80667	15.0	0.80672	16.125
17	541	22.0	0.87333	10.0	0.03847	8.0	53.20000	27.0	0.79629	16.750
18	539	28.0	0.86667	12.5	0.03850	5.0	52.90000	22.0	0.79941	16.875
19	540	28.0	0.86667	12.5	0.03850	10.0	53.26667	30.0	0.79560	20.125
20	14	16.5	0.88000	26.0	0.04541	27.0	68.37333	14.0	0.80706	20.875
21	504	11.0	0.88667	29.5	0.05182	22.0	60.86667	25.0	0.79783	21.875
22	Salford Systems	16.5	0.88000	19.0	0.03962	40.0	96.78667	16.0	0.80631	22.875
23	588	11.0	0.88667	33.0	0.05436	29.0	70.10667	20.0	0.80292	23.250
24	578	22.0	0.87333	24.0	0.04314	37.0	93.02667	11.0	0.81902	23.500
25	382	28.0	0.86667	27.0	0.04991	26.0	68.28667	18.0	0.80500	24.750
26	Martine Cadot	28.0	0.86667	25.0	0.04499	20.0	59.74000	26.0	0.79728	24.750
27	FEG, Japan	16.5	0.88000	20.0	0.03989	39.0	95.72667	29.0	0.79569	26.125
28	561	22.0	0.87333	15.0	0.03900	33.0	79.88667	35.0	0.77032	26.250
29	362	31.5	0.86000	23.0	0.04284	34.0	84.88667	17.0	0.80545	26.375
30	595	22.0	0.87333	32.0	0.05433	28.0	69.39333	24.0	0.79895	26.500
31	182	11.0	0.88667	36.0	0.09157	41.0	101.96667	31.0	0.79009	29.750
32	593	44.5	0.72667	29.5	0.05182	6.5	53.06667	45.5	0.64391	31.500
33	159	31.5	0.86000	50.0	0.16669	19.0	59.67333	28.0	0.79622	32.125
34	98	40.5	0.80000	31.0	0.05375	17.0	58.92667	40.0	0.73852	32.125
35	212	42.5	0.78667	28.0	0.05023	36.0	89.60667	41.0	0.71418	36.875
36	471	33.5	0.85333	38.0	0.10133	44.0	116.20667	34.0	0.77871	37.375
37	154	39.0	0.82000	49.0	0.15974	30.0	74.56000	32.0	0.78721	37.500
38	594	35.0	0.84000	54.0	0.26759	25.0	66.36667	37.0	0.76827	37.750
39	167	33.5	0.85333	39.0	0.10276	43.0	114.85333	38.0	0.76495	38.375
40	590	36.5	0.83333	48.0	0.13528	32.0	77.63333	39.0	0.76341	38.875
41	Pierron/Martino	36.5	0.83333	34.0	0.05856	46.0	179.98667	42.0	0.70717	39.625
42	500	40.5	0.80000	58.0	8.062E4	35.0	86.94000	33.0	0.78668	41.625
43	3	47.0	0.64000	51.0	0.20492	31.0	75.38000	43.0	0.69035	43.000
44	398	38.0	0.82667	56.0	0.48079	45.0	154.10667	36.0	0.76865	43.750
45	WizSoft	42.5	0.78667	35.0	0.06701	56.0	557.98000	44.0	0.64550	44.375
46	589	44.5	0.72667	55.0	0.32017	38.0	94.35333	45.5	0.64391	45.750
47	544	49.5	0.52667	40.5	0.13206	47.5	364.07333	48.5	0.40547	46.500
48	548	49.5	0.52667	40.5	0.13206	47.5	364.07333	48.5	0.40547	46.500
49	545	49.5	0.52667	44.5	0.13329	49.5	375.52000	50.5	0.40466	48.500
50	cai cong zhong	49.5	0.52667	44.5	0.13329	49.5	375.52000	50.5	0.40466	48.500
51	26	56.0	0.07333	37.0	0.09668	55.0	416.32667	47.0	0.44952	48.750
52	546	52.5	0.52000	46.5	0.13337	51.5	389.48667	52.5	0.39686	50.750
53	554	52.5	0.52000	46.5	0.13337	51.5	389.48667	52.5	0.39686	50.750
54	ID16	54.5	0.46667	42.5	0.13241	53.5	398.24667	55.5	0.37974	51.500
55	581	54.5	0.46667	42.5	0.13241	53.5	398.24667	55.5	0.37974	51.500
56	264	57.5	0.02000	53.0	0.23782	42.0	114.82000	54.0	0.39570	51.625
57	187	59.0	0.01333	52.0	0.22880	57.0	810.53333	57.0	0.01727	56.250
58	Eric Group	57.5	0.02000	57.0	0.99076	58.0	851.46000	58.0	0.01453	57.625
59	476	46.0	0.68667	-	-	-	-	-	-	-

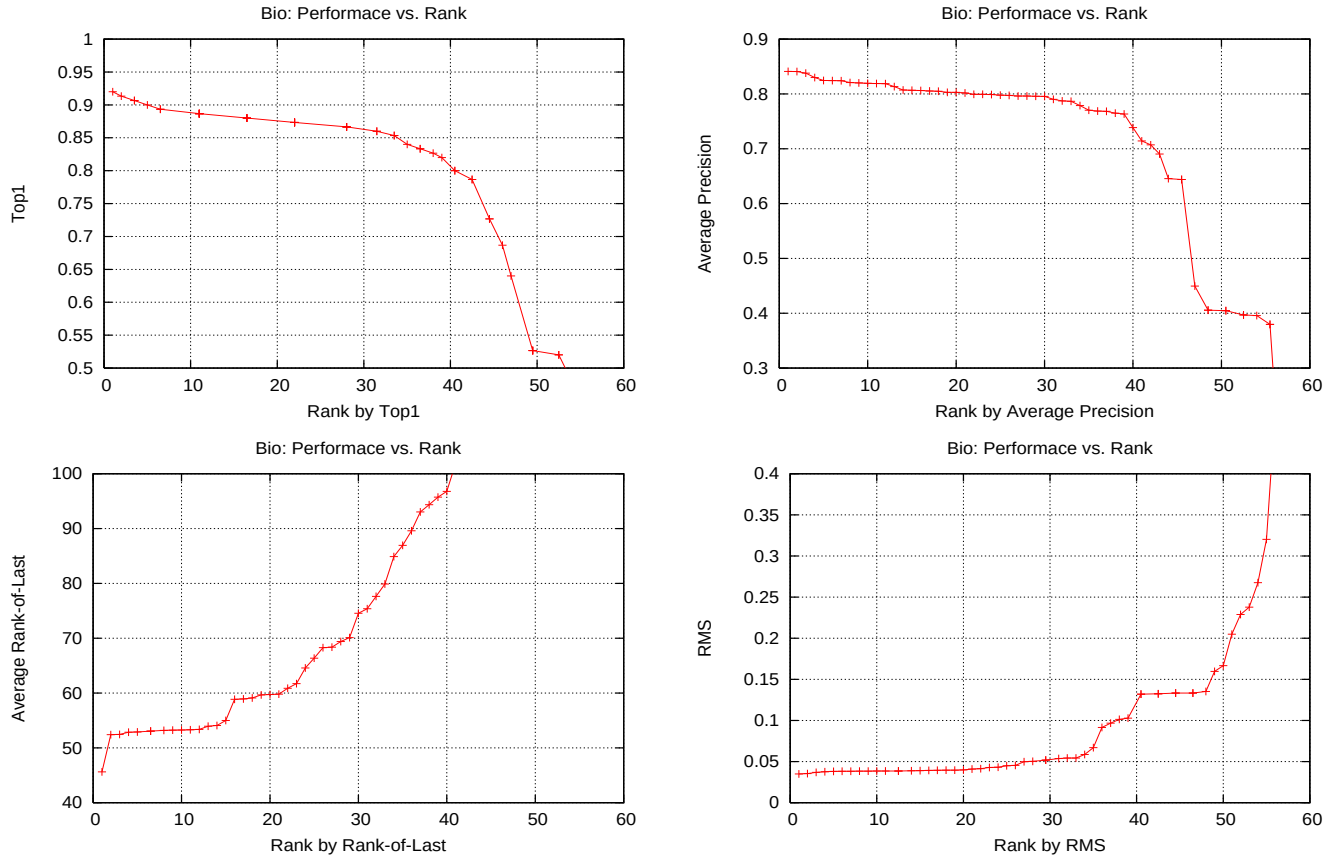

Figure 3: Performance of group vs. rank of group on Protein task.

middle of the plots are rather flat, indicating that a broad fraction of the participants achieved roughly the same performance. At around rank 40 the performance starts dropping rapidly on most measures[5]. This means that roughly 65% of the participants managed to achieve good performance on each of the tasks and metrics. The leftmost end of the graphs is particularly interesting. For many measures the graphs show a change in slope that indicates that the top performers pulled away from the pack and did substantially better than the majority of participants in midfield.

4.1 Bootstrap Analysis

To quantify the statistical significance of the performance gap among the winning predictions, we conducted a bootstrap analysis. This analysis allows us to evaluate how much the overall ranking of the groups on each of the tasks depends on our particular (random) choice of test sets. More precisely, it allows us to estimate the probability that a group would place differently in the competition if we repeated the competition with different randomly drawn test sets.

The setup of our bootstrap experiment was as follows. For r repetitions we drew a bootstrap sample from the original test set. For a test set of size k, this bootstrap sample is generated by drawing k examples from the original test set with replacement. We then evaluated the submitted predictions on the bootstrap sample and ranked all groups by overall performance (average rank across the four metrics for each task). Tables 3 and 4 show how often the 5 groups with the highest performance on the original test sets achieved a particular rank on bootstrap samples. For the Physics task we used $k = 100,000$ and $r = 1000$ repetitions. For the Protein task we took bootstrap samples over the $k = 150$ blocks in the test set with $r = 10,000$ repetitions.

As Table 3 shows, on the Physics task, the participant ordering derived from the original 100k test set is very likely to be the correct ordering. With high probability, the groups that placed first, second, and third on the original test set also placed first, second, and third, respectively, on bootstrap samples. We conclude that there is little uncertainty on the Physics task about which groups won the competition and we can be 95% confident that we have assigned first, second, and third place to the correct groups.

The results are very different for the Protein task. The first column of Table 4 shows that all three groups that scored highest on the original test set have a significant chance of winning first place on bootstrap samples. Surprisingly, the bootstrap analysis suggests that the group that placed second has the highest probability of being in first place. If we interpret the bootstrap results as a significance test, only once we go down to the fourth ranked group, can we conclude with 95% confidence that they did not win the competition independent of a particular test set. Based on this analysis, we declared a three-way tie for first place on

[5]An exception is the RKL measure, where the steep decrease in performance starts at around rank 20-25.

Table 3: Bootstrap Analysis of Results for Physics.

Overall rank on test set	Overall rank on bootstrap sample 1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}
1^{st}	100%	0%	0%	0%	0%
2^{nd}	0%	100%	0%	0%	0%
3^{rd}	0%	0%	94%	6%	0%
4^{th}	0%	0%	6%	93%	1%
5^{th}	0%	0%	0%	1%	76%

MEDai Inc. / Univ. of Central Florida David S. Vogel, Eric Gottshalk, Morgan C. Wang
- 1st Place Overall
- Honorable Mention for ROC Area
- Honorable Mention for Cross Entropy
- Honorable Mention for SLQ Score

Inductis Inc. Arpita Chowdhury, Dinesh Bharule, Don Yan,Lalit Wangikar, Sandeep Tyagi, Titiksha Gautam, Vineet Agrwal, Vivek Gupta
- 2nd Place Overall
- Honorable Mention for Accuracy

Golden Helix Inc. Christophe Lambert
- 3rd Place Overall

Figure 4: Winners and honorable mentions for Physics task.

the Protein task instead of declaring separate first, second, and third place winners as we were able to do on the Physics task.

All winners and honorable mentions are listed in Figures 4 and 5. A description of the approaches they used for the competition can be found in other papers of this issue of SIGKDD Explorations.

Table 4: Bootstrap Analysis of Results for Protein.

Overall rank on test set	Overall rank on bootstrap sample 1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}
1^{st}	14%	29%	26%	16%	8%
2^{nd}	59%	24%	10%	5%	2%
3^{rd}	22%	28%	23%	14%	7%
4^{th}	4%	12%	22%	26%	17%
5^{th}	0%	2%	6%	12%	20%

Univ. of Waikato Bernhard Pfahringer
- 1st Place Overall

Chinese Academy of Sciences Yan Fu, RuiXiang Sun, Qiang Yang, Simin He, Chunli Wang, Haipeng Wang, Shiguang Shan, Junfa Liu, Wen Gao
- Tied for 1st Place Overall
- Honorable Mention for Squared Error
- Honorable Mention for Average Precision

MEDai Inc. / Univ. of Central Florida David S. Vogel, Eric Gottshalk, Morgan C. Wang
- Tied for 1st Place Overall
- Honorable Mention for Top-1

Univ. of Dortmund Dirk Dach, Holger Flick, Christophe Foussette, Marcel Gaspar, Daniel Hakenjos, Felix Jungermann, Christian Kullmann, Anna Litvina, Lars Michele, Katharina Morik, Martin Scholz, Siehyun Strobel, Marc Twiehaus, Nazif Veliu
- Honorable Mention for Rank-of-Last Measure

Figure 5: Winners and honorable mentions for Protein task.

5. ANALYSIS OF THE RESULTS

5.1 Did Groups Effectively Optimize to Individual Performance Measures?

Most reasonable performance metrics are strongly correlated: predictions that yield good performance on one metric often yield good performance on other metrics as well. However, because different metrics reflect different tradeoffs between the predictions and ground truth, a prediction rule that is optimal for one measure is not necessarily optimal for a different measure. This was part of the motivation for this year's KDD-Cup. We wondered how much teams who submitted different sets of predictions for different metrics benefitted by optimizing to each metric. Did the winners win because they understood the data better and were able to train models that would have performed well on *any* metric, or were they able to gain additional benefit by separately optimizing their models for each metric?

About half of the competitors took advantage of the opportunity to submit different predictions for different performance measures. In the following we analyze their submissions to examine the extent to which teams improved performance by optimizing to specific performance metrics. Specifically, we evaluate whether the teams' efforts to optimize to individual performance metrics gave them higher scores or not. Clearly, if a team submitted the same predictions for all four metrics on one task, we do not gain any insight into whether that team could have benefitted from optimizing to the metrics. However, if we received 2 or more different sets of predictions from one team, we can study which set performs best on which performance metric. In particular, does the submission for a particular metric really outperform the predictions from the same team submitted for other metrics?

We first evaluate the team's performance on some metric, A, with the set of predictions that team gave us for metric A. Next, we take one of the other sets of predictions that the same team gave us for a different metric, and evaluate those predictions on metric A. If the team did better with the predictions submitted for metric A, this indicates that their optimization for metric A was effective. If, however, swapping the prediction sets improves performance on the metric, then clearly the team was less effective at optimizing to each metric and in fact would have done better if they had submitted their predictions for a different metric.

Tables 5 and 6 show how often swapping sets of predictions between pairs of metrics helps or hurts performance on those metrics. The columns in the tables show what metric the predictions were originally submitted for. The rows in the tables are the new metrics those predictions are used for. Each entry is the table is a pair of numbers. The positive number is the number of times swapping metrics helped per-

Table 5: Cross metric score for Physics.

Tested on	Submitted for ACC	CXE	AUC	SLQ
ACC		+9,-9	+6,-7	+8,-8
CXE	+4,-16		+0,-17	+3,-12
AUC	+4,-9	+8,-9		+8,-8
SLQ	+4,-12	+8,-7	+5,-9	

Table 6: Cross metric score for Protein.

Tested on	Submitted for APR	RKL	RMS	TOP1
APR		+14,-10	+14,-15	+5,-11
RKL	+6,-18		+1,-18	+7,-19
RMS	+1,-27	+1,-28		+2,-29
TOP1	+6,-6	+13,-9	+12,-16	

formance. The negative entry is the number of times swapping performance hurt performance. If competitors always achieved the best performance on each metric using the predictions they submitted for that metric, all entries would be negative.

From the tables we can see that it is fairly common that a team would have achieved better performance on some metric by using predictions they had submitted for a different metric for that metric instead. In fact, the predictions groups submitted for rank last (RKL) more often than not would have been better predictions for average precision (APR) and TOP1 than the predictions submitted for those metrics. On average, however, swapping hurts performance more than it helps. On the Physics task, swapping submissions helped 67 times but hurt 123 times. On the Protein task, swapping helped 82 times but hurt 206 times.

Unfortunately, swapping sets of predictions is not always sensible: for many pairs of metrics, a good submission for one metric might simply be inappropriate for another metric. For example, a good set of predictions for TOP1 might have a single prediction of class 1 for one case in each block, while the rest of the cases in that block might be predicted as 0. This is a perfectly reasonable set of predictions for TOP1. But these same predictions are unlikely to give a good RMSE score. An even more catastrophic example comes from ordering metrics such as APR and AUC. For these metrics, only the ordering induced by the predictions matters. The predictions can be any numbers (real or integer) on any scale, as long as they provide a well-defined ordering. Since predictions for RMSE, CXE, and SLQ must be between 0 and 1, some APR and AUC predictions can not be used for RMSE, CXE, or SLQ. There are additional instances of predictions that make sense for one metric, but are incompatible with another metric.

Luckily, most teams submitted predictions between 0 and 1 that were more likely to be compatible across performance measures. In order to eliminate invalid swaps, we employ the following test. We define as δ the change in rank resulting from substituting some other set of predictions for the intended set. If the rank of the team on the metric we were testing increased or decreased by more than δ_{max} when we used the predictions from a different metric, then we considered the swap to be invalid. For example, with a δ_{max} of 10, if we found that using a team's APR predictions for RMSE

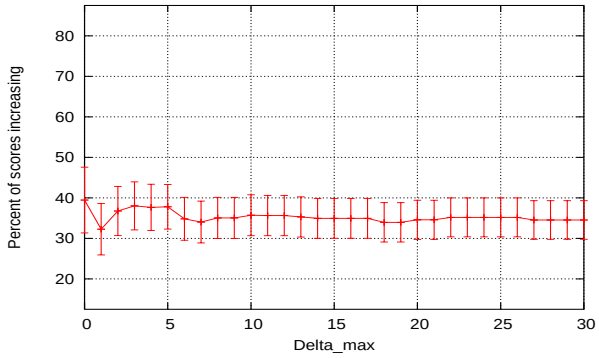

Figure 6: Percentage of swaps that increase performance vs. δ_{max} for Physics.

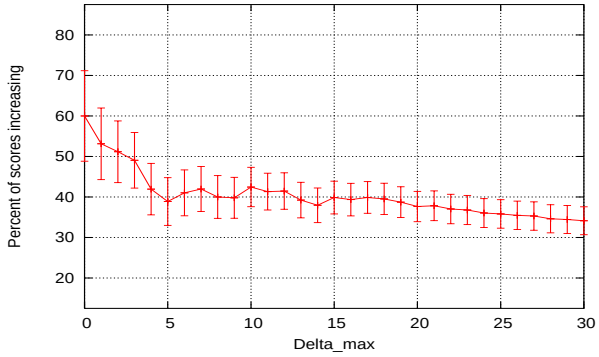

Figure 7: Percentage of swaps that increase performance vs. δ_{max} for Protein.

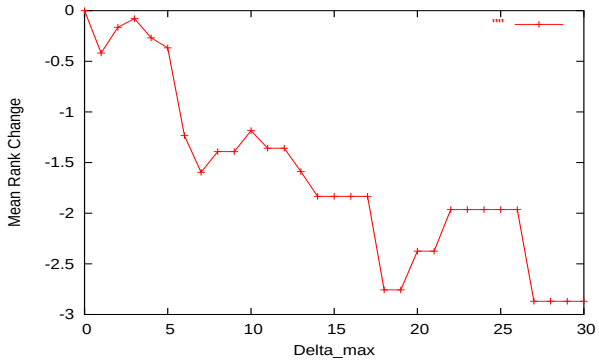

Figure 8: Mean rank change vs. δ_{max} for Physics.

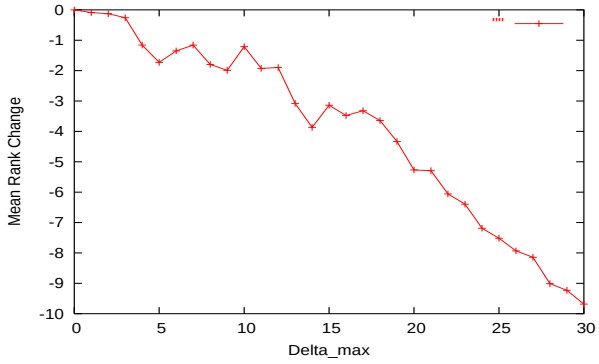

Figure 9: Mean rank change vs. δ_{max} for Protein.

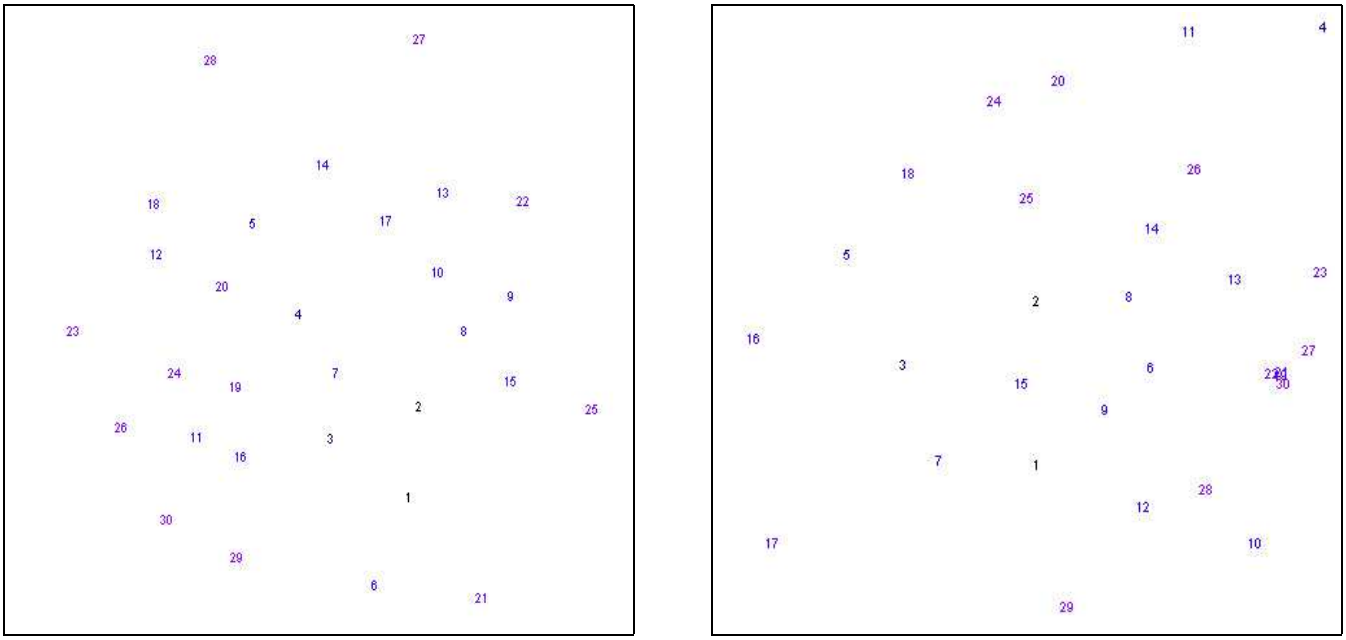

Figure 10: Two dimensional representation of AUC and APR predictions. Normalized stress is 0.08 for AUC and 0.09 for APR.

caused that team's RMSE rank to drop from 5th to 30th, we assumed that those APR predictions were not acceptable as RMSE predictions. This approach assumes that when a set of predictions is incompatible with a performance metric for which it was not submitted, large drops in performance will be observed, enabling us to filter out the incompatibilities.

To avoid bias, if a team's rank on a metric *increased* by a large amount when using predictions for another metric, we supposed that the predictions the team originally gave us for that metric were somehow flawed, and so we did not include those cases either. For example, most teams prevented predictions from reaching 0 or 1 for the CXE metric because even a single misclassification would then cause a near-infinite loss in cross-entropy. But a number of teams did submit predictions for CXE that were 0 or 1. For these teams, the predictions they submitted for AUC might actually yield better CXE than their CXE predictions. We exclude these cases because clearly they have made major mistakes in the predictions they submitted for one or more metrics.

Finally, as we saw in the previous section, the teams towards the bottom third of the rankings did much worse than the top two-thirds of the teams. We are mainly interested in the effect of optimizing to a metric for groups that were able to achieve good performance on these problems, so we did not include in our analysis those predictions that performed very poorly.

Once we had determined which sets of predictions were reasonable to swap, we did a swap analysis for each problem. The number of valid swaps varies with δ_{max}, but there were over 100 on each problem even for moderate values of δ_{max}. For each swap we determined if the swap gave a better performance or a worse performance than the original predictions on that metric. We counted the total number of score increases, *inc*, and the total number of score decreases, *dec*, and then computed the fraction of times that swapping improved performance as opposed to hurt performance. Figure 6 and Figure 7 show the probability that swapping improved performance plotted against δ_{max}, using the top 40 predictions on each metric.

On the physics problem, the probability that swapping prediction sets helps performance is relatively constant at about $p = 0.35$. This is well below 0.5, and the error bars do not include 0.5. This indicates that regardless of δ_{max}, swapping caused decreases in performance significantly more often than swapping caused increases. This means that on the physics problem, groups that optimized to each performance metric improved their performance on the physics metrics at least 65% of the time by doing this optimization.

The protein problem has a slightly different story. For small values of δ_{max}, the graph is actually above 0.5, which indicates that using predictions for an alternate metric tended to help a little in cases where the change in rank (delta) was very small. Note, however, that the error bars for all points above 0.5 include 0.5, so this might be just statistical fluctuation. Also, the swapping score is not as informative for small values of δ_{max}. For example, when δ_{max} is 0, it means that the rank remains the same when using alternate predictions. For the rank to remain the same, the change in score must have been very small. As δ_{max} increases from 0, the fraction of cases for which swapping improves performance quickly drops to 0.4, and for moderate values of δ_{max}, the probability that swapping helps is similar to that of the physics problem, about 0.35. We conclude that most participants who submitted different predictions for different metrics on the Protein task were able to achieve better performance on each metric by optimizing to each metric.

Up to this point, we have only counted the number of swaps that cause scores to increase or decrease. We showed that making a swap does tend to decrease performance, suggesting that the groups were effective at optimizing to each metric. But how big is the difference? Unfortunately, many of

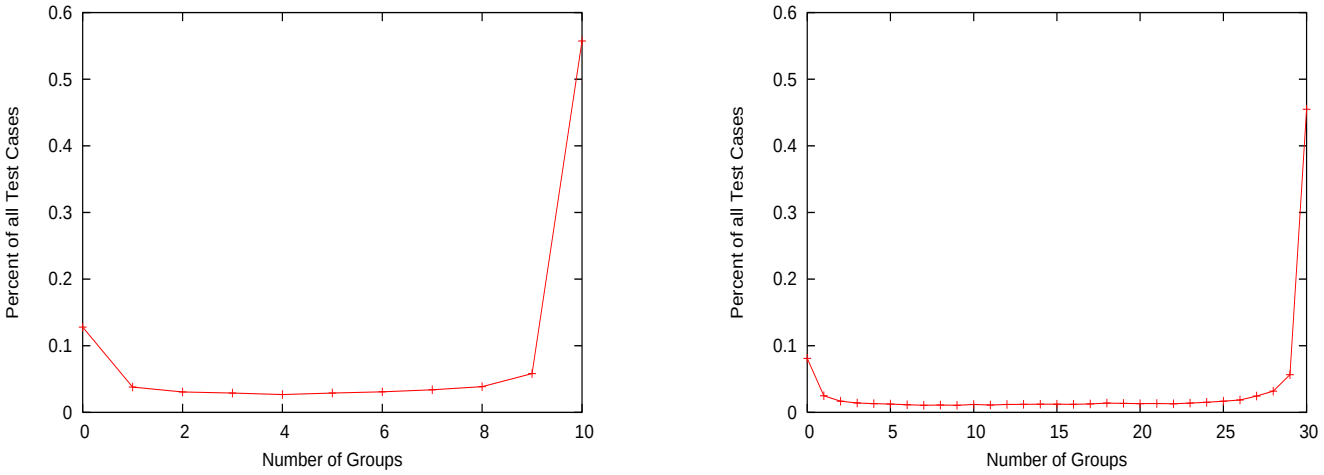

Figure 11: Consistency of predictions among the top 10 (left) and top 30 (right) groups on the Physics task. The y-axis show fraction of test examples. The x-axis shows the number of groups that classify a particular example correctly.

the metrics are non-linear and on different scales, so a simple average of the change in score is not informative. Instead, we will look at the mean rank change caused by swapping. Figure 8 and Figure 9 show the mean rank change as a function of δ_{max}. The plots suggest that, while optimizing to a particular metric does give some benefit, this benefit is typically modest. On the Physics task, swapping submissions between metrics on average lowers a group's rank on that metric only about 1-2 places. On the Protein Task, swapping predictions between metrics would cause a group to rank about 2-7 places lower on each metric. These differences in performance can be substantial in a competition where ranking a few positions lower can be the difference between coming in first and not even being in the top three, but the changes in performance that yield a decrease in rank of 1-7 places typically is rather small[6]. Furthermore, roughly half of the highest scoring competitors did not submit multiple sets of predictions, indicating that optimizing to the particular performance measures was not essential for performing well.

5.2 How Different are each Group's Predictions?

Do good predictions for a particular metric tend to be similar? That is, given two high-scoring sets of predictions for a problem and metric, are the predictions very similar? To answer this question, we looked at the top 30 submissions on 4 different performance metrics. We treated each submission as a vector in Euclidean space, and determined the Euclidean distance between the two vectors of predictions for each pair of submissions. To normalize predictions for the rank based measures (e.g. APR and AUC), we sorted the predictions, and converted each prediction to be its rank, divided by the total number of predictions. After calculating the matrix of all pairwise distances between submissions for a metric, we examined the distance matrix. To visualize the distances, we use Multi-Dimension Scaling (MDS). Projecting the data down to two dimensions with MDS reduces normalized stress to below 0.1, suggesting that the predictions submitted by the top 30 competitors for a metric vary along a low-dimensional manifold.

Figure 10 shows the MDS plots for Physics AUC and Protein APR. Surprisingly, they show that good predictions need not be very similar to each other. Moreover, predictions that are somewhat similar to each other can have surprisingly different performance. For example, in the APR plot, the top 3 sets of predictions are relatively close to each other near the center of the plot. But the distance between the 25^{th} submission and the 2^{nd} submission is less than the distance between the 1^{st} submission and the 2^{nd} submission[7]. Furthermore, notice that the 4^{th} place predictions are all the way up in the upper right corner, far away from the 1^{st}, 2^{nd} and 3^{rd} place predictions. This shows that models with excellent performance do not always achieve that performance by making similar predictions, and models with fairly similar predictions do not always achieve comparable performance.

5.3 The Easy, the Dif£cult, and the Impossible

The previous section showed that the predictions of the best performing groups were not homogeneous. In this section we examine how these differences are distributed among the test examples. Figure 11 plots the fraction of test examples that a particular number of groups predicted correctly for the Physics accuracy task. The left-hand plot includes the 10 groups with the highest accuracy on the Physics task, the right-hand plot includes the 30 groups with the highest accuracy.

Most groups agree on the classification of a large fraction of the examples. On roughly 55% of the test examples, all top 10 groups make the same prediction and classify the example correctly (right-most point of the left-hand graph). Even when considering the top 30 groups, the fraction remains high with 45% of the test cases being correctly classified (and therefore classified the same) by all groups. Interestingly, the predictions also are consistent on many incorrectly classified examples. On about 12% of the test examples the

[6]See Tables 1 and 2 for an idea of how large a difference in performance must be to move several positions in the rankings.

[7]We have verified by looking at the raw Euclidean distances that this is real and not just an artifact of the 2-d MDS projection.

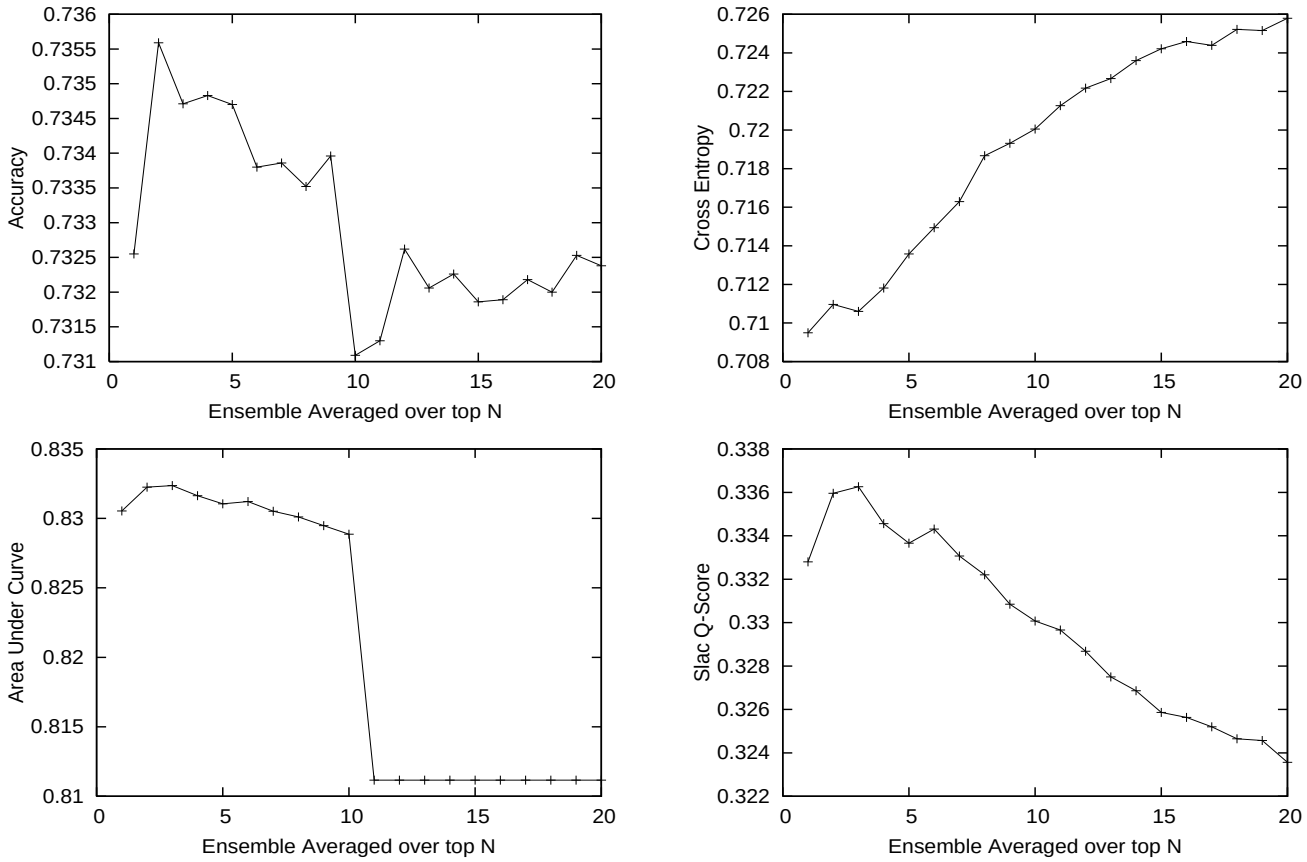

Figure 12: Performance of ensemble on Physics task.

top 10 groups all predict the wrong class (left-most point of the left-hand graph). Again, this figure changes little when including the top 30 groups.

In summary, the graphs show that there is a large fraction of test examples that all groups agree upon in their predictions. The top 10 groups agree on the prediction for 67% of the test cases, and even the top 30 groups agree on more than 50% of the test cases. However, 8% to 12% of the test cases are consistently misclassified. Between the two extremes, the graph is rather flat. This means that only a fairly small fraction of the test cases account for the differences in prediction that were observed for different groups in previous sections.

6. ARE WE HITTING THE CEILING?

Because the Physics and Protein tasks are real-world problems (as opposed to synthetic problems), we do not know what performance ultimately is achievable on each task and metric. On most metrics, the top models have performance close to each other. This might suggest that the best models are at or near the ceiling and better performance is not possible. In this section we try to determine if better performance can be achieved on these problems, or if the best models are already near the ceiling.

An ensemble is a collection of models whose predictions are combined by weighted averaging or voting. Dietterich[3] states that "A necessary and sufficient condition for an ensemble of classifiers to be more accurate than any of its individual members is if the classifiers are accurate and diverse." Recent work shows that models trained with different learning algorithms often make uncorrelated errors. When this is true, an ensemble of good models trained with different learning algorithms often outperforms the best model trained by one of the learning algorithms.[2]

We wondered if ensembles of the best models submitted for each task and metric would improve upon the performance of these best models. To answer this question, we created ensembles that average the predictions of the best N models for each task and metric. To focus our attention on the best models, we vary N from 1 to 20. We then evaluate each of these ensembles on the final test set using the appropriate performance metric.[8]

Figure 12 shows the performance of ensembles formed by averaging the predictions of the best N models for the four Physics metrics. Figure 13 shows the performance of ensembles that combine the best N models submitted for the Protein metrics. The x-axis is the number of models included in the ensemble. The first ensemble ($N = 1$) in each of the eight plots contains only the single best model submitted for that task and metric. Thus $N = 1$ is the performance of the winning model submitted in the competition for each metric. At $N = 2$, the ensembles average the predictions of

[8]Because we use the same test sets to find the best N models, and then to evaluate the performance of ensembles containing these best N models, we do not have truly independent test sets.

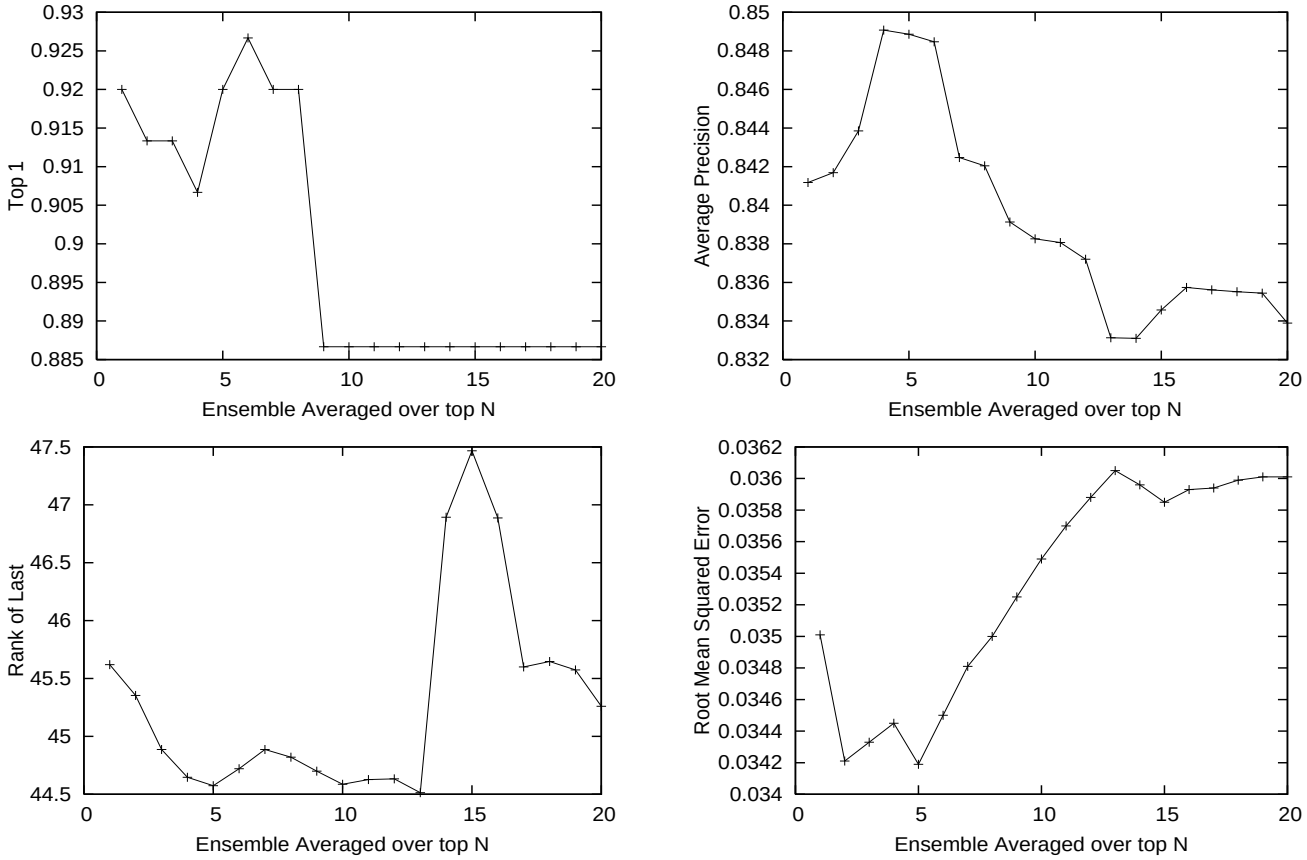

Figure 13: Performance of ensemble on Protein task.

the best two models submitted for that task and metric.

In seven of the eight plots, better performance is achieved for N greater than one. On most problems and metrics, better performance can be achieved by combining the predictions of a few of the best models. For example, on the physics problem, ensembles that average the top 2-9 models all achieve higher accuracy than the best single model. The best accuracy is achieved for an ensemble that combines the top two models. This ensemble yields an accuracy of 73.56%, up 0.30% from the best submitted model which has accuracy 73.26%. While this increase is modest, it is four times larger than the 0.07% difference in performance between the top two submitted models.

Similarly, the best AUC and the best SLAC-Q Score are achieved by averaging the best three models submitted for these metrics. On the cross-entropy metric, however, the best submitted model outperforms any of the ensembles, presumably because there is a large gap in cross-entropy performance between the best model and all other models.

On the protein problem, an ensemble outperforms the best submitted models on each of the four metrics. An ensemble of the best six models increases Top 1 performance from 0.920 to 0.927. An ensemble of 5 models lowers RMS from 0.0350 to 0.0342. An ensemble of the best 13 models lowers Rank of Last from 45.6 to 44.5. And an ensemble containing the top 4 models increases average precision from 0.841 to 0.849. While all of these increases in performance are modest, for three of the four metrics they are equal to or larger than the differences in performance between the best two models submitted for that metric. On Rank of Last, the best model ($RKL = 45.62$) is dramatically better than the second best model ($RKL = 52.42$), and we were surprised to see that ensembles combining the best 2-13 models all improve upon the best single model.

In summary, on 7 of 8 metrics, ensembles that average the predictions of the best N models outperform the best submitted models by margins comparable to the differences in performance we see between the best two submitted models. The best number of models to include in each ensemble depends on the task, metric, and submissions. To be fair, the decision about the best N should not be made using the final test set as we have done here, so the results we present must be taken with a grain of salt. But the graphs do seem to suggest that on both problems and most metrics there still is room for improvement, and the submissions for the competition have not hit the ceiling. As further evidence that we have not achieved peak performance on these problems and metrics, we have received several submissions since the competition closed that improve upon the best performances observed during the competition. See `http://kodiak.cs.cornell.edu/kddcup` for the latest results on each problem.

7. ACKNOWLEDGMENTS

We would like to thank the contributors of the datasets, Ron Elber (Cornell CS) and Charles Young (SLAC), for the time

and effort they invested in creating the data and helping us prepare it for the KDD-Cup 2004. We also thank Johannes Gehrke (Cornell CS) and Mirek Riedewald (Cornell CS) for help with the Physics dataset and SLQ metric. We thank Alex Niculescu, Filip Radlinski, and Claire Cardie for their help with the PERF evaluation software, and also thank the participants from the University of Dortmund and the Chinese Academy of Science who detected bugs in early releases of PERF. This work was partially supported by NSF grants IIS-0412894 and IIS-0412930.

8. CONCLUSION

We presented the tasks and the winners of the 2004 KDD-Cup competition. Our analysis of the results revealed that roughly two thirds of the participating groups found good solutions. While there was evidence that some participants did benefit by optimizing to the different performance measures, the benefits typically were modest, and on average would change their ranks only a few places. Comparing submissions from different groups, we found a substantial amount of diversity in the predictions from different groups, yet determined that much of this diversity occurs on less than 50% of the test cases. The results of an ensemble learning experiment confirms that there is useful diversity among the top competitors, and gives some evidence that it is possible to achieve somewhat better performance than the winning submissions.

While the original KDD-Cup 2004 competition is officially closed, the datasets and a new submission interface remain available on the KDD-Cup WWW site:

`http://kodiak.cs.cornell.edu/kddcup`.

New submissions will be scored immediately after submission and the results are inserted into an expanding table of post KDD-Cup results. A count of the total number of new submissions a group makes for each task and metric is displayed in this table to help prevent groups from overfitting to the test sets by testing too many models. We encourage further participation and research on the tasks of the KDD-Cup 2004.

9. REFERENCES

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley-Longman, Harlow, UK, May 1999.

[2] R. Caruana and A. Niculescu-Mizil. Ensemble selection from libraries of models. In *Proc. 21th International Conference on Machine Learning (ICML'04)*, 2004.

[3] T. G. Dietterich. Ensemble methods in machine learning. *First International Workshop on Multiple Classifier Systems*, pages 1–15, 2000.

About the authors:

Rich Caruana is an Assistant Professor in the Department of Computer Science at Cornell University. He received a Ph.D. in CS at Carnegie Mellon University in 1997 where he was advised by Tom Mitchell and Herb Simon. Most of Caruana's research is in machine learning and data mining. His main focus is on developing new learning methods for problems in medical informatics, though he is interested in most real-world problems that stretch the capabilities of current machine learning methods. A recent focus of his work is optimizing supervised learning methods for different performance criteria.

Thorsten Joachims is an Assistant Professor in the Department of Computer Science at Cornell University. In 2001, he finished his dissertation on maximum-margin approaches to learning text classifiers, advised by Prof. Katharina Morik at the University of Dortmund. From there he also received his Diploma in Computer Science in 1997 with a thesis on WebWatcher, a browsing assistant for the Web. His research interests center on a synthesis of theory and system building in the field of machine learning, with a focus on Support Vector Machines and machine learning with text. He authored the SVM-Light algorithm and software for support vector learning. From 1994 to 1996 he was a visiting scientist at Carnegie Mellon University with Prof. Tom Mitchell.

Lars Backstrom is a recent graduate of Cornell University's computer science department. He is currently researching feature selection for supervised learning with Professor Caruana at Cornell University. Additionally, he is the problem coordinator for TopCoder, Inc., a company provding programming competitions as a recruiting tool. He is in the process of applying to Ph.D. programs in computer science for the fall of 2005.