Estimating The Potential for Combining Learning Models

Adam H. Peterson ADAM@AXON.CS.BYU.EDU
Department of Computer Science, Brigham Young University, Provo, Utah 84604 USA

Tony Martinez MARTINEZ@CS.BYU.EDU
Department of Computer Science, Brigham Young University, Provo, Utah 84604 USA

Abstract

This research describes the COD (*Classifier Output Difference*) distance metric for finding similarity between hypotheses and learning algorithms. This metric is a tool which can be used to measure how similar two hypotheses are. It goes beyond simple accuracy comparisons and provides insights about fundamental differences between learning models. This paper describes how COD works and shows how it can be used to predict the potential for combining hypotheses in an ensemble to improve accuracy.

1. Measuring hypothesis similarity

1.1. Background

Machine learning is the field of study concerned with understanding and creating computational processes that learn. (A process can be said to learn a task if its measured performance on a task improves with experience or data.) Ideally, it would be desirable to fashion computer programs which could solve any task set to them.

As a consequence of the *No Free Lunch Theorem*, learning can not occur without constraining the class of problems upon which learning is to be performed. The problems that the field of machine learning centers on are thus not all possible problems, but the set of problems likely to be of import to ML practitioners (or their clients). We will call this set of machine learning tasks P_I, the "interesting problems." In contrast to the set of all problems (which include a vast number of random problems with no regularity), these problems exhibit some regularity. Unfortunately, P_I is difficult to characterize formally.

Several machine learning algorithms and algorithm families have been specified and developed by the machine learning research community (including neural networks, decision trees, instance based learners, rule based systems, induction systems, etc.). Each machine learning algorithm makes some assumptions about the problems it is applied to, and performs most effectively on problems where those assumptions are more or less satisfied. These assumptions bias the learner towards one solution or another, and this bias is a necessary part of learning (Mitchell, 1980). A fuzzy concept of "coverage" for each learning algorithm can be conceived of as the set of problems on which the learning algorithm performs well. The *No Free Lunch Theorem* guarantees that a learning algorithm or small set of learning algorithms will not be able to "cover" the entire space of possible learning tasks. On the other hand, it is only coverage of P_I that is desired.

It is an open question whether the current state of the art in machine learning has good coverage over the set of interesting problems. For many tasks, good accuracy may be achieved with one or more of the learning algorithms or algorithm families commonly used. With others, even the best current learning algorithms do not perform as well. Sometimes a combination of two or more algorithms may be able to perform much better than any of the algorithms individually. Dynamically selecting and combining learning algorithms or hypotheses is a well-known approach in *meta-learning* (Vilalta & Drissi, 2002), and can result in algorithms whose effective applicability extends across the applicability of the base algorithms, and in some cases beyond.

One family of approaches to combining learning algorithms is ensembles (Opitz & Maclin, 1999). Ensemble techniques use a hypothesis composed of several simpler hypotheses whose output is combined to solve a learning task. In some ensemble approaches, hy-

Appearing in *Proceedings of the ICML-2005 Workshop on Meta-Learning*, Bonn, Germany, 2005.

potheses produced by several different learning algorithms are combined, while in others it is not uncommon to use the same learning algorithm to produce the hypotheses so combined. Many variations exist for ensemble techniques. Bauer and Kohavi (1999), for example, discuss several approaches to Bagging and Boosting.

This research describes the *COD* (*Classifier Output Difference*) distance metric between hypotheses, a tool that can be used to combine individual hypotheses effectively by determining which hypotheses have the most potential for improvement through combination.

Although there are a variety of types of machine learning tasks (agent decision problems, regression, etc.), this research will focus on classification problems.

1.2. Concepts

A *learning task* is a problem from P_I. In other words, it is a problem that someone is likely to want solved at some point. A *learning algorithm* (or simply *algorithm*) is a procedure that, when given a learning task, produces a *hypothesis*, or a model that attempts to solve the learning task. In this research, *hypothesis* and *classifier* will sometimes be used interchangeably, since it is constrained to classification problems.

The COD distance between two hypotheses is the frequency (a real value between 0 and 1) with which they disagree on the classification of patterns. This distance between two hypotheses over a particular data set can be estimated by observing the frequency that the hypotheses disagree with each other on the classification of the patterns from the given set.

2. Related Work

Theoretical research by Almuallim and Dietterich (1992) introduces concepts of algorithm coverage and concept distance which are defined in a theoretical sense by their properties spanning over the feature space. The COD distance is a practical realization of this concept distance. It has several useful features that do not occur in Almuallim and Dietterich's distance. First, COD is empirically computable, while Almuallim's metric is for all but trivial feature spaces prohibitive to compute. Second, COD weights patterns by their likelihood of occurrence in the feature space so COD distances are a measure of the expected frequency of actual disagreement. This is especially valuable when it is noted that two hypotheses may behave similarly in general, but extrapolate very differently in regions of the input space that rarely or never occur. Almuallim's metric weights all input feature instances equally, even instances that would never occur for an actual problem. Third, Almuallim's results are artificial and of theoretical import, while the COD distance is firmly tied to empirical measurement.

Distances between hypotheses have been explored in Kuncheva and Whitaker (2001), who give several measures based on the frequency that classifiers label patterns correctly or incorrectly together. One area where the COD metric differs from these measures is the case where both classifiers are incorrect but may still give differing outputs. Brown et al. (2005) also discuss diversity in ensembles, giving examples of metrics to measure diversity and surveying methods to encourage diversity. Ali and Pazzani (1995) observe that for a set of classifiers to form an effective ensemble, error correlation between the classifiers should be low. Negatively correlated errors were shown to give some of the strongest improvements.

The effectiveness of negative error correlation in improving ensembles was also observed by Liu and Yao (1999). Their work adjusts neural network training to encourage the set of classifiers to have negatively correlated errors. Another approach to combining diverse neural networks is taken by Huang et al. (1995), who combine the output of several "expert" networks with a single combining network, in a stacking approach. Margineantu and Dietterich (1997) recommend two methods for improving the accuracy of ensembles by judicious selection of component hypotheses. Kappa pruning selects models based on an agreement statistic. Reduce-Error pruning chooses hypotheses to maximize voting performance on withheld patterns.

The quantification of distances between hypotheses has tie-ins with *meta-learning*, where machine learning strategies are applied at more than one level in the problem solving process. This idea is used in *landmarking*, put forward by Pfahringer et al. (2000). Landmarking is in some respects the inverse approach of COD. Whereas in landmarking, learning tasks are measured relative to some simple learning algorithms, COD measures distances between hypotheses by using their performance on a learning task.

A more traditional approach to learning task similarity is taken in Zheng (1993) by positing several simple metrics that may be used to categorize learning tasks (e.g. number of inputs and whether all inputs are real-valued). Zheng's approach does not apply meta-learning ideas, and also tries to solve the inverse problem to COD.

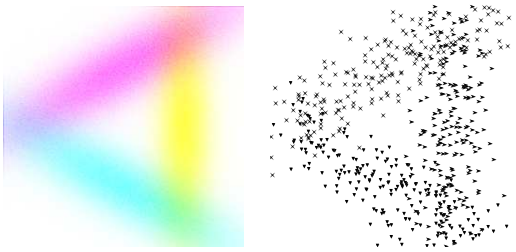

Figure 1. (Left:) An artificial example of a classification learning task with two real-valued input features and three output classes (each normally distributed in the feature space). (Right:) A sampling of the three classes plotted with three different monikers (for grayscale printouts).

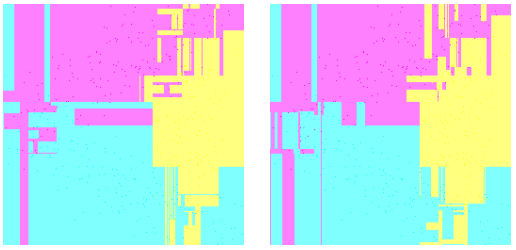

Figure 2. The decision surfaces for two different hypotheses (generated by two different decision tree learning algorithms) on the problem given in figure 1.

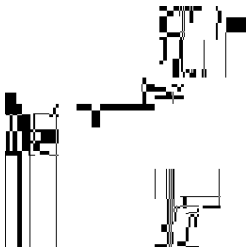

Figure 3. The regions where the two hypotheses from figure 2 disagree.

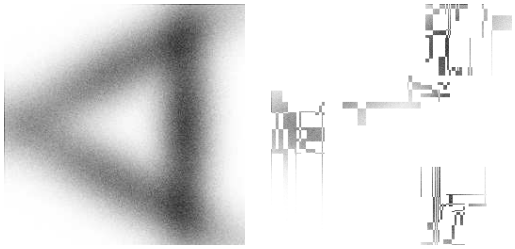

Figure 4. (Left:) The probability of each input occurring in the input space (darker values are more probable). (Right:) The regions from figure 3 weighted by the probability of the input occurring. The proportion of the overall coverage of the right image against the coverage on the left is the COD distance metric.

3. The distance between classifiers

The objective of COD is to measure an estimate of the frequency that two hypotheses (or, by extension, two learning algorithms) will exhibit different behaviors on a task (or set of tasks). For the distance between two individual hypotheses, ideally it would be desirable to integrate the output difference between the two hypotheses across all possible inputs, weighted by the input probability. That is, the ideal COD distance metric over a learning task would be computed something like this:

$$D(H_1, H_2) = \int_{\boldsymbol{x} \in X} |H_1(\boldsymbol{x}) - H_2(\boldsymbol{x})| \cdot \mathrm{P}(\boldsymbol{x}) d\boldsymbol{x} \quad (1)$$

where $\boldsymbol{x}$ is each possible input from the input space X, the quantity $|H_1(\boldsymbol{x}) - H_2(\boldsymbol{x})|$ is 1 when hypothesis 1 outputs a different value from hypothesis 2 and 0 otherwise, and $\mathrm{P}(\boldsymbol{x})$ is the probability of input $\boldsymbol{x}$ occurring for the desired task.

For example, consider the artificial classification problem shown in figure 1. The decision regions for two hypotheses trained on this problem are given in figure 2. The regions of the input space over which the hypotheses disagree is shown in figure 3. The proportion of the input space covered by these regions then forms the basis for the COD metric, after weighting the covered regions by the probability of each input occurring, as shown in figure 4.

Actually performing the integration given above is impractical (and in most cases impossible) for at least two reasons:

- In most cases, the input space X is of large dimension and/or contains a large number of possible inputs for each feature. Tasks where this is not the case tend to be academic in nature. Although they are properly part of P_I, they form a relatively small portion even when taken together.
- For a large number of learning tasks (if not most), the likelihood of an input vector for the task ($P(\boldsymbol{x})$) is not precisely known. Although this can be estimated in some cases, such estimations rely on assumptions which may apply imperfectly or not at all. Inaccuracies in the estimation can easily compound over a large integration to result in an inaccurate distance metric.

COD presents a method for estimating this distance measure between hypotheses in a practical manner, based on differences in their output behaviors. Two

hypotheses may be compared using COD by evaluating each hypothesis on patterns that neither has seen during training and comparing their outputs. The number of patterns disagreed upon is counted and divided by the number of patterns used in the comparison, giving the COD metric value. Taking the typical training set/testing set methodology (where a data set gathered from a task is divided into two disjoint sets, one used for training hypotheses and one used for evaluating their accuracy), the test set may be used to instead find the distance between two hypotheses trained on the same training set (or any training set disjoint from the testing set). Assuming the test set is representative of the learning task, the COD metric can be estimated as:

$$\hat{D}_T(H_1, H_2) = \frac{\sum_{\boldsymbol{x} \in T} |H_1(\boldsymbol{x}) - H_2(\boldsymbol{x})|}{|T|} \qquad (2)$$

where T is the test set.

Measuring distance between two hypotheses on a problem in this way includes the following beneficial effects:

- The measurements are always taken of the behavior of the hypothesis on data that was not used in the training process, so the measurement reflects the behavior of the hypothesis during generalization rather than its idiosyncrasies in handling training data. By extension, the measurements will tend to reflect the generalizing behavior of hypotheses produced by the learning algorithm when several such measurements are aggregated. Disagreement between algorithms on training point classification may be interesting for some types of analysis. For these experiments, however, it is discounted in favor of generalization behavior.

- The data are a sampling from the distribution likely to occur during generalization, so the distance metric tends to reflect the difference in behavior likely to be observed during actual use. One might be tempted to instead train a hypothesis on all of the given data and then measure the distance by sampling manufactured patterns across the input domain of the problem and observing how often the two hypotheses disagree on the output. Such a measure would be less relevant since it would represent an integration across the input domain distributed substantially differently than is likely to occur during generalization.

- The distance is nonzero only when the hypotheses have different *observable* behaviors at the output. If two hypotheses give the same answer most of the time, even though they perform very different computations internally to arrive at their result, the internal differences are largely irrelevant for most purposes employing a hypothesis or algorithm distance (such as determining coverage over a set of learning tasks or measuring diversity for an ensemble).

Instead of using the basic training set/testing set method as a base, n-fold cross-validation may also be used (Dietterich, 1998). In the standard application of cross-validation, multiple hypotheses are trained and tested on rotating subsets of the data set such that each portion of the data set is tested against once (using a classifier not trained on that portion). The COD metric can be similarly measured between two learning algorithms by obtaining a pair of hypotheses, one from each algorithm, over each partitioning of the data set and finding the pairwise distance between the corresponding hypotheses. The individual distances so measured can then be averaged to obtain an estimation of the general distance between two learning algorithms.

In contrast to accuracy measurement methods (such as training set/testing set and cross-validation) which only operate against one hypothesis or learning algorithm, COD operates against a pair of such at a time (to find the distance between them). Where accuracy measurements gathered in such a way during research are often averaged to give an indication of the accuracy of the learning algorithms over a set of learning tasks, COD metrics are averaged to give a *matrix* describing the general distances between learning algorithms over a set of tasks (with an entry corresponding to each pair of algorithms and giving the average distance between them). Such a matrix can then be used in analysis to determine the type of coverage a set of learning algorithms has over the set of tasks.

The COD metric satisfies the triangle inequality between more than two hypotheses. It also satisfies the triangle inequality between two hypotheses and the "perfect classifier" (that is, the hypothetical classifier that gets every example correct). The distance between two hypotheses and their distance to the "perfect classifier" (that is, the hypotheses' error rates) can be used to form an abstract triangle, as visualized in figure 5. By applying the Law of Cosines, a concept of angle between two hypotheses (or learning algorithms, when used in aggregate) can be defined.

The angle value from this triangle may be used as an indication of useful diversity between two hypotheses. When hypotheses perform comparably on a problem while behaving differently (as pictured in figure 6),

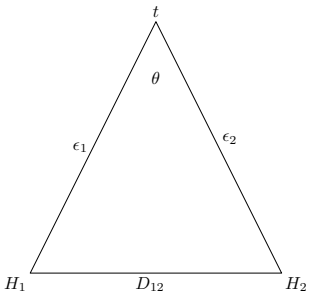

Figure 5. A representation of the triangle formed by the COD metric between two hypotheses and the "perfect classifier." ϵ_1 and ϵ_2 are the errors of H_1 and H_2 on task t. D_{12} is the COD distance between H_1 and H_2. The sides of this triangle are used to find θ.

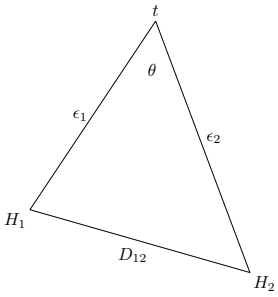

Figure 6. A COD triangle for two hypotheses that are relatively diverse, giving a fairly large value for θ.

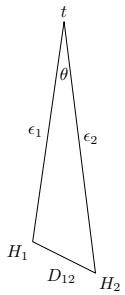

Figure 7. A COD triangle for two similar hypotheses, giving a relatively small value for θ.

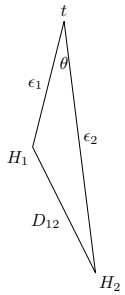

Figure 8. A COD triangle for two hypotheses where one hypothesis significantly dominates the other.

potential for improvement exists by combining the strengths of the two hypotheses. The different behaviors of two such hypotheses are the result of either the hypotheses misclassifying different examples or misclassifying the same examples while giving different wrong answers.

On the other hand, if there is not much behavioral difference between two hypotheses relative to their error rates (giving a smaller angle, as in figure 7), the potential for improvement by combining the hypotheses (or algorithms used to produce them) is less, as improvement tends to center around where the hypotheses do not already agree.

Similarly, the potential for improvement is less between two hypotheses whose angle is small, even if their distance is large (as in figure 8). In such a case, the distance between them is largely the result of one hypothesis's superiority over the other and the second hypothesis's added value through diversity is lessened.

This concept combines the measure of accuracy with the measure of classifier distance in a way that can be useful in evaluating hypotheses for ensembles. The angle between two hypotheses will be large when the distance between them is large and when the errors of the hypotheses are small. In an ensemble, it may be more desirable to include marginally less accurate individual hypotheses if their diversity is greater, since higher diversity can give the ensemble a greater chance of overcoming the individual hypotheses' weaknesses. The angle measure so given can be thought of as a measurement on how skewed the triangle is. If the angle is small, then either one hypothesis generally dominates over another (a bulk of the patterns one hypothesis gets wrong, the other gets wrong in the same way), or the hypotheses get similar accuracies but do not differentiate themselves much over which patterns they get wrong (and where both miss the same pattern, they tend to give the same wrong answer).

In the case of a classification task that is more than just a concept learning task (i.e. has more than two classifications), this measurement (and the associated angle measurement) can be particularly useful. COD will not only indicate how often two hypotheses or learning algorithms miss the same problems, it will also take into account how often they miss them in the same way, i.e. provide the same incorrect classification. If individual hypotheses in an ensemble tend to provide different incorrect answers, the chance is

greater for the subset of hypotheses in the ensemble which classify a pattern correctly to "outvote" (depending on which flavor ensemble is being employed) the incorrect classifiers.

4. Results and Analysis

The experiments below test the premise that the COD angle measurement can be used to predict improvement potential for combining hypotheses in an ensemble. In this set of experiments, the ensembles are constructed using hypotheses produced by the same algorithm. (Combining hypotheses produced by different algorithms is planned in the near future.) The average angle between hypotheses produced by each of several learning algorithms was measured over the *lymph* data set from the Irvine Machine Learning Repository (Merz & Murphy, 1996). The *lymph* task was chosen for preliminary work because it is not an artificial data set and we believe it is likely to be representative. We plan on repeating these experiments on other data sets to support this premise.

The learning algorithms for which the average COD angle was measured are listed below. They were selected to represent several paradigms within the field of machine learning (decision trees, connectionist, Bayesian, and instance based). More than one algorithm from each paradigm was selected for all but the Bayesian paradigm.

DTg (Decision Tree with gain): A simple decision tree algorithm based on ID3 (Quinlan, 1986).

DTr (Decision Tree with gain ratio): A decision tree algorithm similar to *DTg* except *information gain ratio* is used (as recommended by Quinlan (1993)) instead of information gain.

DTe (Decision Tree with expectation ratio): Another decision tree algorithm similar to the first two with a different split metric. Quinlan notes that information gain tends to favor wide splits while gain ratio favors uneven splits. The expectation ratio modifies the gain ratio metric to favor more even splits.

MLP (Multilayer Perceptron): A feed-forward multilayer perceptron trained with the backpropagation algorithm (Rumelhart et al., 1986), using the logistic activation function and sum-squared error as the objective function and a single hidden layer.

SLP (Single Layer Perceptron): A single layer neural network (Rosenblatt, 1958) using the logistic activation function.

NB (Naïve Bayes): A simple naïve Bayes algorithm as described in Lang (1995).

1NN (One Nearest Neighbor): A standard k-Nearest Neighbor approach (Cover & Hart, 1967) with $k = 1$.

5NN (Five Nearest Neighbor): A standard k-NN algorithm with $k = 5$.

1NNp (One Nearest Neighbor, pruned): A standard k-NN approach with $k = 1$ where as each pattern is added during model construction, the pattern is first tested to see if it would be misclassified. If not, it is not added to the classifier.

5NNp (Five Nearest Neighbor, pruned): The same k-NN approach with $k = 5$, with patterns pruned as in *1NNp*.

To find the average angle between hypotheses produced by a learning algorithm, ten hypotheses were trained for each algorithm above. Different hypotheses were obtained by providing ten different samplings of the training set to the algorithm. These samplings were obtained by splitting the data into ten partitions and using nine partitions to train each hypothesis (as in tenfold cross-validation). All other parameters were kept constant (such as learning rate, initial weight settings, etc.) across all ten experiments.

The angle between each pair of hypotheses produced by a learning algorithm was then computed, using the COD distance between the hypotheses and the accuracy measure for the hypotheses to produce a triangle from which the angle can be calculated. These angle measurements were then averaged to produce an average angle between hypotheses for each algorithm on the *lymph* task.

The experiments discussed above on the *lymph* learning task were used to compute COD angle measurements between several hypotheses generated from the same learning algorithms. Although in this case only the sampling was varied to construct the various hypotheses, other ways of inducing variance in the hypotheses can be used as well when constructing an ensemble classifier (such as presentation order or initial conditions for some algorithms). The COD metric would be used in a similar fashion to estimate how much behavioral variation is produced by using these methods of varying the hypothesis behavior.

Next, a simple voting ensemble was constructed for each of the algorithms discussed above. Ten fold cross-validation was used to measure the results. The ensembles were constructed using resampled training

sets as described for the Bagging approach (Breiman, 1996). Thirty individual models were combined using a simple voting scheme. (The Bagging experiments were also performed with ensembles of ten hypotheses and with ensembles of 100 hypotheses. These experiments behaved similar to the ones shown here using thirty models.)

Table 1 gives the results of the individual and ensemble experiments performed on the *lymph* learning task. The first column lists the learning algorithms employed. The second gives the average error rate for individual hypotheses produced by the learning algorithm. The third gives the error rate for bagged ensembles of the given classifiers. The fourth gives the error reduction of the ensemble solution over the individual model average. (A number less than one indicates that the ensemble's error rate is lower.) The final column gives the average COD angle between hypotheses produced by the algorithm.

Table 1. The results of the individual and ensemble experiments performed on the *lymph* learning task.

Algorithm	Individual Error	Bagging Error	Error Reduction	COD Angle
DTg	0.29	0.25	0.86	42.6
DTr	0.30	0.26	0.87	46.7
DTe	0.26	0.20	0.77	48.4
MLP	0.18	0.17	0.94	32.5
SLP	0.17	0.16	0.94	36.4
NB	0.21	0.23	1.10	15.7
1NN	0.20	0.22	1.10	36.1
5NN	0.19	0.18	0.95	26.4
1NNp	0.27	0.20	0.74	59.8
5NNp	0.26	0.20	0.77	56.1

By comparing the last two columns of the table, a tendency for high COD angle to correlate with error reduction emerges. The first three rows show results for the three decision tree variations. These average angles between individual instances of the same learning algorithm with different sampling (42.6, 46.7, and 48.4) are moderately high, indicating a reasonable diversity in the answers given by the hypotheses. These indications are promising for ensemble potential, and are accompanied by ensemble error reduction (14%, 13%, and 23%).

This average can be compared against the results for naïve Bayes in row six. The naïve Bayes average COD angle value is markedly lower (15.7), suggesting that there would be limited utility in employing an ensemble of naïve Bayes hypotheses, compared to the potential in using decision trees. (This is supported by the slight increase in error, 10% for naïve Bayes.) The *MLP* and *SLP* rows show similar results for individual experiments on *lymph*. These learning algorithms each achieve a lower error rate (0.18 and 0.17 respectively) and have higher average COD angles (32.5 and 36.4) than the naïve Bayes variants, but the angles are not as large as for the decision tree variants. (Note that MLP and SLP error are both reduced by 6% by using an ensemble.)

The nearest neighbor approaches exhibit mixed tendencies. The non-pruning varieties admit similar error rates to naïve Bayes (0.2 and 0.19), with somewhat higher variation (average angles between instances of the same algorithm were 36.1 and 26.4). The pruning varieties give much higher results, both in error rate (0.27 and 0.26) and average angle (59.8 and 56.1), which are comparable to the decision tree results. The possibility for improving the performance of the nearest neighbor algorithms through ensemble techniques appears to be greater for the pruning variants, although the ensemble would possibly be mostly making up for lost ground against the non-pruning variants' lower error rate.

The results in table 1 are also shown in a scatter plot in figure 9. The scatter plot shows a clear tendency for higher COD angle to correlate with improved ensemble performance, which confirms the hypothesis.

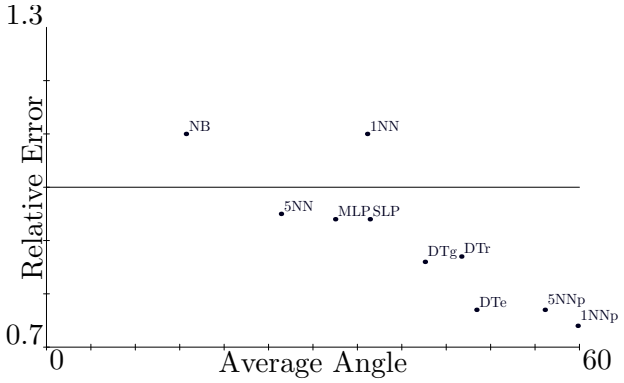

Figure 9. The scatter plot for the results in table 1, graphing error reduction (y axis) against average COD angle (x axis).

5. Conclusion and Future Work

Figure 9 displays a strong correlation between COD angle and error reduction through an ensemble technique. This correlation indicates that the COD angle is predictor of potential improvement through hypothesis combination. A large average COD angle between a set of hypotheses indicates a good potential for accuracy improvement in an ensemble.

We plan on doing similar experiments on other learn-

ing tasks. Additionally, we plan on using the COD angle metric to predict the potential for combining hypotheses produced by different learning algorithms into an ensemble. The use of the COD angle metric will be compared against other approaches to ensemble reduction. Other future work includes pruning ensembles by using the COD metrics to be selective about which hypotheses to combine, and examining methods of combining hypotheses other than ensembles.

References

Ali, K. M., & Pazzani, M. J. (1995). *On the link between error correlation and error reduction in decision tree ensembles* (Technical Report ICS-TR-95-38). University of California, Irvine.

Almuallim, H., & Dietterich, T. G. (1992). On learning more concepts. *Proceedings of the Ninth International Conference on Machine Learning* (pp. 11–19). Morgan Kaufmann.

Bauer, E., & Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, *36*, 105–139.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*, 123–140.

Brown, G., Wyatt, J., Harris, R., & Yao, X. (2005). Diversity creation methods: A survey and categorisation. *Information Fusion*, *6*, 5–20.

Cover, T. M., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, *13*, 21–27.

Dietterich, T. G. (1998). Approximate statistical test for comparing supervised classification learning algorithms. *Neural Computation*, *10*, 1895–1923.

Huang, Y. S., Liu, K., & Suen, C. Y. (1995). The combination of multiple classifiers by a neural network approach. *Journal of Pattern Recognition and Artificial Intelligence*, *9*, 579–597.

Kuncheva, L. I., & Whitaker, C. J. (2001). Ten measures of diversity in classifier ensembles: limits for two classifiers. *Proceedings of the IEEE Workshop on Intelligent Sensor Processing* (pp. 10/1–10/6). Birmingham.

Lang, K. (1995). NewsWeeder: learning to filter netnews. *Proceedings of the 12th International Conference on Machine Learning* (pp. 331–339). Morgan Kaufmann publishers Inc.: San Mateo, California.

Liu, Y., & Yao, X. (1999). Ensemble learning via negative correlation. *Neural Networks*, *12*, 1399–1404.

Margineantu, D. D., & Dietterich, T. G. (1997). Pruning adaptive boosting. *Proc. 14th International Conference on Machine Learning* (pp. 211–218). Morgan Kaufmann.

Merz, C. J., & Murphy, P. M. (1996). UCI repository of machine learning databases. Available at http://www.ics.uci.edu/∼mlearn/MLRepository.html.

Mitchell, T. M. (1980). *The need for biases in learning generalizations* (Technical Report CBM-TR-117). Rutgers Computer Science Department, New Brunswick, New Jersey.

Opitz, D., & Maclin, R. (1999). Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, *11*, 169–198.

Pfahringer, B., Bensusan, H., & Giraud-Carrier, C. (2000). Meta-learning by landmarking various learning algorithms. *Proceedings of the Seventeenth International Conference on Machine Learning, ICML'2000* (pp. 743–750). Morgan Kaufmann, San Francisco, California.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1*, 81–106.

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. Morgan Kaufmann Publishers, Inc.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*, 386–408.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. *Parallel distributed processing: explorations in the microstructure of cognition, vol. 1: foundations*, 318–362.

Vilalta, R., & Drissi, Y. (2002). A perspective view and survey of meta-learning. *Journal of Artificial Intelligence Review*, *18*, 77–95.

Zheng, Z. (1993). *A benchmark for classifier learning* (Technical Report TR474). Basser Department of Computer Science, N.S.W Australia 2006.