SANDIA REPORT

SAND2009-6009
Unlimited Release
Printed September 2009

Toward Improved Branch Prediction through Data Mining

D. Eric Johnson, K. Scott Hemmert

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94-AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone:	(865) 576-8401
Facsimile:	(865) 576-5728
E-Mail:	reports@adonis.osti.gov
Online ordering:	http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone:	(800) 553-6847
Facsimile:	(703) 605-6900
E-Mail:	orders@ntis.fedworld.gov
Online ordering:	http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2009-6009
Unlimited Release
Printed September 2009

Toward Improved Branch Prediction through Data Mining

D. Eric Johnson
The Univeristy of Texas at Austin

K. Scott Hemmert
Scalable Computer Architectures Department
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-1319
kshemme@sandia.gov

Abstract

Data mining and machine learning techniques can be applied to computer system design to aid in optimizing design decisions, improving system runtime performance. Data mining techniques have been investigated in the context of branch prediction. Specifically, a comparison of traditional branch predictor performance has been made to data mining algorithms. Additionally, the possiblity of whether additional features available within the architectural state might serve to further improve branch prediction has been evaluated. Results show that data mining techniques indicate potential for improved branch prediction, especially when register file contents are included as a feature set.

Acknowledgment

This research was conducted by funding provided by Sandia National Laboratories' Lab Directed Research and Development program through the Campus Executive Graduate Research program.

Contents

Figures

Tables

Project Summary

Data mining and machine learning techniques can be applied to computer system design to aid in optimizing design decisions, improving system runtime performance. Data mining techniques have been investigated in the context of branch prediction. Specifically, a comparison of traditional branch predictor performance has been made to data mining algorithms. Additionally, the possiblity of whether additional features available within the architectural state might serve to further improve branch prediction has been evaluated. Results show that data mining techniques indicate potential for improved branch prediction, especially when register file contents are included as a feature set.

For purposes of comparison, data mining branch prediction results are compared to the most accurate known branch predictors to date. The source for these predictors is the Championship Branch Prediction, held in conjunction with MICRO-39. Specifically, the GTL predictor[5], an idealistic predictor not subject to resource constraints that would be necessitated by a hardware implementation, serves as the baseline for this study.

The multilayer perceptron[3], winnow[2], and linear kernel support vector machine[3] algorithms were all investigated. The perceptron was selected because previous work has suggested applicability to branch prediction[1]. The winnow algorithm was chosen because of its suitability at removing uninformative features and at scaling well to high-dimensional spaces. The SVM algorithm was chosen because it too is well suited to high dimensional spaces. However, none of these algorithms were found to differentiate themselves significantly from the state-of-the-art. This is perhaps due to the linear surface of their decision boundaries. This discovery lead to an investigation using decision trees and boosting for branch prediction, where results quickly showed that trees are more relational relative to the features space[3], yielding prediction accuracies significantly better than current state-of-the-art.

The decision tree algorithms LogitBoost, AdaBoostM1, REPTree and ID3Tree were all used to generate classifiers. The data set consists of the most hard to predict benchmarks from the SPEC 2000 integer benchmarks, as defined by the results of Competition Branch Prediction. These classifiers were generated for the top 10 most executed branches where the misprediction rate was greater than 20% for the GTL branch predictor. A 10% contiguous region was used for training each classifier. Each algorithm was trained under two scenarios. First, only global and local branch history bits, up to 62 bits each for a total of 124 feature bits, were considered. Second, both global and local branch history was considered in addition to the register file bits for the x86 general purpose registers `eax`, `ebx`, `ecx`, `edx`, `edi`, `esi`, `ebp`, and `esp`, giving 256 additional feature bits for a total of 380 features. In both scenarios, all possible features were first run through an information gain filter, in order to reduce the possible feature set size down to 32 features. This was found to generally improve overall prediction performance when compared to training using all available features. Given the 32 selected features, finally each classifier was trained and then evaluated. Results for classification using a validation set is shown in Figure 1. The *x*-axis distinguishes each of the 10 individual branches, with the last entry indicating the weighted average for all 10 branches. The *y*-axis shows the branch misprediction rate, where lower is better.

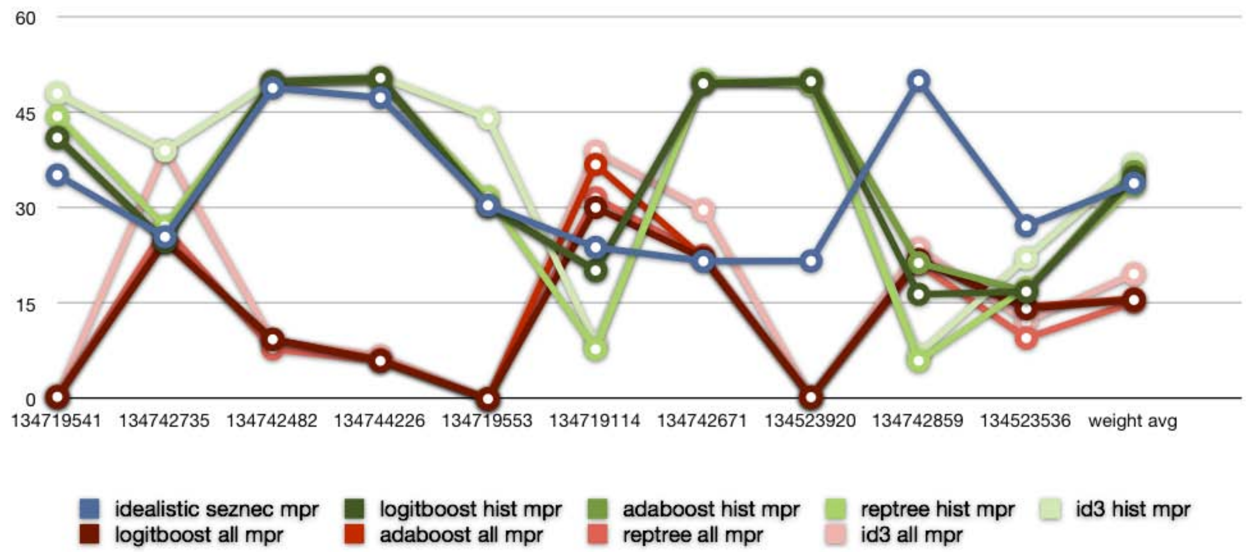

Figure 1. Tree results

The performance of the GTL predictor, the top performing from Championship Branch Prediction, is shown in Figure 1 as the blue line. This result serves as a reference point for our analysis. The performance for the LogitBoost, AdaBoostM1, REPTree and ID3Tree algorithms, given global and local branch history as input features, are shown in Figure 1 as lines in shades of green. On average, these classifiers performed as well as the GTL predictor. It is interesting to note, however, that in some cases these classifiers performed significantly better than the GTL predictor. This suggests that a further analysis into the potential for creating a more accurate branch predictor utilizing only branch history is possible.

Finally, the performance for the LogitBoost, AdaBoostM1, REPTree and ID3Tree algorithms, given global and local branch history in addition to the register file bits as input features are shown in Figure 1 as lines in shades of red. On average, these classifiers performed about twice as well as the GTL and history-only predictors. For one branch, the prediction rate dropped all the way to 0% for every register based classifier. This illustrates that significant potential exists for incorporating the contents of the register file into branch prediction.

With the goal of a hardware-implementation of a branch predictor, the specific results for each algorithm were investigated in greater detail. Of particular interest was the REPTree algorithm. Upon examination of the REPTree (reduced error pruning tree) results, it was found that the tree size could actually be very small while dramatically improving prediction accuracy in comparison to the GTL predictor. One example of a small tree produced by REPTree is shown in Figure 2. This tree models the branch behavior at address 134742482. This branch has a misprediction rate of 48.82% with the GTL predictor. In contrast, the simple REPTree had a 19.58% misprediction rate, which is over 50% more accurate, while only using 3 register bits to make its decision. This suggests significant potential in realizing increased branch prediction accuracy through data mining

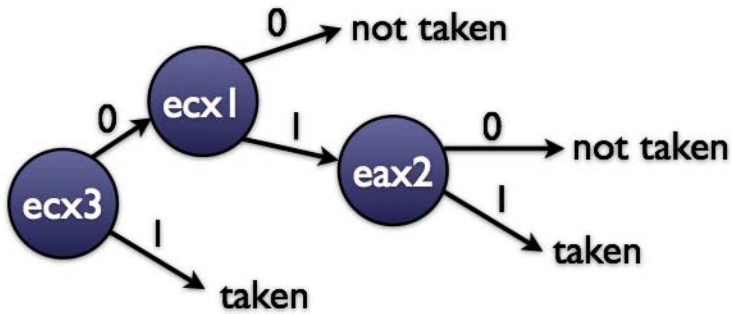

Figure 2. REPTree

techniques.

It was found that decision trees and boosting algorithms can outperform state-of-the-art predictors like the perceptron and GTL. Trees use a relational representation to form a possibly disjunctive boundary which enables it to correctly navigate through a noisy space. Additionally, information gain serves as a successful ranking tool to identify potentially helpful input features to decision trees and boosting algorithms. Finally, it was observed that decision trees and boosting algorithms are capable of creating small and simple trees that can dramatically improve branch prediction accuracy.

References

[1] Daniel A. Jimenez and Calvin Lin. Dynamic branch prediction with perceptrons. In *Proceedings of the Seventh IEEE International Symposium on High Performance Computer Architecture*, 2001.

[2] Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4), April 1988.

[3] Tom M. Mitchel. *Machine Learning*. The McGraw-Hill Co. Inc., 1997.

[4] Andre Seznec. Analysis of the o-gehl branch predictor. In *Proceeding of the 32nd Annual International Symposium on Computer Architecture*, June 2005.

[5] Andre Seznec. Looking for limits in branch prediction with the gtl predictor. In *Competition Branch Prediction 2*, December 2006.

DISTRIBUTION:

1	MS 1319	K. Scott Hemmert, 01422
1	MS 1319	James A. Ang, 01422
1	MS 0899	Technical Library, 9536 (electronic)
1	MS 0123	D. Chavez, LDRD Office, 1011

Sandia National Laboratories