Introduction to Machine Learning: Autumn Semester 2015
Instructor: Prof. Joachim M. Buhmann

Course Project 1

Linear Regression to Predict Processor Performance

1 Introduction

You have seen in class that simple linear models can be very powerful in predicting complex functions. In this task, you are asked to build a model that predicts the delay in microseconds that a processor requires to execute a fixed portion of a program given a set of microarchitectural configurations.

The performance of a processor can greatly vary when configurations such as cache size and register file size are varied. The extent of this variation depends on how the program exploits these characteristics. Fourteen different microarchitectural parameters can be varied across a range of values, and the delay of the processor can be measured under such conditions.

The relationship between microarchitectural characteristics and processor delay might be non-linear, and the given observations might be noisy. However, you should build a linear regressor using the techniques learned in class to find a compromise between complexity and accuracy in predicting the given training set, to avoid overfitting in the presence of noise and in the lack of abundant training data.

To be able to model the non-linear relationship in the given data set, try computing new features from the given ones and adding them to the feature space. The hope is that in this extended feature space, a linear relationship between the inputs and the output variable can be found. Note that the more complex your feature space becomes, the more prone you are to overfitting to noise since you are only given a finite data set to train your regressor.

The project is hosted on Kaggle and you get invited by opening the following link:
"https://kaggle.com/join/y9Br5msvRf5Zpw3nNmBK3dkV".

NOTE: YOU NEED TO REGISTER WITH YOUR OFFICIAL ETH E-MAIL ADDRESS! UNOFFICIAL REGISTERED STUDENTS WILL BE DISQUALIFIED!

1.1 Kaggle and forming teams

In order to map your final score from the Kaggle competition back to the ETHZ grading system of the course, you must use your @ethz.ch email address for the Kaggle user account used during the competition.

It is allowed to solve this project in teams with up to three team members. Please form the teams from inside of the Kaggle project page yourself. For this, someone from your team:

- Opens the Kaggle project page
- Select "My Team" from navigation on the left hand side
- Enters the "Team Name" in the field on the top
- Under "Invite someone to join your team" puts in the emails from the other team mates and then clicks the "Send" button

If you are looking for team mates and cannot find someone during the lecture, feel free to use the Kaggle forum of the project to look for other team members. You find the forum by clicking on "Forum" from the Kaggle project page.

Note: While it is possible to also form teams later during the project, we recommend forming the teams right away to avoid problems with the total amount of made submission when merging a team. For more information see: "https://www.kaggle.com/wiki/FormingATeam".

2 Data set description

2.1 Input

We are going to consider 14 input features for this problem. Each feature is a microachitectural configuration and is an integer variable. The names and possible values of each feature are described in Table 1.

Name	Range and possible values
Width	2,4,6,8
ROB size	32 to 160
IQ size	8 to 80
LSQ size	8 to 80
RF sizes	40 to 160
RF read ports	2 to 16
RF write ports	1 to 8
Gshare size	1K to 32K
BTB size	256 to 1024
Branches allowed	8,16,24,32
L1 Icache size	64 to 1024
L1 Dcache size	64 to 1024
L2 Ucache size	512 to 8K
Depth	9 to 36

Table 1: Microarchitectural parameters and their possible values

Note: The meaning of each feature is not important for performing well at this task. You can obtain the perfect score without knowing anything about microprocessors.

2.2 Output

You are asked to build a model that predicts the delay in microseconds that a processor requires to execute a fixed portion of a program given a set of microarchitectural configurations. The delay is a positive real number.

2.3 Training Set

This data is formatted as a comma-separated values (CSV) file in which each line corresponds to an observation. Each observation consists of 16 values: an unique id of the data set, 14 microarchitectural parameters (in the same order as they have been introduced above) followed by the delay. Each line has the following format: id, width, ROB size, IQ size, LSQ size, RF sizes, RF read ports, RF write ports, Gshare size, BTB size, branches allowed, L1 Icache size, L1 Dcache size, L2 Ucache size, depth, delay. The training set is in the file "training.csv".

2.4 Validation and Test Sets

Both validation and test set contain configurations that have not been measured yet. Your task is to predict the delay for a given configuration. You will be given several configurations that specify the 14 microarchitectural parameters and you are asked to predict the delay for each configuration. The data sets are given in the file "validate_and_test.csv". (Often the validation and test set are provided separately. As Kaggle has only a single submission, the validation and test set are combined here.)

The formatting of the prediction file is as follows:

- Same line format as the training set except that the delay is not given (each line has only the 15 comma-separated configuration features).
- **Required output:** a file that contains the predictions. Make sure to use the same ids as in the "validate_and_test.csv" file and column names as show in the "example_solution_handin.csv" file. Kaggle won't be able to score your submission if you forget the column headers!

After making the submission Kaggle will compare your solution to the ground truth and compute the error. From the submitted data you will get only feedback on half of your predictions. This feedback corresponds to the validation data set and will be listed in Kaggle's "public leaderboard". The final score and grading will be made from the other half of your prediction in a "private leaderboard" (this corresponds to the test data set). During the competiton, you will see only the results to the "public leaderboard".

To generate the submission file the following code snippet might be helpful. It shows you how to write your prediction with the necessary column headers to a file:

```
% IMPORTANT: Make sure to have the delays as column vector.
delays = [100;30]; %<<< Compute your prediction here
ids = [1;2];       %<<< Use the ids from the prediction file

% Generate a single array with the Ids and delays.
res = horzcat(ids, delays);

% Convert the array to a table and add column headers.
table = array2table(res, 'VariableNames', {'Id', 'Delay'});

% Write the obtained table to a CSV file.
writetable(table, 'your_prediction.csv');
```

3 Evaluation and Grading

Each submission (upload of a prediction file for a given data set) will be ranked according to the Root Mean Squared Error (RMSE) of the predictions. We will call this function of the error the "cost" of the predictor, because a better predictor will have a lower RMSE. Since we have the measured delay for each configuration in the validation set and test set we can calculate response.

We compare the cost of the submission to a baseline prediction which is 850. Hitting this baseline corresponds to a grade of 4. The required score to get a 6 will be determined after the competition has finished.

3.1 Report handin

In addition to your predictions on the test set you need to provide a brief report that explains how you obtained your results. One report per team is enough.

We include a template for LATEX in the file "report.tex". If you do not want to use LATEX, please use the same sections as shown in "report.pdf". Upload a zip file with the report (as a PDF file) along with your code or parameters/screenshots of the tools you used. For further instructions refer to the report template.

We might ask you to show us what you did, so please keep the necessary files until the end of the semester.

3.2 Deadline

You will be able to submit predictions starting from **Friday, 2.10.2015** until **Friday, 31.10.2015, 23:59:59 UTC**.

4 Questions

If you have questions regarding the project, please ask during the tutorial next week. Afterwards contact Julian Viereck (jviereck@student.ethz.ch) via email and use "[ML]" in your email subject. For example "[ML] Question about ...".