6.830 Project Proposal
Machine Learning Algorithms for In-Database Analytics

Franck Dernoncourt, Rebecca Taft and Sumaiya Nazeen

Overview

Our project will focus on extending the functionality of MADlib. MADlib is an open source machine learning and statistics library which works with Postgres or Greenplum to provide in-database analytics. Although some machine learning algorithms have been implemented in MADlib, there is room for additional contributions. We plan to implement three different machine learning algorithms for MADlib, and will attempt to contribute any resulting code to the MADlib community codebase. We will also assess the performance of our implementations on different datasets and compare their performance with the same algorithms in different platforms.

Approach

Algorithm Implementation:

We have chosen the three algorithms below since they represent a wide range of types of machine learning algorithms: one clustering algorithm, one classification algorithm, and one prediction algorithm.

- Biclustering: Biclustering is an algorithm commonly used in genomics to find similar genes and similar conditions in gene expression data. The algorithm simultaneously clusters both the rows and columns of a matrix. We will write a user-defined function in MADlib modeled after the R implementation of biclustering. The implementation we will focus on uses the sparse singular value decomposition algorithm as described by M. Lee, et al. (Biometrics, 2010). Documentation of the R implementation is available at http://www.inside-r.org/packages/cran/s4vd/docs/ssvd. It may be difficult to translate the R implementation into C++, so if we find a C++ machine learning library which we can call that would be ideal.
- Model-Based Boosting: Model based boosting refers to a class of gradient descent classification algorithms for optimizing general loss functions that uses component-wise least squares, either of parametric linear form or smoothing splines, or regression trees as base learners for fitting generalized linear, additive and interaction models to potentially high-dimensional data. R has a package called mboost which implements generalized linear and generalized additive models utilizing flexible boosting algorithms for (constrained) minimization of the corresponding empirical risk function. We will implement the mboost equivalents in C++ for MADlib. We can test these algorithms on gene expression data.

- Genetic Programming: We plan to use genetic programming for performing non-linear symbolic regression. The term "symbolic regression" represents the process during which measured data are fitted by suitable mathematical formulas like $x^2 + c$ or $\sin(x) + {}^{1}/_{1+e^{x}}$ etc. Symbolic regression can be based on evolutionary algorithms and its main aim is to "synthesize" a program (mathematical formulas, computer programs, logical expressions, etc.) in an evolutionary way which will solve user-defined problems. While the domain of evolutionary algorithms is numerical in nature (real, complex, integer, discrete), the domain of symbolic regression is functional in nature, i.e. it consists of a set of functions and a set of terminals. The final program is synthesized from a mixture of both sets, and can be quite complicated from a structural point of view. We plan to use it to unravel unknown relations between attributes.

Performance Analysis

- We will test our implementations with different sized datasets and see how our implementations scale if the data cannot fit in main memory.
- We will compare our implementation of each algorithm to existing implementations in R, as well as other platforms.
- We will compare the performance of our algorithms with serial execution and parallel execution if we get access to the Greenplum platform.

Resources

We plan to use the following datasets for testing:

- NCBI GEO Microarray data
- MIMIC data (blood pressure waveform)
- Datasets from UCI Machine Learning Repository (Website: http://archive.ics.uci.edu/ml/)

Milestones

- Install PostgreSQL and MADlib on a server
- Create a dummy MADlib function to see if we can use it on our database
- Implement our three algorithms in C++ in the MADlib framework
- Compare performance of algorithms on different datasets and against implementations in R (client side) and other platforms
- If time allows, compare serial vs. parallel execution of our implementations
- Contribute any resulting code to the MADlib community codebase
- Write report and prepare presentation

References

[1] Mihee Lee, Haipeng Shen, Jianhua Z. Huang and J. S. Marron1 "Biclustering via Sparse Singular Value Decomposition", Biometrics, 2010

[2] Hellerstein, Joseph M., et al. "The MADlib analytics library: or MAD skills, the SQL." *Proceedings of the VLDB Endowment* 5.12 (2012): 1700-1711.
http://vldb.org/pvldb/vol5/p1700_joehellerstein_vldb2012.pdf

[3] Hothorn, Torsten, and Peter Bühlmann. "Model-based boosting in high dimensions." *Bioinformatics* 22.22 (2006): 2828-2829.
http://bioinformatics.oxfordjournals.org/content/22/22/2828.full

[4] Torsten Hothorn, Peter Buhlmann, Thomas Kneib, Mattthias Schmid and Benjamin Hofner (2010), Model-based Boosting 2.0. Journal of Machine Learning Research, 11, 2109-2113
http://cran.r-project.org/web/packages/mboost/vignettes/mboost.pdf

[5] Contributing to MADlib https://github.com/madlib/madlib/wiki/Contribution-Guide