CSE 232A
Graduate Database Systems

Arun Kumar

About Paper Reviews
Goal of Peer Review in Research

❖ Gatekeeping for quality of publication venue
❖ Collation of technical knowledge in the field
❖ Identify/support emerging research problems/areas
❖ Recognize/reward technical novelty, creativity, depth
❖ Provide constructively critical feedback to authors
❖ Appreciate strong efforts of authors
Goal of Paper Reviews in 232A

❖ Teach how to read cutting-edge research papers with a “critical thinking” mindset
❖ Teach how to appreciate/evaluate emerging ideas in an objective, honest, and balanced manner
❖ Make you take the paper readings seriously! :) 
❖ Perhaps try to identify research gaps and extensions?
Towards a Unified Architecture for in-RDBMS Analytics

Xixuan Feng  Arun Kumar  Benjamin Recht  Christopher Ré
Department of Computer Sciences
University of Wisconsin-Madison
{xfeng, arun, brecht, chrisre}@cs.wisc.edu

ABSTRACT
The increasing use of statistical data analysis in enterprise applications has created an arms race among database vendors to offer ever more sophisticated in-database analytics. One challenge in this race is that each new statistical technique must be implemented from scratch in the RDBMS, late 1990s and early 2000s, this brought a wave of data mining toolkits into the RDBMS. Several major vendors are again making an effort toward sophisticated in-database analytics with both open source efforts, e.g., the MADlib platform [17], and several projects at major database vendors. In our conversations with engineers from Oracle [38] and PMC Gauss DB [31], we learned that a key bottleneck is

ACM SIGMOD 2012
Project Bismarck
(Topic: DB for ML)
(Setting) Integration of ML procedures with RDBMSs becoming popular for large-scale analytics over RDBMS-resident data without moving/copying data

(Problem) Redesigning every individual ML procedure for in-RDBMS execution from scratch is a long, tedious, and wasteful development process

(Approach) This paper proposes a unified abstraction and software architecture for a large class of ML procedures based on incremental gradient descent (IGD) that is implementable using existing common RDBMS abstraction of user-defined aggregate (UDA)
Each RDBMS has its own tools for ML problems. Usually they have different tools for different ML algorithms, which makes them difficult to maintain and cause tons of development overhead.

Since most of the ML techniques can be represented as algorithms solving convex programming problem, i.e. minimizing some convex cost function, it is possible to use one single architecture to unify all of them.

The authors proposed a unified architecture based on IGD and UDA allowing developer to adapt it to different ML problems with little development overhead.

The authors also proposed a modified reservoir sampling technique called MRS. What is more, the authors studied the influence of data ordering and parallelized BISMARCK.
Sample good strong points from S1

❖ Generality. A single framework solves multiple problems, making maintaining and development easier. The reuse of codes are drastically improved. One optimization for BISMARCK means optimizations for ALL.

❖ UDA-based. It is very easy to re-implement for different RDBMSes.

❖ Efficiency. It is only a little slower than an common aggregation and it out-performs many of the built-in tools provided by RDBMSes.
Generality means loss of speciality. Using IGD for all convex problems may cause a consequence that some of the ML techniques can be more efficiently solved by some other specific techniques. This is the tradeoff.

The limitations of IGD are also limitations of BISMARCK: internally sequential, hard to tune. Problems cannot be solved by IGD cannot use BISMARCK.

RDBMSes have more to offer. BISMARCK only utilizes the UDA of databases. There is still space for optimization. Especially for distributed databases.
Other good strong points from class

- New optimization strategies can be tested using Bismarck without having to make changes for all analytic techniques.

- The paper honestly studies its overhead as well as thoroughly compares the result of integrations in three different RDBMS.

- The experiments are compelling. Their use of wall-clock-time measurements, and benchmarks against native UDA speeds, presents a strong case.

- The organization of the paper is very helpful such that reader who has little knowledge in this area can read and grasp the main concepts. The authors take real-world examples where necessary to explain the concepts which is helpful.
The theoretical justification of why IGD is essentially commutative and algebraic lacked depth. The claim that averaging models trained on different segments of the data would lead to convergence seemed dubious.

Another limitation is that this architecture is designed for single node RDBMS. Currently, more applications move to cloud services and use distributed database or frameworks such as Hadoop and Spark.

The assumption that the data is static might not be upheld in a production environment and Bismarck has no provision to support online learning.

Strong assumption that the state (model parameters) fit in RAM.