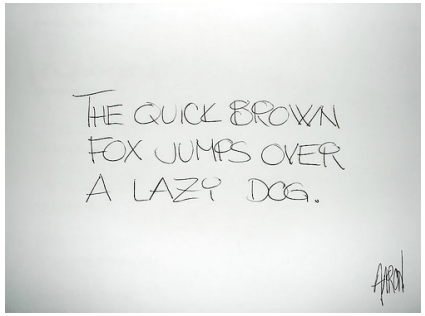
Christopher Ré

Joint work with the Hazy Team

http://www.cs.wisc.edu/hazy

Two Trends that Drive Hazy

1. Data in unprecedented number of formats

2. Arms race for deeper understanding of data

Automated → Statistical **AND** Manage Data → RDBMS

Hazy integrates statistical techniques into an RDBMS

Hazy Hypothesis: Handful of statistical operators capture a diverse set of applications.

An RDBMS in One-Slide

A **relational database management system** (RDBMS) is a software artifact that *simplifies building applications that use large amounts of data* by providing:

1. data storage,
2. sophisticated query processing infrastructure, and
3. a programming model that simplifies concurrent access to data (transactions).

Model: data stored as set of relations (sets of tuples) and transformed via first-order logic statements (SQL).

Hazy: Extend an RBDMS to handle the requirements of applications that *use statistical data analysis*.

Outline

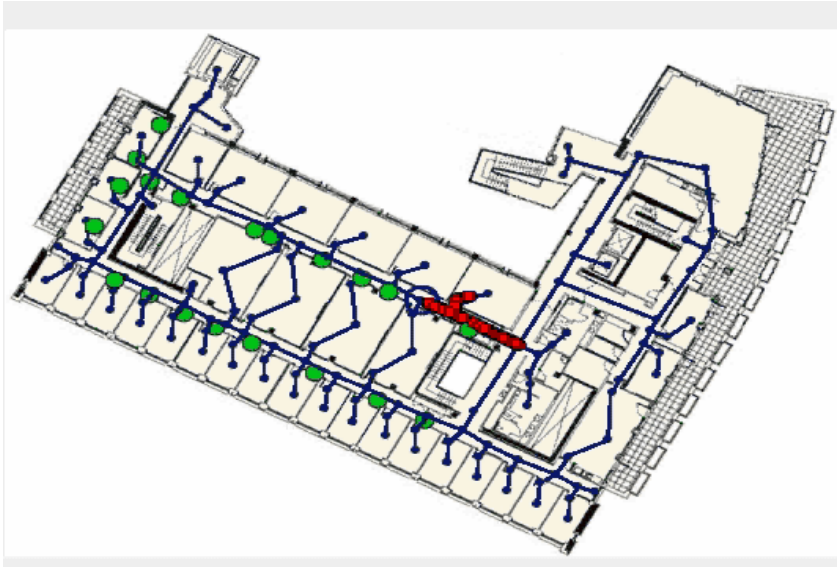
- Three Application Areas for Hazy
- Drill Down: One Text Application
- Maintaining the Output of Classification
- Hazy Heads to the South Pole

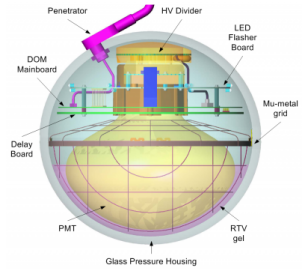
Extract and **Classify** sentiment about products, ad campaigns, and customer facing entities.

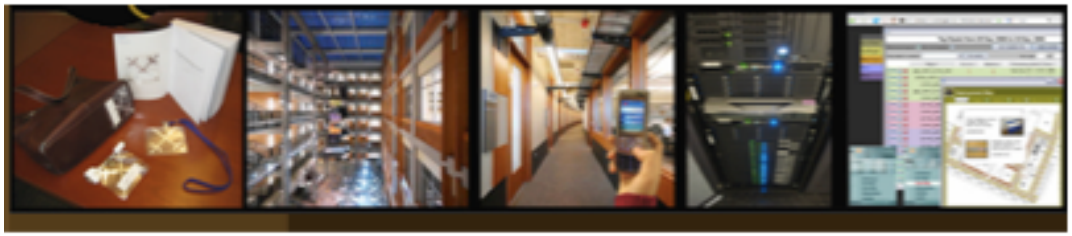
Data constantly generated on the Web, Twitter, Blogs, and Facebook

Build tools to lower cost of analysis

Statistical tools for **extraction** (e.g., CRFs) and **classification** (e.g., SVM). Performance and maintenance are data management challenges (DMC)







A physicist **interpolates** sensor readings and uses **regression** to more deeply understand their data

DMC: Transform and maintain large volumes of sensor data and derived analysis

Models that extract entities from sequences of words
are similar to
models that extract physical meaning from sensor readings.

HATHI TRUST Digital Library

Google books

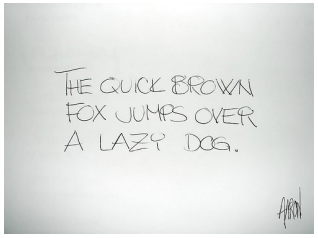
Browse popular books

A social scientist wants to **extract** the frequency of **synonyms** of English words in 18th century texts.

Getting text is challenging!
(statistical model errors)

Output of speech and OCR models similar to output of text labeling models

OCR & Speech



DMC: Process large volumes of statistical data

Takeaway and Implications

Statistical processing on large data
enables a wide variety of new applications.

Hazy Hypothesis: Handful of statistical operators
capture a diverse set of applications

Key challenges are maintenance and performance
(data management challenges)

Outline

Three Application Areas for Hazy

Drill Down: One Text Application

Maintaining the Output of Classification

Hazy Heads to the South Pole

DBLife jeffrey naughton Search Help The Cimple Project

Jeffrey F. Naughton

Bing Citeseer DBLP Google Google Scholar Kosmix Wikipedia Yahoo!

from Google Images

Professor
http://pages.cs.wisc.edu/~naughton/
University of Wisconsin-Madison
USA
Papers cited 11,398 times
H-Index of 55

Related People

- Raghu Ramakrishnan
- David J. DeWitt
- Raghav Kaushik

Recent News

Mixed Mode XML Query Processing

Classify publications by subject area

2009

146 The Case for a Structured Approach to Managing Unstructured Data. AnHai Doan, Jeffrey F. Naughton, Akanksha Baid, Xiaoyong Chai, Fei Chen, Ting Chen, Eric Chu, Pedro DeRose, Byron J. Gao, Chaitanya Gokhale, Jiansheng Huang, Warren Shen, Ba-Quy Vuong. CIDR 2009. [Information Extraction] Cited by 2 Web Search BibTeX Download

On the Computation of Multidimensional Aggregates cited 1 time - details

News Archive

Sorted by Year/Conf, Year/Citation, Citation

Community Statistics

2009

146 The Case for a Structured Approach to Managing Unstructured Data. AnHai Doan, Jeffrey F. Naughton, Akanksha Baid, Xiaoyong Chai, Fei Chen, Ting Chen, Eric Chu, Pedro DeRose, Byron J. Gao, Chaitanya Gokhale, Jiansheng Huang, Warren Shen, Ba-Quy Vuong. CIDR 2009. [Information Extraction] Cited by 2 Web Search BibTeX Download

145 Anonymization of Set-Valued Data via Top-Down, Local Generalization. Yeye He, Jeffrey F. Naughton. PVLDB (2): 934-945 (2009). [Data Privacy] Web Search BibTeX Download

144 Combining keyword search and forms for ad hoc querying of databases. Eric Chu, Akanksha Baid, Xiaoyong Chai, AnHai Doan, Jeffrey F. Naughton. SIGMOD Conference 2009, 349-360. [Database Search] Cited by 1 Web Search BibTeX Download

143 Efficiently incorporating user feedback into information extraction and integration programs. Xiaoyong Chai, Ba-Quy Vuong, AnHai Doan, Jeffrey F. Naughton. SIGMOD Conference 2009, 87-100. [Information Extraction] Web Search BibTeX Download

- SIGMOD 2010 (Committee Members) [1]
- SIGMOD 2010 (SIGMOD Net Initiatives Committee) [2]
- CIDR 2009 (PC) [3]
- SIGMOD 2009 (Committee Members) [4]

more

Related Organizations

- University of Wisconsin-Madison
- Microsoft Research
- Microsoft
- IBM Almaden Research Center

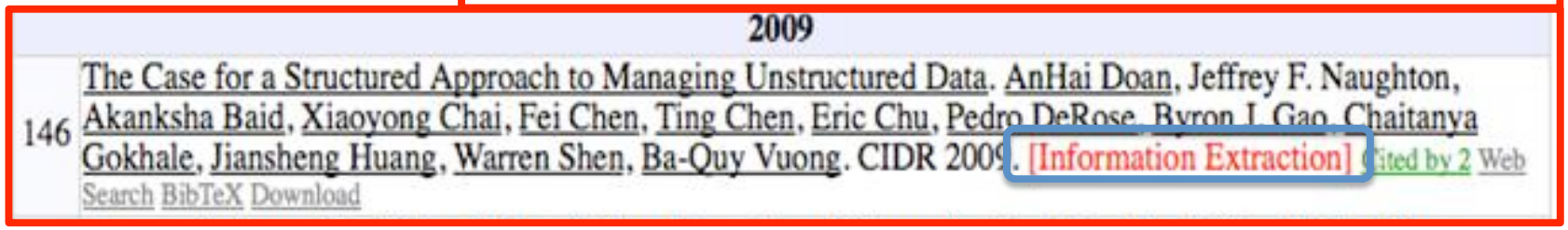
more

Panels

- VLDB 2002 [5]

The workflow requires several steps

Classify publication by subject area



Simplified workflow

1. Paper references are crawled from the Web.
2. Entities (Papers, Authors,…) are ***extracted*** and ***deduplicated***.
3. Each paper is ***classified*** by subject area
4. DB is queried to render Web page.

We still use the RDBMS for rendering, reports, etc.

Hazy Evidence: We know names for these operators

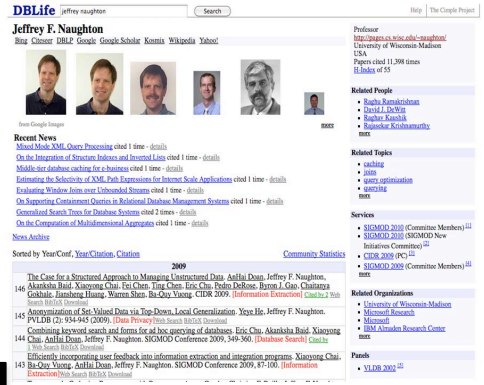
An Example of How Hazy Helps

Learning and Inference Declaratively Specified Together

Tuples In. Tuples out. *Hazy handles the statistical and traditional details.*

CREATE CLASSIFICATION VIEW V(id,label)
ENTITIES FROM Papers
EXAMPLES FROM Example

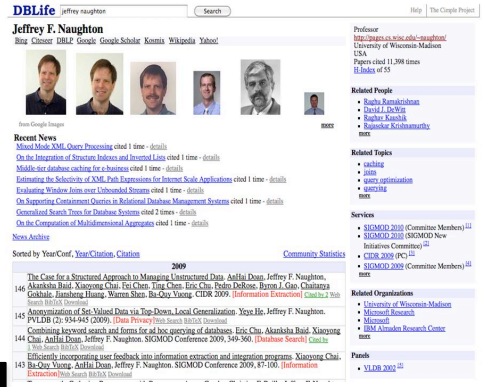
Declarative SQL-Like Program



Hazy/ RDBMS

Hazy Helps with Corrections

Paper 10 is not about *query optimization* -- it is about *Information Extraction*



CREATE CLASSIFICATION VIEW V(id,label)
ENTITIES FROM Papers
EXAMPLES FROM Example

Declarative SQL-Like Program

Hazy/ RDBMS

Easy as an INSERT: Update fixes that entry – and perhaps more – *automatically*.

Design Goals: Hazy should…

- … look like standard language (SQL)
  - Ideal: application unaware of statistical techniques
  - Build on solutions for classical data management problems
- … automate routine tasks
  - E.g., updates propagate through the system
  - Eventually, order operators for performance

Where Hazy is Now

Building Like Mad (Cows)

User declares task to Hazy using SQL (First Order Logic)

- In PostgreSQL, we've built:
  - **Classification:** SVMs, Least Squares
  - **Cluster/Equivalence**: Synonyms and Coreference
  - **Factor Analysis**: Low-Rank Matrix Factorization
  - **Transducers for Sequences:** Text, Audio, & OCR
  - **Sophisticated Reasoning:** Markov Logic Networks

Model-based Views (Deshpande et al)

Beat them at their own game: Using Hazy, we rebuilt prior systems with higher quality and performance!

Reasoning by Analogy…

Hazy Hypothesis: Handful of statistical operators capture a diverse set of applications.

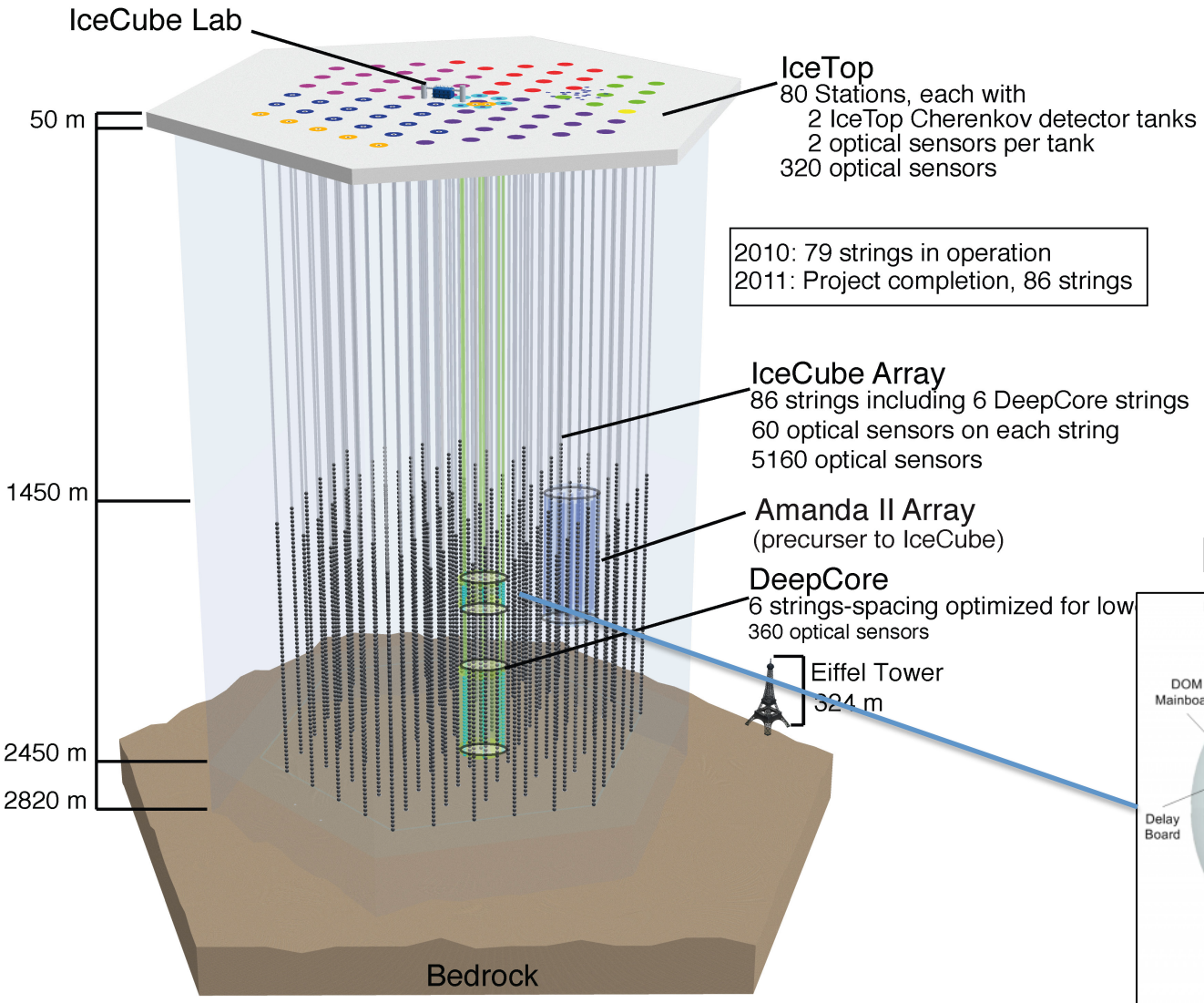
Classical RA	Hazy Operator
Selection	Classification
Projection	Clustering (Equivalence)
Join	Factor Analysis
SQL's LIKE	Transducer algebra
Constraints	Markov Logic Networks

Left hand-side (+ set difference) = First Order Logic

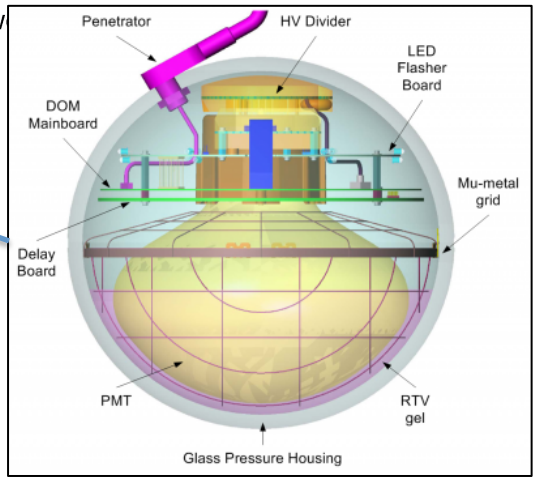
Hazy Heads to the South Pole

IceCube

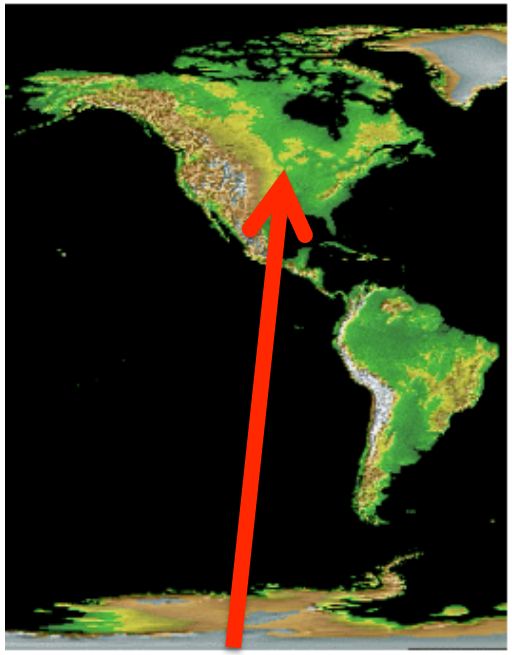
IceCube



Digital Optical Module (DOM)

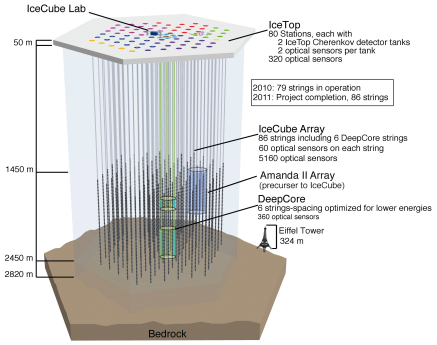


Workflow of IceCube



In Madison: Lots of data analysis.

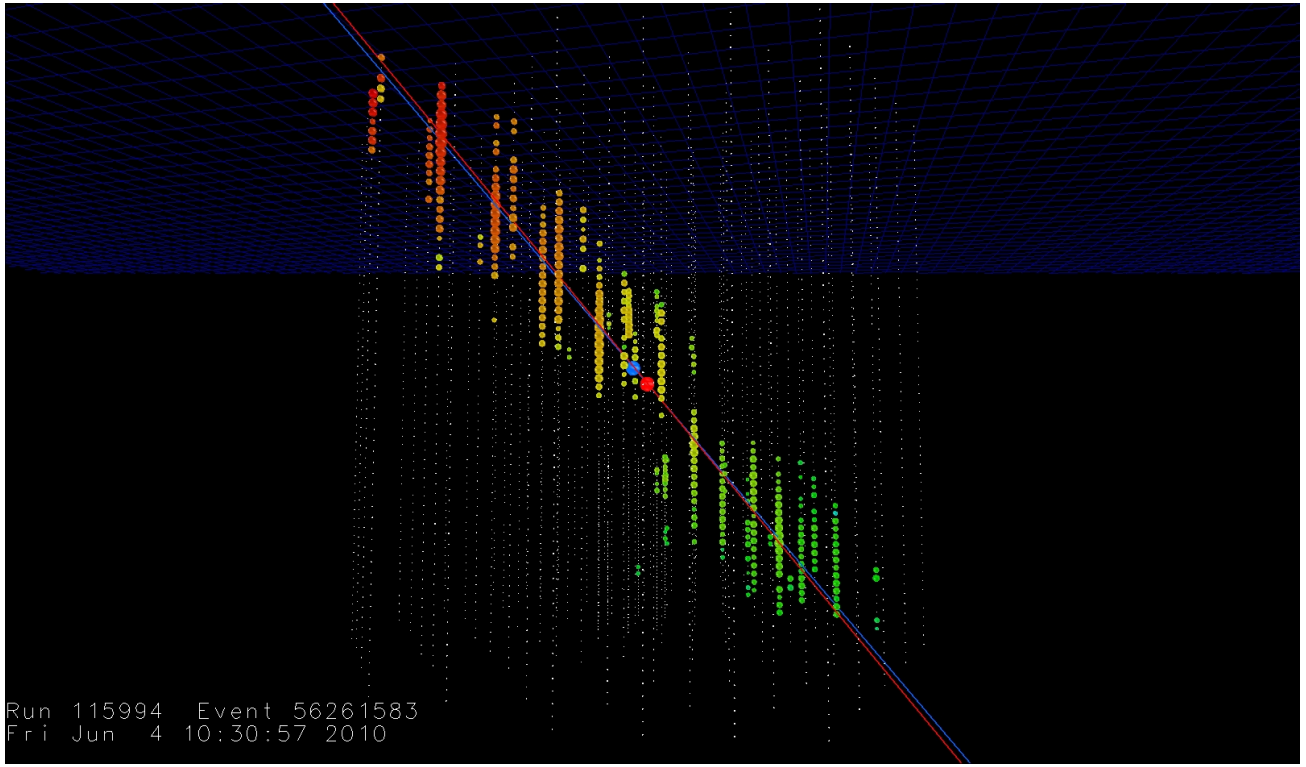
Via satellite: Interesting DOM readings



At Pole: Algorithm says “Interesting!”

In Ice: Detection occurs.

A Key Step: Detecting Track



Here, Speed ≈ Quality

Mathematical structure used to help track neutrinos is similar to labeling text/tracking/OCR!

Framework: Regression Problems

$$\min_x P(x) + \sum_{i=1}^{N} f(x, y_i)$$

x	the model
y_i	A data item
f	Scores the error
P	Enforces prior

Examples:

1. **Neutrino Tracking**: y_i is a DOM (sensor) reading
2. **CRFs**: y_i is (docment, labeling)
3. **Netflix**: y_i is (user,movie,rating)

Others tools also fit this model, e.g., SVMs

Claim: General data analysis technique that is amenable to RDBMS processing

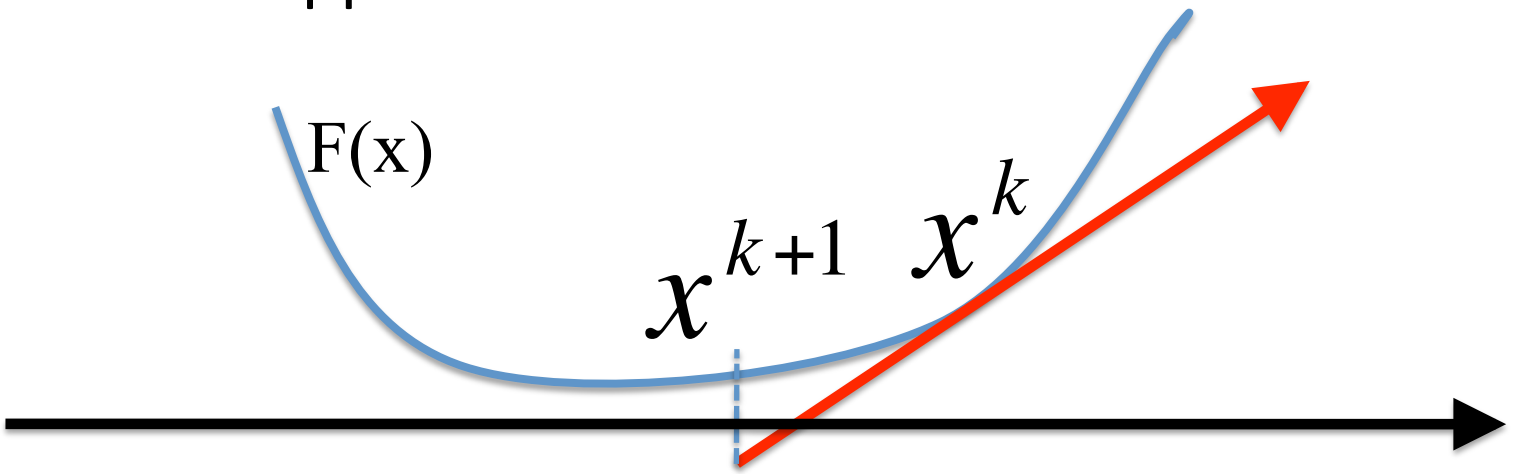
Background: Gradient Methods

$$F(x) = P(x) + \sum_{i=1}^{N} f(x, y_i)$$

Gradient Methods: Iterative.

$$x^{k+1} = x^k - \nabla F(x^k)$$

1. Take current x,
2. Derivate F wrt x,
3. Move in opposite direction



Incremental Gradient Methods

$$F(x) = P(x) + \sum_{i=1}^{N} f(x, y_i)$$

Gradient Methods: Iterative. $x^{k+1} = x^k - \nabla F(x^k)$

1. Take current x,
2. *Approximate* derivative of F wrt x,
3. Move in opposite direction

$$\nabla F(x) \approx \nabla P(x) + N \nabla f(x, y_j)$$

Sample a single data item to approximate the gradient

Incremental Gradient Methods (iGMs)

Why use iGMs? Provably, iGMs converge to an optimal for many problems, but the real reason is:

iGMs are *fast.*

Technical connection: iGM processing isomorphic to processing a tuple, so RBDMS processing techniques apply

RDBMS can choose high performance data execution plans using cost models of disk, memory, cpu, etc.

RDBMS abilities are not fully utilized

We may be able to access data orders of magnitude faster at the expense of some bias in the iGM steps. What is the trade off?

How does bias affect convergence of iGMs?

Noise-free Least Squares

$$\min_{x \in \mathbb{R}^d} \sum_{i=1}^{m} (a_i^T x - b_i)^2 = \min_{x \in \mathbb{R}^d} \|Ax - b\|_2^2$$

Simplification: noise-free (exists some x^* s.t. $Ax^*=b$)

$$x^{(m)} = x_* + \prod_{i=1}^{m} (I - \alpha a_{\eta(i)} a_{\eta(i)}^T)(x^{(0)} - x_*)$$

Constant stepsize $\alpha/2$

How we select $\eta(i)$ matters:

1. With replacement, converges at 1/k rate. (fastest in theory)
2. Worst case, deterministic ordering (quadratically slower)
3. Without replacement, empirically fastest (no better than 2)

Gradients Map to Matrix Norms

Update rules explicitly after m steps (epoch)

$$\mathbb{E}_{\mathrm{wr}}(x^{(m)} - x_*) = \left(I - \frac{\alpha}{m} \sum_{i=1}^{m} a_i a_i^T \right)^m (x^{(0)} - x_*)$$

For some ordering η

$$x^{(m)} - x_* = \prod_{i=1}^{m} \left(I - \alpha a_{\eta(i)} a_{\eta(i)}^T \right) (x^{(0)} - x_*)$$

$$\left(\frac{1}{N} \sum_{i=1}^{N} c_i \leq \right)^N \geq \prod_{i=1}^{N} c_i$$

AGMI for scalars

From scalar AGMI one may hope

For any η geometric mean

smaller 2-norm arithmetic mean

AGMI fails for two or more matrices

The Arithmetic-Geometric Mean Inequality does not hold for matrices. (fails for 3 matrices)! BUT,

THEOREM *Let $X_1, \ldots, X_m$ be $d \times d$ positive definite matrices. Then*

$$\left\| \prod_{i=1}^{m} X_i \right\| \leq \left\| \frac{d}{m} \sum_{i=1}^{m} X_i \right\|^m . \tag{7}$$

In 2d, worst ordering as k=1,…,n is given by.

$$X_k = \begin{bmatrix} \cos^2\left(\frac{\pi k}{n}\right) & \cos\left(\frac{\pi k}{n}\right)\sin\left(\frac{\pi k}{n}\right) \\ \cos\left(\frac{\pi k}{n}\right)\sin\left(\frac{\pi k}{n}\right) & \sin^2\left(\frac{\pi k}{n}\right) \end{bmatrix}$$

But, we only need the bound to hold on *average*

THEOREM *For every $N \geq 1$ and X_k defined above,*

$$\left\| \frac{1}{N!} \sum_{\pi \in S_N} X_{\sigma(i)} \right\|_2 = -\lambda(N) 2^{-N} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

where F is the hypergeometric function and

$$\lambda(N) = {}_2F_3 \left[\begin{matrix} 1 & -N/2+1/2 & -N/2 \\ 1/2 & -N+1 & \end{matrix} ; 1 \right] \in \mathcal{O}(N^{-1})$$

Even in this case, without replacement is asymptotically faster (on average).

Proof exploits symmetry: frame is a representation of $\mathbf{Z}_n$

More applications than a cube of ice!

- **Recommending Movies on Netflix**
  - Experts: Low-rank Factorization.
  - Old SOTA : 4+ hours.
  - In RDBMS : 2.5 hours.
  - Hazy-MM : 2 minutes.

Same Quality

Prof. Benjamin Recht

Buzzwords: *A novel parallel execution strategy for incremental gradient methods to optimize convex relaxations with constraints or proximal point operators.*

Where can DB help optimization?

Scalability: Operate on data sets much larger than memory with reasonable performance

High-level data manipulation layer.
Many apps simple models + lots of data = win.

Cost models for data access --
more than counting steps.

Conclusion

Future of data management is in managing these less precise sources

Hazy Hypothesis: Handful of statistical operators capture a diverse set of applications.

Key challenges: performance and maintenance. Hazy attacks this.