Hazy: Making Statistical Applications Easier to Build and Maintain

Christopher Ré
Computer Sciences
University of Wisconsin-Madison
Two Trends that Drive Hazy

1. Data in a large number of formats

2. Arms race to deeply understand data

Statistical tools attack both 1. and 2.

Hazy = statistical + data management

Hazy’s Goal: Find common patterns when deploying statistical tools on data.
The next breakthrough in data analysis may not be a new data analysis algorithm...

...but may be in the ability to rapidly combine, deploy, and maintain existing algorithms.
DeepDive: Slinging Signal at Scale
Build a system that is able to read the Web and answer questions.

Machine Reading: “List members of the Brazilian Olympic Team in this corpus with years of membership”
What about Barack Obama?
• wife is Michelle Obama
• went to Harvard Law School
• …

Billions of webpages
Billions of tweets
Billions of events
Billions of videos
Billions of photos
Billions of blogs

http://research.cs.wisc.edu/hazy/wisci/
Demo: DeepDive

Tasks we perform:
- Web Crawling
- Information Extraction
- Deep Linguistic Processing
- Audio/Video Transcription
- Tera-byte Parallel Joins

Some Information
- 50TB Data
- 500K Machine hours
- 500M Webpages
- 400K Videos
- 7Bn Entity Mentions
- 114M Relationship Mentions

Declare graphical models at Web scale
100 Nodes
100 TB
X 1000 @ UW-Madison
X 100K @ US Open
Science Grid

100 Nodes
100 TB
100K Machine-Hrs
14B structured sentences

Magic Happens!

3M Entities
7B Mentions
100M Relations

500M Webpages
500K Videos
50TB Data

Statistical Inference

Web Serving

DeepNLP

MaltParser

Data Acquisition

Raw Compute Infrastructure

Storage Infrastructure

Stats. Infrastructure
One goal of my work is to turn data analysis into a commodity.
Other Places we sling together signal

How much carbon is in North America? (Peak Oil? Policy?)

UW Geoscience, Shanan Peters

Detect Neutrinos.

**Our Code:** By next year, 100k+ neutrinos detected out of 100Bn+ Events.
Application Takeaways

Statistical + data management enables a wide variety of new applications.

**Goal:** framework to rapidly combine, deploy, & maintain existing algorithms.

One Framework to Sling Signals at Scale...and one technical nugget.
Elementary: A Framework to Sling Signal at Scale
Markov Logic by Example

Rules
3: \(\text{wrote}(s,t) \land \text{advisedBy}(s,p) \rightarrow \text{wrote}(p,t)\) //students’ papers tend to be co-authored by advisors
5: \(\text{advisedBy}(s,p) \land \text{advisedBy}(s,q) \rightarrow p = q\)
\(-\infty: \text{advisedBy}(s,p) \rightarrow \text{professor}(p)\)
...

Evidence
- \(\text{wrote}(\text{Tom}, \text{Paper1})\)
- \(\text{wrote}(\text{Tom}, \text{Paper2})\)
- \(\text{wrote}(\text{Jerry}, \text{Paper1})\)
- \(\text{professor}(\text{John})\)
...

\(\text{advisedBy}(?, ?)\) //who advises whom

*Variables universally quantified.*
Semantics of Markov Logic Networks

For each weighted formula, ground each free variable in all possible ways:
output is a set \((w,r)\) where \(w\) is a weight and \(r\) is a propositional rule.

\[
\Pr(I) = Z^{-1} \exp \left( \sum_{(w,r) \in \Gamma} w \mathbb{1}[I \models r] \right)
\]

\(Z\) is a normalizing constant

Standard exponential model: MaxEnt Justified
MLN Inference in Two Steps

3 wrote(s, t) ∧ advisedBy(s, p) → wrote(p, t)

Step 1: Grounding

3 wrote(Tom, P1), advisedBy(Tom, Jerry) → wrote(Jerry, P1)
3 wrote(Tom, P1), advisedBy(Tom, Bob) → wrote(Bob, P1)
3 wrote(Bob, P1), advisedBy(Bob, Jerry) → wrote(Jerry, P1)

Step 2: Sample

Markov Random Field (MRF)

<table>
<thead>
<tr>
<th>advisee</th>
<th>advisor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tom</td>
<td>Jerry</td>
</tr>
<tr>
<td>Tom</td>
<td>Bob</td>
</tr>
</tbody>
</table>
Main Idea to improve Grounding

**Input:** An MLN program.

**Output:** A database of facts (the MRF)

Challenge: MRF may be huge – larger than main memory.

Similar challenge to the datalog/prolog era. Maybe an RDBMS can help?
Grounding via SQL in Tuffy

Program Transformed into many SQL queries (Bottom-up)

3 \(\text{wrote}(s, t) \land \text{advisedBy}(s, p) \rightarrow \text{wrote}(p, t)\)

\[
\begin{align*}
\text{SELECT} & \quad w1.id, a.id, w2.id \\
\text{FROM} & \quad \text{wrote} \ w1, \ \text{advisedBy} \ a, \ \text{wrote} \ w2 \\
\text{WHERE} & \quad w1.person = a.advisee \ \text{AND} \ w1.paper = w2.paper \\
& \quad \text{AND} \ a.advisor = w2.person \ \text{AND} \ldots
\end{align*}
\]

An RDBMS

RDBMS executes SQL using sophisticated algorithms
Grounding: Top-down vs. Bottom-up

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Relational Classification</th>
<th>Entity Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Alchemy</strong></td>
<td>4000 sec</td>
<td>7 hr</td>
</tr>
<tr>
<td>[C++]</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Tuffy</strong></td>
<td>40 sec</td>
<td>3 min</td>
</tr>
<tr>
<td>[PostgreSQL]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Traditional data processing techniques have something to say

Lesion study: join algorithms avoid $N^2$ blow up.
R(x,y,z) :- C(x,y), C(y,z), C(z,x)

R returns all triangles, e.g., Coreferent Triples.

Q: If C contains N tuples, how big can R be?

How many triangles can a graph with N edges contain? (In query, Nodes = constant values and Edges = Tuples)

Right asymptotic is |R| = \( \Theta(N^{3/2}) \)

Known as Loomis-Whitney, Bollabás-Thomason, Fractional Cover Inequality…

Can we compute R in time O(N^{3/2})?
Sketch: The Power of Two Choices

A simple algorithm allows us to run in time $O(N^{3/2})$ -- which is worst case optimal!

**Main technical idea:** Call a node **heavy** if it has more than $N^{1/2}$ neighbors. Let $H$ be set of heavy nodes.

**Two Cases (Let’s ignore log factors)**

(1) **If v in H, check all edges to see if they form a triangle with v.**

Time at most $N|H|$.
Since $|H| \leq N^{1/2}$
Total time: $\leq N^{3/2}$

SUM_{i} \in (i, \infty) \cdot \text{in}(i) \cdot \text{out}(i) \leq N^{3/2}
(1, \infty)-Hölder.

Union linear in output

(2) **If v not in H, hash each pair leaving v into C.**
Our Results

**Positive:** Generalize to *arbitrary* join queries.
- First algorithm with worst-case guarantees for all join algorithm [PODS 2012, best paper]
- These bounds recently studied in FOCS, STOC, & PODS. We showed this line of work is equivalent to Bollabás-Thomason (1990s) in Geometry.
- First constructive proof of Loomis-Whitney (1940s).
- Same intuition, only more involved 😊

**Negative:** Any traditional database plan is asymptotically slower on some inputs.
Other Hazy work

Dealing with data in different formats
- **Staccato. Arun Kumar.** Query OCR documents [PODS 10 & VLDB 12]
- **Hazy goes to the South Pole. Mark Wellons** Trigger Software for IceCube

Machine Learning Algorithms
- **Hogwild! IGD on convex problems with no locking** [NIPS 11] with B. Recht and S. Wright
- **Jellyfish** faster than Hogwild! on Matrix Completion [Opt Online 11] with B. Recht
- **Non-commutative Arithmetic-Geometric Means** [COLT 2012] with B. Recht
- **Bismarck** in-database analytics. **Arun Kumar and Aaron Feng** [SIGMOD 12]

Maintain Supervised Techniques on Evolving Corpora
- **iClassify. M. Levent Koc.** Maintain classifiers on evolving examples [VLDB 11]
- **CRFLex. Aaron Feng.** Maintain CRFs on Evolving Corpora [ICDE 12]

Populate Knowledge Bases from Text
- **Tuffy.** SQL + weights to combine multiple models [VLDB 11]
- **Felix.** Markov Logic on Millions of Documents [ACL2012, IJWIS12, VLDS2012]
  - **Jude Shavlik, Feng Niu, Ce Zhang, Josh Slauson, and Xixi Luo**

Database Theory
- **Worst-case Optimal Join Algorithms** [PODS 2012] with H. Ngo and A. Rudra, **Best Paper**
- **Cardinality Estimation using Entropy Maximization** [PODS 2010] with D. Suciu, **Best Of**
- **Transducing Markov Sequences** [PODS 2010] with B. Kimmelfeld, **Best Of**
Conclusion

Argued that the next breakthrough is in the ability to quickly sling signal at scale.

**Today**: One framework to sling signal at scale (Elementary)

Download everything!
http://www.cs.wisc.edu/hazy
Thank you to our Sponsors