MCMC Inference Inside the DB for Extraction, Resolution, Alignment, Provenance and Queries

Andrew McCallum

Department of Computer Science
University of Massachusetts Amherst

Joint work with Michael Wick, Karl Schultz, Sameer Singh, Sebastian Reidel.
1997: KB of research papers
2003: New KB: papers, people,...
1. Table extraction using conditional random fields
   David Pinto, Andrew McCallum, Xin Wei, W. Bruce Croft
   SIGIR, 2003
   The ability to find tables and extract information from them is a necessary component of data mining, question answering, and other information retrieval tasks. Documents often contain tables in order to communicate densely packed, multi-dimensional information. Tables do this by employing layout patterns to efficiently indicate fields and records in two-dimensional form. Their rich combination of formatting and content present difficulties for traditional language modeling techniques, however. This paper presents ... (17 citations)

2. Learning table extraction from examples
   A. Tengli, Yun Yang, Nianli Ma
   In Proceedings of the 20th International Conference on Computational Linguistics (COLING, 2004) (0 citations)

3. Computational Aspects of Resilient Data Extraction from Semistructured Sources
   Hasan Davulcu, Guizhen Yang, Michael Kifer, idhar Ramakrishnan
   PODS, 2000
   Automatic data extraction from semistructured sources such as HTML pages is rapidly growing into a problem of significant importance, spurred by the growing popularity of the so called "shopbots" that enable end users to compare prices of goods and other services at various web sites without having to manually browse and fill out forms at each one of these sites. The main problem one has to contend with when designing (5 citations)

4. Learning Information Extraction Rules for Semi-Structured and Free Text
   Stephen Soderland
   Machine Learning vol 34, pages 233, 1999
   A wealth of on-line text information can be made available to automatic processing by information extraction (IE) systems. Each IE application needs a separate set of rules tuned to the domain and writing style. WHISK helps to overcome this knowledge-engineering bottleneck by learning text extraction rules automatically. WHISK is designed to handle text styles ranging from highly structured to free text, including text that is neither rigidly formatted nor composed (82 citations)

5. Automatic Table Ground Truth Generation and a Background-Analysis-Based Table Structure Extraction Method

Done
Table extraction using conditional random fields

David Pinto, Andrew McCallum, Xin Wei, W. Bruce Croft

SIGIR, 2003

Abstract:
The ability to find tables and extract information from them is a necessary component of data mining, question answering, and other information retrieval tasks. Documents often contain tables in order to communicate densely packed, multi-dimensional information. Tables do this by employing layout patterns to efficiently indicate fields and records in two-dimensional form. Their rich combination of formatting and content present difficulties for traditional language modeling techniques, however. This paper presents the use of conditional random fields (CRFs) for table extraction, and compares them with hidden Markov models (HMMs). Unlike HMMs, ...

References: (16) Sorted by date | citations | alphabetically
- Fei Sha, Fernando C N Pereira. Shallow Parsing with Conditional Random Fields. HLT-NAACL, 2003 (42 citations)
- Andrew Kachites McCallum. MALLET: a machine learning for language toolkit. 2002 (9 citations)
- David Pinto, Michael S. Brandstein, RE Coleman, W. Bruce Croft, Matthew King, Wei Li, Xin Wei. QuASM: a system for question answering using semi-structured data. JCDL, 2002 (2 citations)
- Martin J. Wainwright, Tommi Jaakkola, Alan S. Willsky. Exact MAP Estimates by (Hyper)tree Agreement. NIPS, 2002 (5 citations)
- John Lafferty, Andrew McCallum, Fernando C N Pereira.

Bibtex Entry: [Edit]
@inproceedings{pinto2003table,
    author = "David Pinto and Andrew McCallum and Xin Wei and W. Bruce Croft",
    title = "Table extraction using conditional random fields",
    booktitle = "SIGIR",
    pages = "235",
    year = "2003"
}

Topics:
- experimental results (20.2%), classification (13.1%), information retrieval (10.1%), speech recognition (9.1%), operations (7.1%), en automatique (6.1%), data (4%), escherichia coli (3%)

Citations: (17) Sorted by date | citations | alphabetically
- Trevor Cohn, Alvy Ray Smith, Melissa Osborne. Scaling Conditional Random Fields Using Error-Correcting Codes. Association for Computational Linguistics, pages 10-17, 2005 (2 citations)
- Charles A. Sutton, Khashayar Rohanimanesh, Andrew McCallum. Dynamic conditional random fields: factored probabilistic models for labeling and segmenting sequence data. ICML, 2004 (8 citations)
W. Bruce Croft

Distinguished Professor
Department of Computer Science, University of Massachusetts
BRUCE CROFT, Amherst, MA, 01003-9264
Email: croftg@cs.umass.edu
URL: http://ciir.cs.umass.edu/personnel/croft.html

Publications: (1 to 40 of 233) (total 1436 citations)
Sorted by date | citations

2004
• Donald Metzler, W. Bruce Croft. Combining the language model and inference network approaches to retrieval. Inf. Process. Manage. vol 40, pages 735, 2004 (1 citation)
• Xiaoyong Liu, W. Bruce Croft. Cluster-based retrieval using language models. SIGIR, 2004 (0 citations)
• Andrés Corrada-Emmanuel, W. Bruce Croft. Answer models for question answering passage retrieval. SIGIR, 2004 (0 citations)
• Chirag Shah, W. Bruce Croft. Evaluating high accuracy retrieval techniques. SIGIR, 2004 (1 citation)
• Haizheng Zhang, W. Bruce Croft, Brian N. Levine, Victor R. Lesser. A Multi-Agent Approach for Peer-to-Peer Based Information Retrieval System. AAMAS, 2004 (1 citation)
• Donald Metzler, Victor Lavrenko, W. Bruce Croft. Formal multiple-bernoulli models for language modeling. SIGIR, 2004 (0 citations)
• Stephen Cronen-Townsend, Yu Zhou, W. Bruce Croft. A framework for selective query expansion. CIKM, 2004 (0 citations)

2003
• W. Bruce Croft. Language Models for Information Retrieval. ICDE, 2002 (0 citations)

Co-authors | Cited authors | Citing authors: (1 to 40 of 257)
Sorted by date | number | name

• Stephen Cronen-Townsend, 2004 2002 2001
• Donald Metzler, 2004 2004 2003
• Xiaoyong Liu, 2004 2002
• Andrés Corrada-Emmanuel, 2004
• Victor R. Lesser, 2004
• Brian N. Levine, 2004
• Chirag Shah, 2004
• Haizheng Zhang, 2004
• Yu Zhou, 2004
CISE Research Infrastructure: Infrastructure to Support Research on Networked Multimedia Information Systems

James F. Kurose, John A. Stankovic, Donald F. Towsley, Krithi Ramamritham, J. Eliot B Moss, W. Richards Adrian, W. Bruce Croft, Kathryn McKinley
NSF Grant EIA-9502639, August 1, 1995 - December 29, 1999

Abstract:
This award provides support to equip a networked, experimental testbed to enable research in the development of the operating system, I/O, networking, object management, and information retrieval components of future networked multimedia information systems. The testbed will consist of two shared-memory multiprocessor facilities attached to several parallel mass storage I/O devices and a high-speed ATM network. The research team will be developing several key hardware and software technologies needed to support future networked, multimedia information systems. Specific research areas include operating systems, I/O, networking, object management and information retrieval.

Papers: (17) Sorted by date | citations | alphabetically
This may be only a partial list of papers for this grant:

- Emery D. Berger, Benjamin G. Zom, Kathryn S. McKinley. Composing High-Performance Memory Allocators. PLDI, 2001 (7 citations)
- Sally Floyd, Mark Handley, Jitendra Padhye, Jörg Widmer. Equation-based congestion control for unicast applications. SIGCOMM, 2000 (229 citations)
- Sally Floyd, Mark Handley, Jitendra Padhye. Equation-Based Congestion Control for Unicast Applications \Lambda. 2000
- Supratik Bhattacharyya, Don Towsley, James F. Kurose. Design and Analysis of Loss Indication Filters for Multicast Congestion Control. CMPSCI Technical Report TR 99-46, Department of Computer Science University of Massachusetts Amherst, 2000 (0 citations)
- Brendon Cahoon, Kathryn S. McKinley. Tolerating Latency by Prefetching Java Objects. To appear: Workshop on Hardware Support for Objects and Microarchitectures for Java, 1999 (3 citations)
- Jitendra Padhye, James F. Kurose, Donald F. Towsley, Rajeev Koodli. A TCP-Friendly Rate Adjustment Protocol for Continuous Media Flows over Best Effort Networks CMPSCI
Rexa KB

- 8 million research papers
- 2 million authors
- 400k grants, 90k institutions, 10k venues
- User corrections
Steps to Building a KB

Cartoon: missing ontology discovery, schema alignment,…

Gather raw data → Extraction → Resolution → Entities “truth”

query

answer

Information Extraction isn’t perfect. Uncertainty.
1. How to represent & inject uncertainty from IE into DB?
2. IE isn’t “one-shot.” Add new data later; redo inference.
3. Want DB infrastructure to manage IE.
4. Want to use DB contents to aid IE.
Steps to Building a KB

“Truth is inferred, not observed.”
Steps to Building a KB

“Truth is inferred, not observed.”
“Truth is inferred, not observed.”

*Constructivist Epistemology*

**Probabilistic DBs & Epistemology**

Database stores “knowledge” = the “truth”

**Many Prob DBs:**
- Inference is for answering queries about the “truth”.
- “Truth” injected from other components, representing uncertainty in their output.

**I advocate:**
- Inference for queries, *plus* for discovering the “truth” from raw observations, (e.g. extraction, matching).
- Represent all dependencies necessary for truth discovery models, (which are probabilistic).
Practical Motivations

• Incremental KB
  - Ability to add new data, evidence, corrections
    - E.g. [Chu,...Doan 2007], [Mansuri, Sarawagi, 2006],...

• Uncertainty
  - Ability to reflect uncertainty in query answers
    - E.g. Suciu, Sarawagi, Koch, Deshpande, Re,...
  - Ability to change the “truth” with new evidence

• Unified infrastructure
  - Joint inference in extraction, matching & other inference
    - E.g. Weikum group’s “SOPHIE”
  - Raw data, provenance, human corrections in same system

Must represent all dependencies necessary for performing IE...
Outline

• Motivate “Inference inside the DB”

• Graphical models for Extraction & Integration
  - Extraction  (linear-chain CRFs)
  - Coreference  (pairwise & entity-wise CRFS, MCMC)
  - Information Integration  (really hairy CRFs, MCMC)

• Probabilistic Programming: FACTORIE

• Probabilistic Programming inside a DB

• Ongoing Work
Conditional Random Fields

Undirected graphical model, trained to maximize conditional probability of output (sequence) given input (sequence)

Finite state model

Graphical model

\[ p(y|x) = \frac{1}{Z_x} \prod_{t=1}^{\bar{x}} \phi(y_t, y_{t-1}) \phi(x_t, y_t) \exp \left( \sum_k \lambda_k f_k(x_t, y_t) \right) \]
Information Extraction with Linear-chain CRFS

State-of-the-art accuracy on many tasks.
Outline

• Motivate “Inference inside the DB”
• Graphical models for Extraction & Integration
  - Extraction  (linear-chain CRFs)
  - Coreference  (pairwise & entity-wise CRFS, MCMC)
  - Information Integration  (really hairy CRFs, MCMC)
• Probabilistic Programming: FACTORIE
• Probabilistic Programming inside a DB
• Ongoing Work
Entity Resolution

- Mr. Hill
- Dana Hill
- Amy Hall
- Dana
- She

“mention”
Entity Resolution

Mr. Hill

Amy Hall

Dana Hill

Dana

she

“entity”

“entity”
Entity Resolution

“entity”

Mr. Hill

Dana Hill

Dana

Amy Hall

she
Entity Resolution

Mr. Hill

Dana Hill

Amy Hall

Dana

she
CRF for Co-reference
CRF for Co-reference

Make pair-wise merging decisions *jointly* by:
- calculating a joint prob.
- including all edge weights
- enforcing transitivity.

\[
p(y|\vec{x}) = \frac{1}{Z_{\vec{x}}} \exp \left( \sum_{i,j} \sum_{l} \lambda_l f_l(x_i, x_j, y_{ij}) \right) + \text{mechanism for preserving transitivity}
\]
Pairwise Affinity is not Enough
Pairwise Affinity is not Enough
Pairwise Comparisons Not Enough Examples:

- \( \forall \) mentions are pronouns?
- Entities have multiple attributes \((name, email, institution, location)\); need to measure “compatibility” among them.
- Having 2 “given names” is common, but not 4.
  - e.g. Howard M. Dean / Martin, Dean / Howard Martin
- Need to measure size of the clusters of mentions.
- \( \exists \) a pair of lastname strings that differ > 5?

We need to ask \( \exists, \forall \) questions about a set of mentions
We want first-order logic!
Pairwise Affinity is not Enough
Ask arbitrary questions about all entities in a partition with *first-order logic*...
Partition Affinity CRF
Partition Affinity CRF
Partition Affinity CRF
Partition Affinity CRF
How can we perform inference and learning in models that cannot be “unrolled”? 

Can’t use belief propagation. 
Can’t use standard integer linear programming.
Don’t represent all alternatives...
Don’t represent all alternatives... just one at a time

Markov Chain Monte Carlo
Metropolis-Hastings

Given factor graph with target variables $y$ and observed $x$

$$
P(y|x) = \frac{1}{Z_X} \prod_{y^i \in \mathcal{F}} \psi(x, y^i)
$$

$\mathcal{F}$ feasible region defined by deterministic constraints
e.g. clustering, parse-tree projectivity.

$q$ proposal distribution $q(y'|y) : \mathcal{F} \times \mathcal{F} \to [0, 1]$

1. Begin with some initial configuration $y_0 \in \mathcal{F}$
2. For $i=1,2,3,\ldots$ draw a local modification $y' \in \mathcal{F}$ from $q$
3. Probabilistically accept jump as Bernoulli draw with param $\alpha$

$$
\alpha = \min \left( 1, \frac{p(y')} {p(y) \cdot q(y'|y)} \right)
$$

Can do MAP inference with decreasing temperature on ratio of $p(y)$’s
M-H Natural Efficiencies

1. Partition function cancels

\[
\frac{p(y')}{p(y)} = \frac{p(Y = y'|x; \theta)}{p(Y = y|x; \theta)} = \frac{1}{Z_x} \prod_{y_i \in y'} \psi(x, y^i) = \frac{1}{Z_x} \prod_{y \in y} \psi(x, y^i) = \frac{\prod_{y_i \in y'} \psi(x, y^i)}{\prod_{y \in y} \psi(x, y^i)}
\]

2. Unchanged factors cancel

\[
\frac{p(y')}{p(y)} = \frac{1}{Z_x} \prod_{y_i \in y'} \psi(x, y^i) = \frac{\prod_{y_i \in y} \psi(x, y^i)}{\prod_{y \in y} \psi(x, y^i)} = \frac{\prod_{y_i \in y^i} \psi(x, y^i)}{\prod_{y_i \in \delta_y} \psi(x, y^i)} \frac{\prod_{y_i \in y'} \psi(x, y^i)}{\prod_{y_i \in \delta_y} \psi(x, y^i)} = \frac{\prod_{y_i \in \delta_y} \psi(x, y^i)}{\prod_{y_i \in \delta_y} \psi(x, y^i)}
\]

How to learn parameters \( \theta \) for \( p(Y = y|x; \theta) \) ?

Sample Rank
Partition Affinity Results

Leads to best-in-the-world results.
DARPA ACE coref: 69% → 82% $B^3$
Supports inference of canonical entity representation
“inferred truth”
Outline

• Motivate “Inference inside the DB”

• Graphical models for Extraction & Integration
  - Extraction (linear-chain CRFs)
  - Coreference (pairwise & entity-wise CRFs, MCMC)
  - Information Integration (really hairy CRFs, MCMC)

• Probabilistic Programming: FACTORIE

• Probabilistic Programming inside a DB

• Ongoing Work
### Information Integration

#### Database A (Schema A)

<table>
<thead>
<tr>
<th>First Name</th>
<th>Last Name</th>
<th>Contact</th>
</tr>
</thead>
<tbody>
<tr>
<td>J.</td>
<td>Smith</td>
<td>222-444-1337</td>
</tr>
<tr>
<td>J.</td>
<td>Smith</td>
<td>444 1337</td>
</tr>
<tr>
<td>John</td>
<td>Smith</td>
<td>(1) 4321115555</td>
</tr>
</tbody>
</table>

#### Database B (Schema B)

<table>
<thead>
<tr>
<th>Name</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Smith</td>
<td>U.S. 222-444-1337</td>
</tr>
<tr>
<td>John D. Smith</td>
<td>444 1337</td>
</tr>
<tr>
<td>J Smith</td>
<td>432-111-5555</td>
</tr>
</tbody>
</table>

#### Schema Matching

<table>
<thead>
<tr>
<th>Schema A</th>
<th>Schema B</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Name</td>
<td>Name</td>
</tr>
<tr>
<td>Last Name</td>
<td>Phone</td>
</tr>
<tr>
<td>Contact</td>
<td></td>
</tr>
</tbody>
</table>

#### Coreference

- **John #1**
  - J. Smith
  - J. Smith
  - John Smith
  - John D. Smith

- **John #2**
  - John Smith
  - J Smith

#### Canonicalization:

<table>
<thead>
<tr>
<th>Entity#</th>
<th>Name</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>523</td>
<td>John Smith</td>
<td>222-444-1337</td>
</tr>
<tr>
<td>524</td>
<td>John D. Smith</td>
<td>432-111-5555</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Schema Matching

\[ P(Y \mid X) = \frac{1}{Z_X} \prod_{y_i \in Y} \psi_{y_i}(y_i, x_i) \prod_{y_j \in Y} \psi_{y_j}(y_j, x_j) \]

\[ \psi(y_i, x_i) = \exp \left( \sum_k \lambda_k f_k(y_i, x_i) \right) \]

- \( x_6 \) is a set of attributes \{phone, contact, telephone\}
- \( x_7 \) is a set of attributes \{last name, last name\}
- \( f_{67} \) is a factor between \( x_6 \)/\( x_7 \)
- \( y_{67} \) is a binary variable indicating a match (no)
- \( f_7 \) is a factor over cluster \( x_7 \)
- \( y_7 \) is a binary variable indicating match (yes)
• $x_1$ is a set of mentions \{J. Smith, John, John Smith\}
• $x_2$ is a set of mentions \{Amanda, A. Jones\}
• $f_{12}$ is a factor between $x_1/x_2$
• $y_{12}$ is a binary variable indicating a match \(\text{no}\)
• $f_1$ is a factor over cluster $x_1$
• $y_1$ is a binary variable indicating match \(\text{yes}\)
• Entity/attribute factors omitted for clarity

**Coreference and Canonicalization**
Schema Matching

\[ P(Y \mid X) = \frac{1}{Z_X} \prod_{y_i \in Y} \psi_w(y_i, x_i) \prod_{y_i, y_j \in Y} \psi_b(y_{ij}, x_{ij}) \]

\[ \psi(y_i, x_i) = \exp \left( \sum_k \lambda_k f_i(y_i, x_i) \right) \]

Coreference and Canonicalization
**Dataset**

- Faculty and alumni listings from university websites, plus an IE system
- 9 different schemas
- ~1400 mentions, 294 coreferent

**Example schemas:**

<table>
<thead>
<tr>
<th>DEX IE</th>
<th>Northwestern Fac</th>
<th>UPenn Fac</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Name</td>
<td>Name</td>
<td>Name</td>
</tr>
<tr>
<td>Middle Name</td>
<td>Title</td>
<td>First Name</td>
</tr>
<tr>
<td>Last Name</td>
<td>PhD Alma Mater</td>
<td>Last Name</td>
</tr>
<tr>
<td>Title</td>
<td>Research Interests</td>
<td>Job+Department</td>
</tr>
<tr>
<td>Department</td>
<td></td>
<td>Office Address</td>
</tr>
<tr>
<td>Company Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Home Phone</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Office Phone</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fax Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E-mail</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Coreference Results

<table>
<thead>
<tr>
<th></th>
<th>Pair</th>
<th></th>
<th>MUC</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F1</td>
<td>Prec</td>
<td>Recall</td>
<td>F1</td>
</tr>
<tr>
<td>No Canon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ISO</td>
<td>72.7</td>
<td>88.9</td>
<td>61.5</td>
<td>75.0</td>
</tr>
<tr>
<td>CASC</td>
<td>64.0</td>
<td>66.7</td>
<td>61.5</td>
<td>65.7</td>
</tr>
<tr>
<td>JOINT</td>
<td>76.5</td>
<td>89.7</td>
<td>66.7</td>
<td>78.8</td>
</tr>
<tr>
<td>Canon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ISO</td>
<td>78.3</td>
<td>90.0</td>
<td>69.2</td>
<td>80.6</td>
</tr>
<tr>
<td>CASC</td>
<td>65.8</td>
<td>67.6</td>
<td>64.1</td>
<td>67.6</td>
</tr>
<tr>
<td>JOINT</td>
<td>81.7</td>
<td>90.6</td>
<td>74.4</td>
<td>84.1</td>
</tr>
</tbody>
</table>

ISO = isolated  CASC = cascade  JOINT = joint inference

~15% error reduction from joint model
## Schema Matching Results

<table>
<thead>
<tr>
<th></th>
<th>Pair</th>
<th>MUC</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F1</td>
<td>Prec</td>
<td>Recall</td>
<td>F1</td>
</tr>
<tr>
<td>No Canon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ISO</td>
<td>50.9</td>
<td>40.9</td>
<td>67.5</td>
<td>69.2</td>
</tr>
<tr>
<td>CASC</td>
<td>50.9</td>
<td>40.9</td>
<td>67.5</td>
<td>69.2</td>
</tr>
<tr>
<td>JOINT</td>
<td>68.9</td>
<td>100</td>
<td>52.5</td>
<td>69.6</td>
</tr>
<tr>
<td>Canon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ISO</td>
<td>50.9</td>
<td>40.9</td>
<td>67.5</td>
<td>69.2</td>
</tr>
<tr>
<td>CASC</td>
<td>52.3</td>
<td>41.8</td>
<td>70.0</td>
<td>74.1</td>
</tr>
<tr>
<td>JOINT</td>
<td>71.0</td>
<td>100</td>
<td>55.0</td>
<td>75.0</td>
</tr>
</tbody>
</table>

ISO = isolated    CASC = cascade    JOINT = joint inference

~60% error reduction from joint model
Ontology Alignment

Illinois Semantic Integration Archive

- Course catalog hierarchy
  - 104 concepts
  - 4360 data records
- Company profile hierarchy
  - 219 concepts
  - 23139 data records

(F1)

<table>
<thead>
<tr>
<th></th>
<th>Course Catalog</th>
<th>Company Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLUE</td>
<td>89.9</td>
<td>81.5</td>
</tr>
<tr>
<td>[Doan 2003]</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Partition CRFs</strong></td>
<td><strong>94.3</strong></td>
<td><strong>84.5</strong></td>
</tr>
<tr>
<td>[Wick, R, McCallum 2008]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

state-of-the-art
Schema Matching

\[ P(Y \mid X) = \frac{1}{Z_X} \prod_{y_i \in Y} \psi_{y_i}(y_i, x_i) \prod_{y_i \neq y_j \in Y} \psi_{y_i, y_j}(y_i, x_i) \]

\[ \psi(x_i) = \exp \left( \sum_{k} \lambda_k f_k(x_i) \right) \]
Outline

• Motivate “Inference inside the DB”
• Graphical models for Extraction & Integration
  - Extraction (linear-chain CRFs)
  - Coreference (pairwise & entity-wise CRFs, MCMC)
  - Information Integration (really hairy CRFs, MCMC)
• Probabilistic Programming: FACTORIE
• Probabilistic Programming inside a DB
• Ongoing Work
Probabilistic Modeling in the Last Few Years

• Models ever growing in richness and variety
  - hierarchical
  - spatio-temporal
  - relational
  - infinite

Complex, dynamic dependency structures

Developing the representation, reasoning and learning for a new model is a significant task.
Probabilistic Programming Languages

• Make it easy to represent rich, complex models, using the full power of programming languages
  - data structures
  - control mechanisms
  - abstraction

• Inference and learning come for free (or sort of)

Provides language to easily create new models
Small Sampling of Probabilistic Programming Languages

- Logic-based
  - Markov logic, BLOG, PRISM

- Functional
  - IBAL, Church

- Object Oriented
  - Figaro, Infer.NET
Our Approach to Probabilistic Programming

- **Object-oriented**: Variables, factors, inference & learning methods are objects, inheritance...

- **Embedded** in a general-purpose programming language.

- **Scalable** to many millions of variables and factors. Optional DB back-end.

[McCallum, Rohanemanesh, Wick, Schultz, Singh, 2008]
Our Approach to Probabilistic Programming

“Imperatively-defined Factor Graphs”

Traditional **declarative** semantics of factor graphs, with some **imperative** definition of construction & operation.

- Imperatively defined jump functions
- Imperative variable value coordination
- Imperatively defined mapping from neighbor variables to features
- Imperatively defined model structure

[McCallum, Rohanemanesh, Wick, Schultz, Singh, 2008]

```scala
// Joint segmentation, classification, coref on entities
// DATA TEMPLATES
class Document extends VariableSequence[Token]
class Token(word:String) extends CategoricalVariable(word)
class Mention extends SpanVariable[Token] {
  val entity = new RefVariable[Entity]
}
class Entity extends SetVariable[Mention] {
  var canonical:String = ""
  def add(m:Mention, d:DiffList) = {
    super.add(m,d); m.set(this,d)
    canonical = recomputeCanonical(members)
  }
  def remove(m:Mention, d:DiffList) = {
    super.remove(m,d); m.set(null,d)
    canonical = recomputeCanonical(members)
  }
}

// FACTOR TEMPLATES
  def unroll1 (m:Mention) = Factor(m, m.entity)
  def unroll2 (e:Entity) = for (mention <- e.mentions)
    yield Factor(mention, e)
  def statistics(m:Mention, e:Entity) =
    Bool(distance(m.string, e.canonical) < 0.5)
}

// INFERENCE
val sampler = new ProposalSampler[Mention] {
  def propose(m:Mention) = {
    // Move Mention m to a randomly-sampled Entity.
    entities.sample.add(m)
  }
}
val documents = loadData()
sampler.process(documents.mentions), 1)
```
• “Factor Graphs, Imperative, Extensible”
• Implemented as a library in Scala [Martin Odersky]
  - object oriented & functional
  - type inference
  - runs in JVM (complete interoperation with Java)
• Library, not new “little language”
  - integrate data pre-processing & eval. w/ model spec
  - leverage OO-design: modularity, encapsulation, inheritance
• Scalable
  - large input data, factors, graphical model tree width
  - efficient discriminative learning
• Integrate declarative & procedural knowledge
  - Flexible, natural, easy-to-use

http://code.google.com/p/factorie
Experimental Results

• Joint Segmentation & Coreference of research paper citations.
  - 1295 mentions, 134 entities, 36487 tokens

• Compare with MLNs (Alchemy)
  - Same observable features

• FACTORIE results:
  - ~25% reduction in error (segmentation & coref)
  - 3-20x faster

- coref results:

<table>
<thead>
<tr>
<th></th>
<th>Prec/Recall</th>
<th>F1</th>
<th>Cluster Rec.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fellegi-Sunter Joint MLN</td>
<td>78.0/97.7</td>
<td>86.7</td>
<td>62.7</td>
</tr>
<tr>
<td></td>
<td>94.3/97.0</td>
<td>95.6</td>
<td>78.1</td>
</tr>
<tr>
<td>Isolated Joint</td>
<td>97.09/95.42</td>
<td>96.22</td>
<td>86.01</td>
</tr>
<tr>
<td></td>
<td>95.34/98.25</td>
<td>96.71</td>
<td>94.62</td>
</tr>
</tbody>
</table>
Outline

• Motivate “Inference inside the DB”
• Graphical models for Extraction & Integration
  – Extraction (linear-chain CRFs)
  – Coreference (pairwise & entity-wise CRFs, MCMC)
  – Information Integration (really hairy CRFs, MCMC)
• Probabilistic Programming: FACTORIE
• Probabilistic Programming inside a DB
• Ongoing Work
Traditional AKBC with Pr DB

Documents → Extraction & Matching → Database
Traditional AKBC with Pr DB

Documents

Extraction & Matching

Database
Traditional AKBC with Pr DB
Traditional AKBC with Pr DB

Documents

Extraction & Matching

[Diagram of data flow]
Traditional AKBC with Pr DB

Documents

Extraction & Matching

query

prob

infer

answer

query
AKBC with Inference inside DB

Documents
AKBC with Inference inside DB

Documents
AKBC with Inference inside DB

Documents

Extraction & Matching

DB contains only one possible world at a time.
AKBC with Inference inside DB

Documents

Extraction & Matching

MH inference
AKBC with Inference inside DB

Documents

Extraction & Matching

MH inference
AKBC with Inference inside DB

Documents

Extraction & Matching

MH inference
AKBC with Inference inside DB

Documents

Extraction & Matching

query

MH inference

SQL answer
AKBC with Inference inside DB

Documents

Extraction & Matching

query

MH inference

SQL answer
AKBC with Inference inside DB

Documents

Extraction & Matching

query

MH inference

SQL answer
AKBC with Inference inside DB

Documents

Extraction & Matching

MH inference

SQL answer
AKBC with Inference inside DB

Documents

Extraction & Matching

query

MH inference

SQL answer
AKBC with Inference inside DB

Documents

Extraction & Matching

query

MH inference

SQL answer
AKBC with Inference inside DB

Documents

Extraction & Matching

query

MH inference

SQL answer
AKBC with Inference inside DB

Documents

Extraction & Matching

query

MH inference

SQL answer
AKBC with Inference inside DB

Documents

Extraction & Matching

query

MH inference

SQL answer

answer

[Wick, McCallum, Miklau 2010]
### Experiments in MySQL

[Wick, McCallum, Miklau 2010]

<table>
<thead>
<tr>
<th>Sentence</th>
<th>Token</th>
<th>String</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>20</td>
<td>Mr.</td>
<td>PER</td>
</tr>
<tr>
<td>3</td>
<td>21</td>
<td>Green</td>
<td>PER</td>
</tr>
<tr>
<td>3</td>
<td>22</td>
<td>said</td>
<td>O</td>
</tr>
<tr>
<td>3</td>
<td>23</td>
<td>.</td>
<td>O</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>Green</td>
<td>PER</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>chairs</td>
<td>O</td>
</tr>
<tr>
<td>Sentence</td>
<td>Token</td>
<td>String</td>
<td>Label</td>
</tr>
<tr>
<td>----------</td>
<td>-------</td>
<td>--------</td>
<td>-------</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>Mr.</td>
<td>PER</td>
</tr>
<tr>
<td>3</td>
<td>21</td>
<td>Green</td>
<td>PER</td>
</tr>
<tr>
<td>3</td>
<td>22</td>
<td>said</td>
<td>O</td>
</tr>
<tr>
<td>3</td>
<td>23</td>
<td>.</td>
<td>O</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>Green</td>
<td>PER</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>chairs</td>
<td>O</td>
</tr>
<tr>
<td>Sentence</td>
<td>Token</td>
<td>String</td>
<td>Label</td>
</tr>
<tr>
<td>----------</td>
<td>-------</td>
<td>--------</td>
<td>-------</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>Mr.</td>
<td>PER</td>
</tr>
<tr>
<td>3</td>
<td>21</td>
<td>Green</td>
<td>PER</td>
</tr>
<tr>
<td>3</td>
<td>22</td>
<td>said</td>
<td>O</td>
</tr>
<tr>
<td>3</td>
<td>23</td>
<td>.</td>
<td>O</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>Green</td>
<td>PER</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>chairs</td>
<td>O</td>
</tr>
</tbody>
</table>
### Experiments in MySQL

[Wick, McCallum, Miklau 2010]

<table>
<thead>
<tr>
<th>Sentence</th>
<th>Token</th>
<th>String</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>20</td>
<td>Mr.</td>
<td>PER</td>
</tr>
<tr>
<td>3</td>
<td>21</td>
<td>Green</td>
<td>PER</td>
</tr>
<tr>
<td>3</td>
<td>22</td>
<td>said</td>
<td>O</td>
</tr>
<tr>
<td>3</td>
<td>23</td>
<td>.</td>
<td>O</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>Green</td>
<td>PER</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>chairs</td>
<td>O</td>
</tr>
<tr>
<td>Sentence</td>
<td>Token</td>
<td>String</td>
<td>Label</td>
</tr>
<tr>
<td>----------</td>
<td>-------</td>
<td>--------</td>
<td>-------</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>Mr.</td>
<td>PER</td>
</tr>
<tr>
<td>3</td>
<td>21</td>
<td>Green</td>
<td>PER</td>
</tr>
<tr>
<td>3</td>
<td>22</td>
<td>said</td>
<td>O</td>
</tr>
<tr>
<td>3</td>
<td>23</td>
<td>.</td>
<td>O</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>Green</td>
<td>PER</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>chairs</td>
<td>O</td>
</tr>
<tr>
<td>Sentence</td>
<td>Token</td>
<td>String</td>
<td>Label</td>
</tr>
<tr>
<td>----------</td>
<td>-------</td>
<td>--------</td>
<td>-------</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>Mr.</td>
<td>PER</td>
</tr>
<tr>
<td>3</td>
<td>21</td>
<td>Green</td>
<td>PER</td>
</tr>
<tr>
<td>3</td>
<td>22</td>
<td>said</td>
<td>O</td>
</tr>
<tr>
<td>3</td>
<td>23</td>
<td>.</td>
<td>O</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>Green</td>
<td>PER</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>chairs</td>
<td>O</td>
</tr>
</tbody>
</table>
Experiments in MySQL

[Wick, McCallum, Miklau 2010]

DARPA ACE Newswire
NER task.
**Sampling under Materialized View**

[Wick, McCallum, Miklau 2010]

**Query 1:** select STRING from TOKEN where LABEL=‘PER’

1 sample per 10k proposals

Time to reduce sq-error to threshold

---

![Scalability of Query Evaluation (log scale)](image)

Legend:
- △ materialized sampling
- ○ naive sampling

**Axes:**
- x-axis: number of tuples (millions)
- y-axis: time (minutes)

**Data Points:**
- 1 million rows

---

**Diagrams:**
- [Figure 3: A skip chain conditional random field that includes “skip” edges, or factors between tokens with the same string.](image)
- [Figure 4: The benefits of view maintenance query evaluation.](image)
- [Figure 5: Scalability over several orders of magnitude.](image)
**Inference Loss, Aggregate Queries**

**Query 2**
SELECT COUNT(*)
FROM TOKEN
WHERE LABEL='B-PER'

**Query 3**
SELECT T.doc_id
FROM Token T
WHERE (SELECT COUNT(*))
FROM Token T1
WHERE T1.label='B-ORG' AND T.doc_id=T1.doc_id

*count # persons*

*documents where #persons = #orgs*

**Aggregate Query Evaluation: Normalized Loss Over Time**

![Graph showing normalized loss over time for queries 2 and 3.](image)

Legend:
- △ Query #2
- ○ Query #3

Set size comparison!
Nearly Perfect Parallelization

Parallelizing Query Evaluation

Parallelizing Query Evaluation

Legend

- Paralleled
+ Ideal linear improvement

# parallel branches (MCMC chains)

squared error

Figure 5: Multiple evaluators in parallel

Figure 6: Squared loss over time for two aggregate queries
AKBC with Inference inside DB

Documents

Extraction & Matching

query

MH inference

SQL answer

answer
Particle Filtering

Documents

Extraction & Matching

query

answer
Particle Filtering with compact representation

[Schultz, McCallum, Miklau 2010]

Documents

Extraction & Matching

query

answer
Outline

• Motivate “Inference inside the DB”

• Graphical models for Extraction & Integration
  - Extraction (linear-chain CRFs)
  - Coreference (pairwise & entity-wise CRFS, MCMC)
  - Information Integration (really hairy CRFs, MCMC)

• Probabilistic Programming: FACTORIE

• Probabilistic Programming inside a DB

• Ongoing Work
Ongoing Application Work

- **Relation Extraction from NYTimes**
  - 74 relation types
  - 200k documents, 100k entities
  - Trained (distantly) from FreeBase
  - ~3 hours to train then run on test data
  - 98% precision @100, 74% accuracy overall

- **Entity Resolution on NYTimes**
  - 1+ million mentions
  - Trained (distantly) from Wikipedia
  - 90% pairwise F1
Ongoing Methodological Work

• More efficient inference and learning. (Michael Wick)
• Particle filtering with DBs as particles. (Karl Shultz)
• Massive-scale Entity Resolution. (Sameer Singh)
• User corrections & provenance.
• Sensible query responses with identity uncertainty
• Distributed OODBs or other alternative back-ends.
• Query-specific inference. Inference caching wrt to $E[query]$.
• Mixed generative & discriminative modeling.
• Exercise this infrastructure on a substantial system.