Life of A Knowledge Base (KB)

- A *knowledge base system* is a special kind of database management system to for knowledge base management.

- **KB extraction**: knowledge extraction using statistical models in NLP/ML literature

- **KB expansion**: knowledge inference using deductive rules and knowledge with uncertainty

- **KB evolution**: knowledge update given new evidence and new data sources

- **KB integration**: knowledge integration from multiple sources (e.g., human, data sets, models)
Uncertainty Management

• Where does Uncertainty come from?
  – Inherent in the state-of-the-art NLP results
  – Incorrect, conflicting data sources
  – Derived facts and query results

• Uncertainty vs. NULL

• Uncertainty vs. MAP/majority voting

• *Probability Theory* should lay the foundation for uncertainty management \(\rightarrow\) *Probabilistic Knowledge Base System*
Probabilistic Knowledge Base System

Statistical machine learning models and data processing and management systems are the backbones.
Web Does Not Know All

• For example, the Web has much information on:
  – X Food contains Y Chemical
  – Y Chemicals prevents Z Disease

• The Web has little information on:
  – X Food prevents Z Disease

• We can infer knowledge in addition to those extracted from the Web using first-order rules
  – First-order Rules can be automatically extracted from Web [Sherlock-Holmes]
  – Uncertainties (e.g., rules, facts) needs to be populated, maintained and managed
A Markov logic network is a set of formulae with weights. Together with a finite set of constants, it defines a Markov network (i.e., factor graph).

0.859 Contains(Food, Chemical) :-
   IsMadeFrom(Food, Ingredient),
   Contains(Ingredient, Chemical)

0.855 Prevents(Food, Disease) :-
   Contains(Food, Chemical),
   Prevents(Chemical, Disease)

\[ C = \{F1, C1, D1, I1\} \]
Explicit vs. Implicit Information
Does Obama live in the US?
Knowledge Expansion
Challenges

• Scalable inference for knowledge expansion
• Rule learning
  – Inference rules, e.g.:
    PoliticianOf(Person,City) $\rightarrow$ LivesIn(Person,City)
    RetriedFrom(Prof,School) $\rightarrow$ WorkedFor(Prof,School)
  – Constraint rules, e.g.: (functional dependencies, conditional FDs)
    Every person can only have one birth place
    Every city can only belong to one country
From Big Data to Big Knowledge

- Information...
- Knowledge...
- Wisdom!
From Big Data to Big Knowledge

Data

Facts

Rules
Rule Learning over Knowledge Bases: Association Rule Mining

- T1: Milk, Eggs, Chips, Bread
- T2: Pizza, Eggs, Meat
- T3: Soda, Eggs, Milk, Meat
Association Rules

T1
- Milk
- Eggs
- Chips
- Bread

T2
- Pizza
- Eggs
- Meat

T3
- Soda
- Eggs
- Milk
- Meat

Eggs → Milk?

Meat → Eggs?
Association Rule Metrics

• **Support** (B -> H)
  – \( \text{Pr}(B \land H) \)
  – \# transactions \((B \land H) / \text{total # transactions} \)

• **Confidence** (B -> H)
  – \( \text{Pr}(H | B) \)
  – = \( \text{Pr}(H \land B) / \text{Pr}(B) \) (Bayes Rule)
  – \# transactions \((B \land H) / \# \text{transactions (B)} \)

• **Lift** (B -> H)
  – \( \text{Pr}(H \land B) / \text{Pr}(B)\text{Pr}(H) \)
  – \# transactions \((B \land H) \times \text{total # transactions} / \# \text{trans (B)} \times \# \text{trans (H)} \)
Association Rules

T1
- Milk
- Eggs
- Chips
- Bread

T2
- Pizza
- Eggs
- Meat

T3
- Soda
- Eggs
- Milk
- Meat

Eggs → Milk?
Support = 2/3
Confidence = 2/3

Meat → Eggs?
Support = 2/3
Confidence = 1
AMIE: Association Rules in a KB

- $x, y$ are instance and $X, Y$ are classes
- Items $\rightarrow p(x, y)$
  - $\text{memberOf}(x, y), \text{isAlliedWith}(x, z), \text{etc.}$
- Transactions $\rightarrow R(x,y) = \{p_i(x,y)\}, \ i=1...N$
  - $\{\text{isAlliedWith}(x,y), \text{hasFather}(x,y), \text{etc.}\}$
- Support of $p(X,Y) \rightarrow q(X,Y)$
  - $\# (X,Y) : B \ & \ H$
- Confidence of $p(X,Y) \rightarrow q(X,Y)$
  - $\# (X, Y) : B \ & \ H / \#(X, Y) : B$
Example Rule

• Extending to $X, Y, Z$
• Items -> $p(x,y), p(x,z)$ or $p(y,z)$
• Transactions -> $R(x,y,z)$
• Support of $p(X,Y) \land q(X,Y) \rightarrow r(X,Y)$
  — # $(X,Y,Z) : B \land H$
• Confidence of $p(X,Y) \rightarrow q(X,Y)$
  — # $(X,Y,Z) : B \land H / #(X,Y,Z) : B$
• Example Rule:

isAlliedWith \( (X, Y) \landmemberOf (Y, Z) \rightarrow \memberOf (X, Z) \)
Querying ProbKB’s

- SPARQL over JENA RDF store $\rightarrow ??$ And ??
- Top-K answers with ranked confidence value...
Summary

- IE/NLP techniques
- KBs and probabilistic KBs
- MLN model, inference and rule learning
Logistics

• Project proposal due Sept. 26th, instruction of the proposal report will be forthcoming
• Group size min = 2, max = 4, medium = 3
• Computing resources: AWS credits and department/UF computing facilities
• Example project ideas will be posted related to graph analysis, text/image retrieval and knowledge base construction
• AWS, Map-Reduce, SQL/Hive, noSQL, Tableau..
  → coursera, UF CISE intro to DS spring 14
Probabilistic graphical models

Probabilistic models

Graphical models

Directed
(Bayesian networks)

Undirected
(Markov Random fields - MRFs)
Bayesian Networks

- Directed Acyclic Graph (DAG)
  - Nodes are random variables
  - Edges indicate causal influences
  - Conditional independence: each RV is conditionally independent of other nodes given its parent nodes
Conditional Probability Tables

- Each node has a **conditional probability table (CPT)** that gives the probability of each of its values given every possible combination of values for its parents (conditioning case).
  - Roots (sources) of the DAG that have no parents are given prior probabilities.

<table>
<thead>
<tr>
<th>Burglary (P(B))</th>
<th>Earthquake (P(E))</th>
</tr>
</thead>
<tbody>
<tr>
<td>.001</td>
<td>.002</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Alarm (P(A))</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>T</td>
</tr>
<tr>
<td>T</td>
</tr>
<tr>
<td>F</td>
</tr>
<tr>
<td>F</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>JohnCalls (P(J))</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>T</td>
</tr>
<tr>
<td>F</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MaryCalls (P(M))</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>T</td>
</tr>
<tr>
<td>F</td>
</tr>
</tbody>
</table>
Joint Distributions for Bayes Nets

• A Bayesian Network implicitly defines a joint distribution.

\[ P(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid \text{Parents}(X_i)) \]

• Example

\[ P(J \land M \land A \land \neg B \land \neg E) \]
\[ = P(J \mid A)P(M \mid A)P(A \mid \neg B \land \neg E)P(\neg B)P(\neg E) \]
\[ = 0.9 \times 0.7 \times 0.001 \times 0.999 \times 0.998 = 0.00062 \]
Naïve Bayes as a Bayes Net

• Naïve Bayes is a simple Bayes Net

\[
Y \xrightarrow{} X_1 \xrightarrow{} X_2 \xrightarrow{} \cdots \xrightarrow{} X_n
\]

• Priors $P(Y)$ and conditionals $P(X_i|Y)$ for Naïve Bayes provide CPTs for the network.
Markov Networks

- **Undirected graph** over a set of random variables, where an edge represents a dependency.
- The **Markov blanket** of a node, $X$, in a Markov Net is the set of its neighbors in the graph (nodes that have an edge connecting to $X$).
- Every node in a Markov Net is **conditionally independent** of every other node given its Markov blanket.
Distribution for a Markov Network

• The distribution of a Markov net is most compactly described in terms of a set of potential functions (a.k.a. factors, compatibility functions), $\phi_k$, for each clique, $k$, in the graph.

• For each joint assignment of values to the variables in clique $k$, $\phi_k$ assigns a non-negative real value that represents the compatibility of these values.

• The joint distribution of a Markov network is then defined by:

$$P(x_1, x_2, \ldots, x_n) = \frac{1}{Z} \prod_k \phi_k(x_{\{k\}})$$

Where $x_{\{k\}}$ represents the joint assignment of the variables in clique $k$, and $Z$ is a normalizing constant that makes a joint distribution that sums to 1.

$$Z = \sum_x \prod_k \phi_k(x_{\{k\}})$$
Sample Markov Network

<table>
<thead>
<tr>
<th>B</th>
<th>A</th>
<th>( \phi_1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>100</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>1</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>200</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>E</th>
<th>A</th>
<th>( \phi_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>50</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>10</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>1</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>200</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>J</th>
<th>A</th>
<th>( \phi_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>75</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>10</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>1</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>200</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>M</th>
<th>A</th>
<th>( \phi_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>50</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>10</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>200</td>
</tr>
</tbody>
</table>
Logistic Regression as a Markov Net

• Logistic regression is a simple Markov Net
  – potential functions $\phi_k(x\{k\})$ instead of conditional probability tables $P(X_i|Y)$

• But only models the **conditional distribution**, $P(Y|X)$ and **not the full joint** $P(X,Y)$

• Same as a discriminatively trained naïve Bayes.
Generative vs. Discriminative

• **Generative models** and are *not* directly designed to maximize the performance of sequence labeling. They model the **joint distribution** $P(O,Q)$.

• Generative NLP models are trained to have an accurate probabilistic model of the underlying language, and not all aspects of this model benefit the sequence labeling task.

• **Discriminative models** (CRFs) are specifically designed and trained to maximize performance of labeling, which leads to **more accurate** results. They model the **conditional distribution** $P(Q | O)$. 
Classification Models

Naïve Bayes

Generative

Conditional

Logistic Regression

Discriminative