Relational Markov Networks

presented by Danny Wyatt

Statistical Relational Learning

CSE574

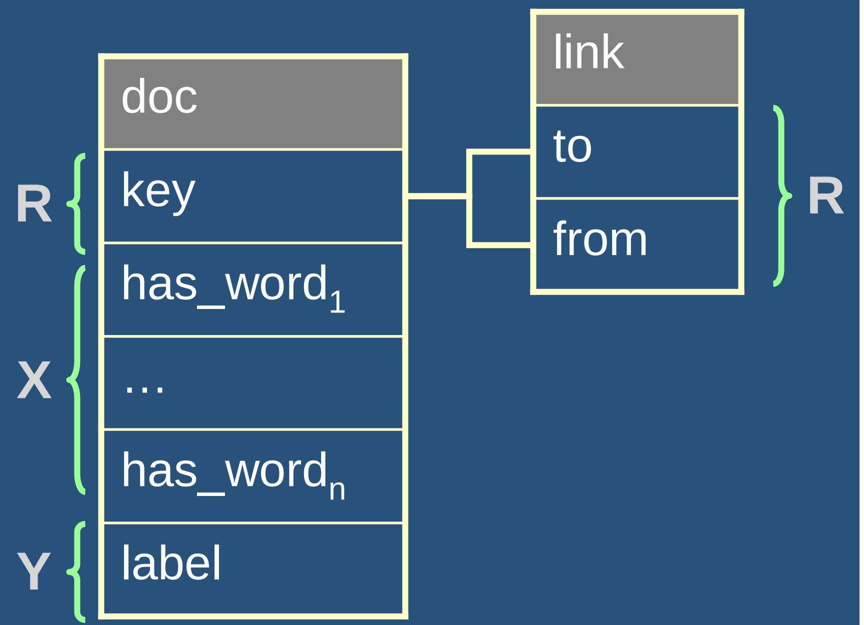
Spring 2005

Origins of RMNs

- Devised for collective classification
  - Classifying an entire set of data at once
  - Taking into account relations between data points
- Markov nets because
  - Undirected, cycles aren’t a problem
  - Easy to learn discriminatively

Classifying Relational Data

- Data fits into a schema, $\mathcal{E}$
  - Tables layout in a database
- Entities with attributes
  - Content attributes **X**
  - Label attributes **Y**
  - Relation attributes **R**
    - Includes a unique key
- Instantiation of a schema, I($\mathcal{E}$)
  - The data in the database



Clique Templates

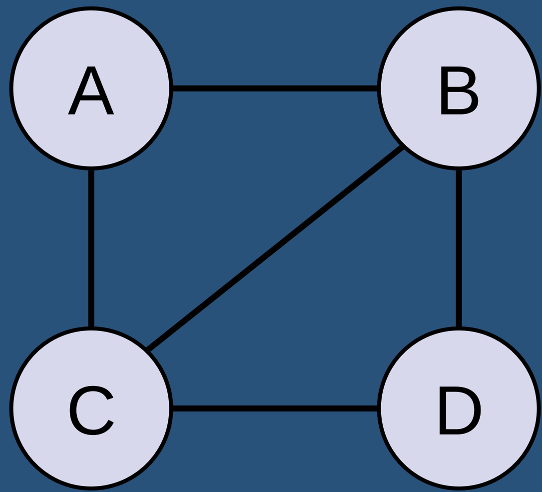
- Query over the data
  - Returns a set of tuples of attributes
- Example: connect labels of pages where one links to the other

```
SELECT d1.label, d2.label
FROM doc d1, doc d2
WHERE link.from = d1
      AND
      link.to = d2;
```

Clique Templates

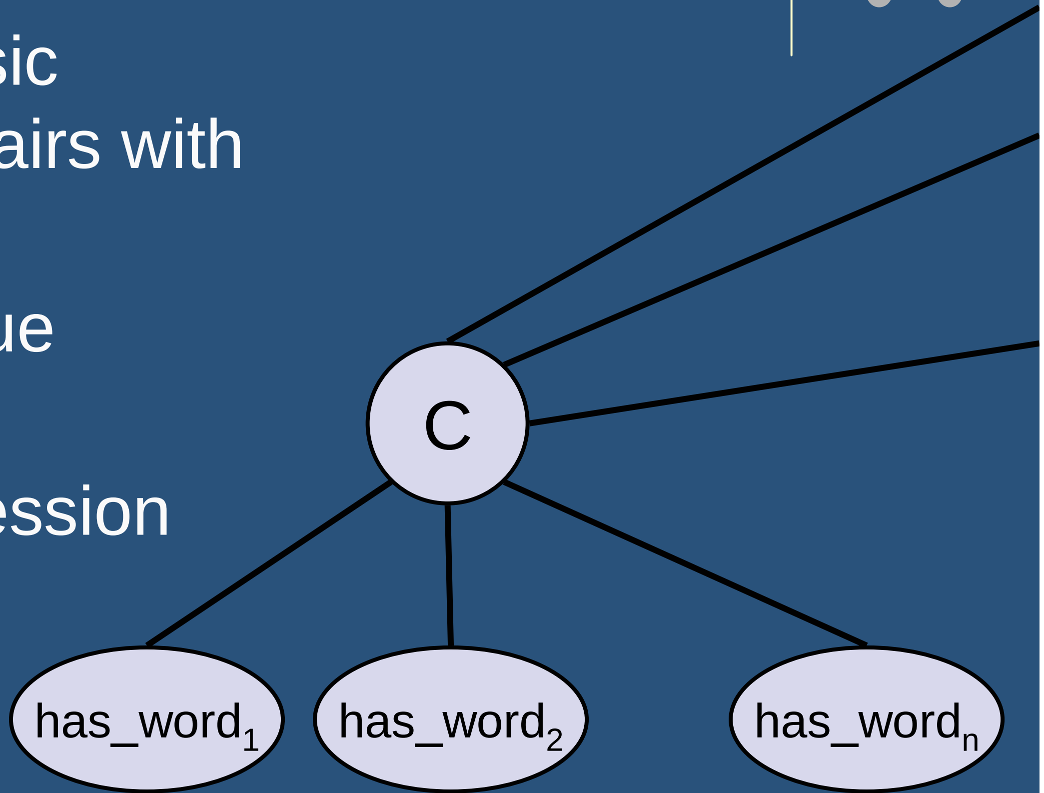
- Clique between all attributes in a tuple
  - Unrolls into entire network
- Example: query results to cliques

d1	D2
A	B
A	C
B	C
B	D
D	C



Clique Templates

- "Non-relational," intrinsic attributes are simple pairs with label
- Still specified with clique templates
- Becomes logistic regression



Clique Templates, Formally

- **F** = {F_i}
  - Set of attributes
  - FROM in SQL
  - Join in relational calculus
- **W(F.R)**
  - Boolean conditions of the form $F_i.R_j = F_k.R_l$
  - WHERE in SQL
  - Selection in relational calculus
- **F.S – F.X – F.Y**
  - Subset of attributes in **F**
  - SELECT in SQL
  - Projection in relational calculus

Why just over relations?

Clique Templates, extended

- In principle, not limited to this formulation
- Expressiveness is limited only by query language
- SQL is equivalent to finite first order logic
  - Extensions provide recursion, fixpoint, and more
- But make sure to consider query complexity
  - Building the cliques could take time

Relational Markov Network

- Set of clique templates, **C**
- Set of potential functions, Φ
  - $\phi_c(V_c) = \exp\{\mathbf{w}_c\ \mathbf{f}_c(V_c)\}$
  - Feature **f** is indicator for state of clique
  - **w** is weight vector
- Defines a conditional distribution over labels of an instantiation

$$P(I.y|I.x, I.r) = \frac{1}{Z(I.x, I.r)} \prod_{C \in \mathbf{C}} \prod_{c \in C(I)} \phi_c(I.x_c, I.y_c)$$

Log Likelihood, Global Formulation

small 'c': specific grounding

$$\log P(I.y | I.x, I.r) =$$

$$\sum_{C \in \mathbf{C}} \sum_{c \in C(I)} \mathbf{w}_C \cdot \mathbf{f}_C(I.x_c, I.y_c) - \log Z(I.x, I.r)$$

big 'C': clique template

$$\sum_{C \in \mathbf{C}} \mathbf{w}_C \cdot \mathbf{f}_C(I.y, I.x, I.r) - \log Z(I.x, I.r)$$

$$= \sum_{c \in C(I)} \mathbf{f}_C(I.x_c, I.y_c)$$

$$\mathbf{w} \cdot \mathbf{f}(I.y, I.x, I.r) - \log Z(I.x, I.r)$$

Learning the weights

- Maximize log likelihood of labels given observations, with training instantiation *I*
- Zero mean Gaussian prior on weights to avoid overfitting

$$L(\mathbf{w}, I) =$$

$$\mathbf{w} \cdot \mathbf{f}(I.y, I.x, I.r) - \log Z(I.x, I.r) - \frac{\mathbf{w} \cdot \mathbf{w}}{2\sigma^2} + C$$

prior

Learning the weights

- Gradient is difference between observed and expected feature counts

$$\nabla L(\mathbf{w}, I) =$$

$$\mathbf{f}(I.y, I.x, I.r) - \mathbf{E}_{\mathbf{w}}[(I.Y, I.x, I.r)] - \frac{\mathbf{w}}{\sigma^2}$$

Learning the weights

$$\mathbf{E}_{\mathbf{w}}[(I.Y, I.x, I.r)] =$$
$$\sum_{I.y'} \mathbf{f}(I.y', I.x, I.r) P_{\mathbf{w}}(I.y' | I.x, I.r)$$

complete labeling of
entire network

- Expectation involves summing over all assignment configurations
  - Does not decompose per instance
  - All labels are correlated

Learning, Inference in practice

- Weight learning
  - gradient descent
- Inference
  - Loopy belief propagation
  - But could use anything in principle

Another extension [Liao05]

- Clique over all attributes in entire *set* returned
- Clique size unknown, must aggregate
  - Feature value is part of query
- Example: pages only link to a small set of other classes

```
SELECT COUNT DISTINCT d2.label
FROM doc d1, doc d2
WHERE link.from = d1
      AND
      link.to = d2;
```

5 Dimensions of SRL

- Probabilistic model
  - Markov nets
- Relational model
  - Relational databases
- Learning
  - Parameters, with gradient descent
  - Not structure
- Inference
  - Any MN inference method
  - …unless structure changes with inference
- Aggregation
  - None in Taskar’s method
  - SQL aggregation in Liao’s

Results [Taskar02]

- WebKB data set
  - Classify web pages as belonging to faculty, student, course, etc.
- RMNs 8% more accurate than logistic regression, on average
  - Up to 15% more on some data
- RMNs 10% more accurate than PRMs
  - Up to 35%
  - Possible benefit of discriminative model

Results [Liao05]

- GPS location information about a person
  - Augmented with place information (restaurants, stores)
  - Label the activities performed
- Extended clique templates work
  - 20% boost in accuracy
- Can learn priors for weights
  - 25% boost in accuracy

Comparison to MLNs

- MLNs
  - Data, rules, and queries all in FOL
  - Never need to know about Markov net
  - Inference can ground partial network
- RMNs
  - Data and rules in SQL
  - Queries over Markov nets
  - Inference grounds full network

Conclusion

- RMNs provide a convenient way of specifying a parameter-tied Markov net
- Can accept user-defined features
  - e.g. continuous values
- Still very close to a Markov net