Graphical models and their Role in Databases

VLDB 2007 Tutorial

Amol Deshpande[1] Sunita Sarawagi[2]

[1]University of Maryland

[2]IIT Bombay

Why a tutorial on graphical models at VLDB?

- VLDB
  - Many DB tasks use probabilistic modeling
    - ⋆ Core: Selectivity estimation, Imprecise databases,
    - ⋆ Appication: Information extraction, Duplicate elimination, sensor networks.
    - ⋆ Data mining: Classification (naive Bayes, logistic), clustering (EM)
  - Probabilistic modeling is simultaneously
    - ⋆ intuitive (low barrier to entry)
    - ⋆ subtle (important to understand well for correctness & efficiency)
- Graphical models
  - Fundamental tools for intuitively and efficiently modeling probabilities
  - Distilled body of knowledge from many fields (let us build upon them, instead of reinveting)

VLDB wants to broaden, GM a fun and useful candidate for broadening

Probabilistic modeling

- Given: several variables: $x_1, \ldots x_n$, n is large.
- Task: build a joint distribution function $\Pr(x_1, \ldots x_n)$
- Goal: Answer several kind of projection queries on the distribution
- Basic premise
  - Explicit joint distribution is dauntingly large
  - Queries are simple aggregates over the joint distribution.

Example: Selectivity estimation in databases

- Variables are columns of a table

Age	Income	Experience	Degree	Location
10 ranges	7 scales	7 scales	3 scales	30 places

- An explicit joint distribution over all columns not tractable: number of combinations: $10 \times 7 \times 7 \times 3 \times 30 = 44100$.
- Queries: Estimate number of people with
  - Income $>$ 200K and Degree="Bachelors",
  - Income $<$ 200K, Degree="PhD" and experience $>$ 10 years.
  - Many, many more.

Alternatives to an explicit joint distribution

- Assume all columns are independent of each other: bad assumption
- Use data to detect pairs of highly correlated column pairs and estimate their pairwise frequencies
  - Many highly correlated pairs
    income $\not\perp\!\!\!\perp$ age, income $\not\perp\!\!\!\perp$ experience, age$\not\perp\!\!\!\perp$experience
  - Ad hoc methods of combining these into a single estimate
- Go beyond pairwise correlations: understand finer dependencies
  - income $\not\perp\!\!\!\perp$ age, but income $\perp\!\!\!\perp$ age $\mid$ experience
  - experience $\perp\!\!\!\perp$ degree, but experience $\not\perp\!\!\!\perp$ degree $\mid$ income

Graphical models make explicit an efficient joint distribution from these independencies

Graphical models

Model joint distribution over **several** variables as a product of smaller factors that is

1. *Intuitive* to represent and visualize
   - Graph: represent structure of dependencies
   - Potentials over subsets: quantify the dependencies
2. *Efficient* to query
   - given values of any variable subset, reason about probability distribution of others.
   - many efficient exact and approximate inference algorithms

Graphical models $=$ graph theory $+$ probability theory.

Graphical models in use

- Roots in statistical physics for modeling interacting atoms in gas and solids [1900]
- Early usage in genetics for modeling properties of species [1920]
- AI: expert systems (1970s-80s)
- Now many new applications:
  - Error Correcting Codes: Turbo codes, impressive success story (1990s)
  - Robotics and Vision: image denoising, robot navigation.
  - Text mining: information extraction, duplicate elimination, hypertext classification, help systems
  - Bio-informatics: Secondary structure prediction, Gene discovery
  - Data mining: probabilistic classification and clustering.

Overall plan

- Fundamentals
  - Representation
  - Exact inference
- Applications
  - Selectivity estimation
  - Probabilistic databases
- Applications: Sensor data management
- Fundamentals
  1. Learning a graphical model
  2. Conditional Random Fields
- Applications
  - Information extraction
  - Duplicate elimination

Part I: Fundamentals of Graphical Models

Part I: Outline

1. Representation
   - Directed graphical models: Bayesian networks
   - Undirected graphical models
2. Inference Queries
   - Exact inference on chains
   - Exact inference on general graphs
3. Constructing a graphical model
   - Graph Structure
   - Parameters in Potentials
4. Approximate inference
   - Generalized belief propagation
   - Sampling: Gibbs, Particle filters

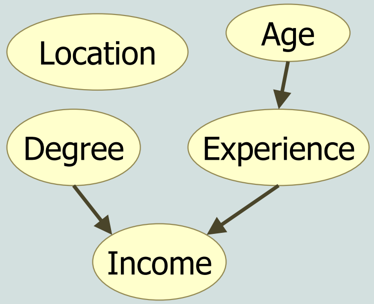
Representation

Structure of a graphical model: Graph + Potential

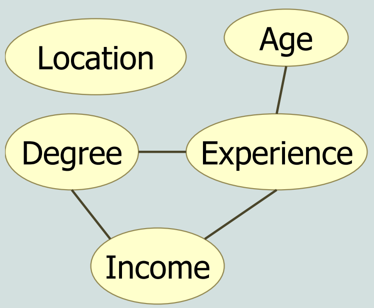
Graph

- Nodes: variables $\mathbf{x} = x_1, \ldots x_n$
  - Continuous: Sensor temperatures, income
  - Discrete: Degree (one of Bachelors, Masters, PhD), Levels of age
- Edges: direct interaction
  - Directed edges: Bayesian networks
  - Undirected edges: Markov Random fields

Directed



Undirected



Representation

Potentials: $\psi_c(\mathbf{x}_c)$

- Scores for assignment of values to subsets c of directly interacting variables.
- Which subsets? What do the potentials mean?
  - Different for directed and undirected graphs

Probability

Factorizes as product of potentials

$$\Pr(\mathbf{x} = x_1, \ldots x_n) \propto \prod \psi_S(\mathbf{x}_S)$$

Directed graphical models: Bayesian networks

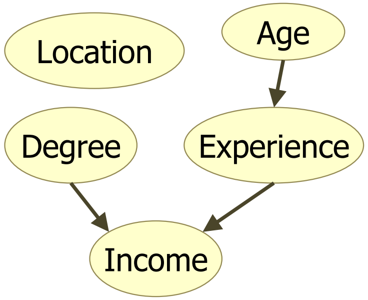
- Graph G: directed acyclic
  - Parents of a node: $\text{Pa}(x_i)$ = set of nodes in G pointing to x_i
- Potentials: defined at each node in terms of its parents.

$$\psi_i(x_i, \text{Pa}(x_i)) = \Pr(x_i|\text{Pa}(x_i))$$

- Probability distribution

$$\Pr(x_1 \ldots x_n) = \prod_{i=1}^{n} \Pr(x_i|pa(x_i))$$

Example of a directed graph



$\psi_1(L) = \Pr(L)$

NY	CA	London	Other
0.2	0.3	0.1	0.4

$\psi_2(A) = \Pr(A)$

20–30	30–45	$>$ 45
0.3	0.4	0.3

or, a Guassian distribution $(\mu, \sigma) = (35, 10)$

$\psi_2(E, A) = \Pr(E|A)$

	0–10	10–15	$>$ 15
20–30	0.9	0.1	0
30–45	0.4	0.5	0.1
$>$ 45	0.1	0.1	0.8

$\psi_2(I, E, D) = \Pr(I|D, A)$

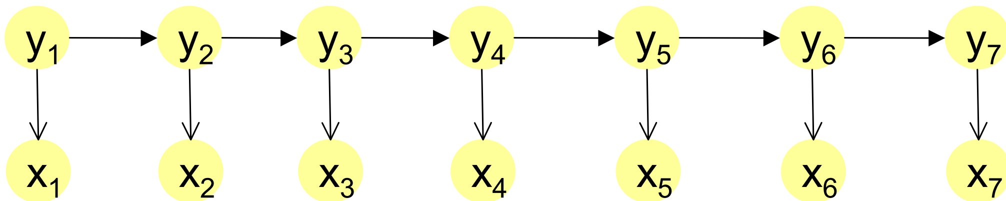
3 dimensional table, or a histogram approximation.

Probability distribution

$\Pr(\mathbf{x} = L, D, I, A, E) = \Pr(L)\Pr(D)\Pr(A)\Pr(E|A)\Pr(I|D, E)$

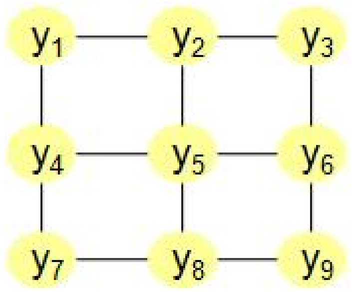
Popular Bayesian networks

- Hidden Markov Models: speech recognition, information extraction



  - State variables: discrete phoneme, entity tag
  - Observation variables: continuous (speech waveform), discrete (Word)
- Kalman Filters: State variables: continuous
  - Discussed later
- PRMs: Probabilistic relational networks:
  - An important relevant class for relational data
  - Discussed later
- QMR (Quick Medical Reference) system

Undirected graphical models



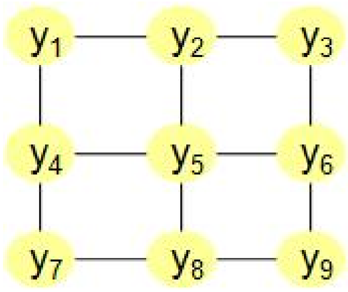
- Graph G: arbitrary undirected graph
- Useful when variables interact symmetrically, no natural parent-child relationship
- Example: labeling pixels of an image.
- Potentials defined on arbitrary subcliques C of G. Popular choices:
  - Node potentials
  - Edge potentials

Probability distribution

$$\Pr(\mathbf{x} = y_1 \dots y_n) = \frac{1}{Z} \prod_{C \in G} \psi_C(\mathbf{y}_C)$$

where $Z = \sum_{\mathbf{y}'} \prod_{C \in G} \psi_C(\mathbf{y}'_C)$

Example



$y_i = 1$ (part of foreground), 0 otherwise.

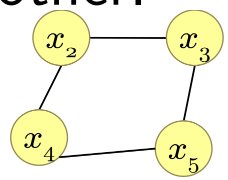
- Node potentials
  - $\psi_1(0) = 4,\ \psi_1(1) = 1$
  - $\psi_2(0) = 2,\ \psi_2(1) = 3$
  - ....
  - $\psi_9(0) = 1,\ \psi_9(1) = 1$
- Edge potentials: Same for all edges
  - $\psi(0,0) = 5,\ \psi(1,1) = 5,\ \psi(1,0) = 1, \psi(0,1) = 1$
- Probability: $\Pr(y_1 \ldots y_9) \propto \prod_{k=1}^{9} \psi_k(y_k) \prod_{(i,j)\in E(G)} \psi(y_i, y_j)$

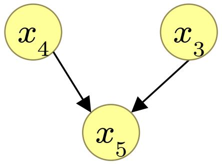
Popular undirected graphical models

- Interacting atoms in gas and solids [1900]
- Markov Random Fields in vision for image segmentation
- Conditional Random Fields for information extraction

Comparing directed and undirected graphs

- Some distributions can only be expressed in one and not the other.





- Potentials
  - Directed: conditional probabilities, more intuitive
  - Undirected: arbitrary scores, easy to set.
- Dependence structure
  - Directed: Complicated d-separation test
  - Undirected: Graph separation: $A \perp\!\!\!\perp B \mid C$ iff C separates A and B in G.
- Often application makes the choice clear.
  - Directed: Causality
  - Undirected: Symmetric interactions.

Part I: Outline

1. Representation
   - Directed graphical models: Bayesian networks
   - Undirected graphical models
2. Inference Queries
   - Exact inference on chains
   - Exact inference on general graphs
3. Constructing a graphical model
   - Graph Structure
   - Parameters in Potentials
4. Approximate inference
   - Generalized belief propagation
   - Sampling: Gibbs, Particle filters

Inference queries

1. *Marginal probability queries over a small subset of variables:*
   - Find Pr(Income='High & Degree='PhD')
   - Find Pr(pixel $y_9 = 1$)

$$\Pr(x_1) = \sum_{x_2 \dots x_n} \Pr(x_1 \dots x_n)$$

2. *Most likely labels of remaining variables:* (MAP queries)
   - Find most likely entity labels of all words in a sentence
   - Find likely temperature at sensors in a room

$$\mathbf{x}^* = \operatorname{argmax}_{x_1 \dots x_n} \Pr(x_1 \dots x_n)$$

Exact inference on chains

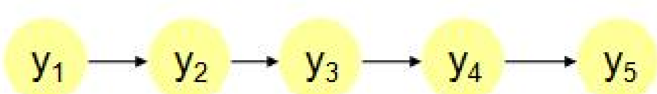
- Given,
  - Graph 

  - Potentials: $\psi_i(y_i, y_{i+1})$
  - $Pr(y_1, \ldots y_n) = \prod_i \psi_i(y_i, y_{i+1})$
- Find, $\Pr(y_i)$ for any i, say $\Pr(y_5 = 1)$
  - Exact method: $\Pr(y_5 = 1) = \sum_{y_1,\ldots y_4} Pr(y_1, \ldots y_4, 1)$ requires exponential number of summations.
  - A more efficient alternative...

Exact inference on chains

$$
\begin{aligned}
\Pr(y_5 = 1) &= \sum_{y_1,\ldots y_4} Pr(y_1, \ldots y_4, 1) \\
&= \sum_{y_1}\sum_{y_2}\sum_{y_3}\sum_{y_4} \psi_1(y_1, y_2)\psi_2(y_2, y_3)\psi_3(y_3, y_4)\psi_4(y_4, 1) \\
&= \sum_{y_1}\sum_{y_2} \psi_1(y_1, y_2) \sum_{y_3} \psi_2(y_2, y_3) \sum_{y_4} \psi_3(y_3, y_4)\psi_4(y_4, 1) \\
&= \sum_{y_1}\sum_{y_2} \psi_1(y_1, y_2) \sum_{y_3} \psi_2(y_2, y_3) B_3(y_3) \\
&= \sum_{y_1}\sum_{y_2} \psi_1(y_1, y_2) B_2(y_2) \\
&= \sum_{y_1} B_1(y_1)
\end{aligned}
$$

An alternative view: flow of beliefs $B_i(.)$ from node $i+1$ to node i



Adding evidence

Given fixed values of a subset of variables $\mathbf{x}_e$ (evidence), find the

1. *Marginal probability queries over a small subset of variables:*
   - Find Pr(Income='High | Degree='PhD')

$$\Pr(x_1) = \sum_{x_2 \ldots x_m} \Pr(x_1 \ldots x_n | \mathbf{x}_e)$$

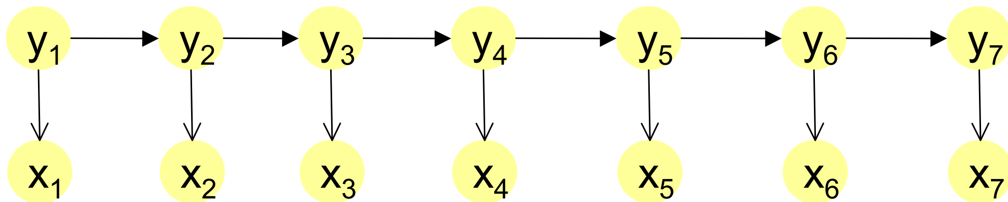
2. *Most likely labels of remaining variables:* (MAP queries)
   - Find likely temperature at sensors in a room given readings from a subset of them

$$\mathbf{x}^* = \text{argmax}_{x_1 \ldots x_m} \Pr(x_1 \ldots x_n | \mathbf{x}_e)$$

Easy to add evidence, just change the potential.

Inference in HMMs

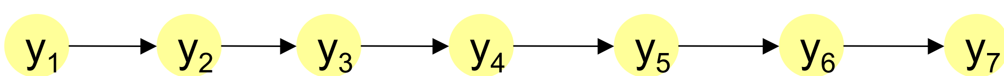
- Given,
  - Graph



  - Potentials: $\Pr(y_i|y_{i-1}), \Pr(x_i|y_i)$
  - Evidence variables: $\mathbf{x} = x_1 \dots x_n = o_1 \dots o_n$.
- Find most likely values of the hidden state variables.
  $\mathbf{y} = y_1 \dots y_n$

$$\text{argmax}_{\mathbf{y}} \Pr(\mathbf{y}|\mathbf{x} = \mathbf{o})$$

- Define $\psi_i(y_{i-1}, y_i) = \Pr(y_i|y_{i-1}) \Pr(x_i = o_i|y_i)$
- Reduced graph only a single chain of y nodes.



- Algorithm same as earlier, just replace "Sum" with "Max"

This is the well-known Viterbi algorithm

Exact inference on trees

- Basic steps for marginal and MAP queries.
  - Perform sum/max over leaf node potential and send resulting "belief" to parent.
  - Each internal node, on getting beliefs from its children
    1. Multiplies incoming beliefs with its own potentials
    2. Performs sum/max on the result
    3. Sends resulting "belief" factor to parent.
- Root has the answer.

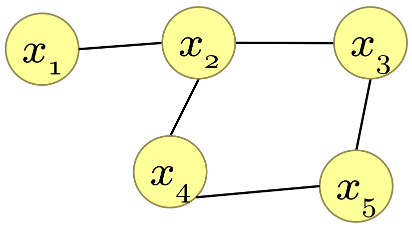
Linear in the number of nodes in the graph

Junction tree algorithms

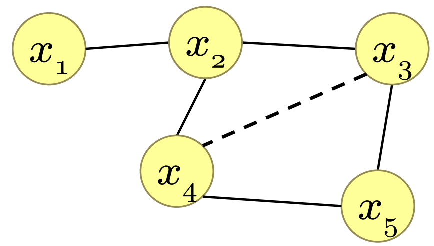
- An optimal general-purpose algorithm for exact marginal/MAP queries
- Simultaneous computation of many queries
- Efficient data structures
- Complexity: $O(m^w N)$ $w=$ size of the largest clique in (triangulated) graph, $m =$ number of values of each discrete variable in the clique. $\rightarrow$ linear for trees.
- Basis for many approximate algorithms.
- Many popular inference algorithms special cases of junction trees
  - Viterbi algorithm of HMMs
  - Forward-backward algorithm of Kalman filters

Creating a junction tree from a graphical model

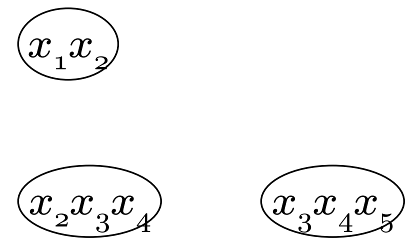
1. Starting graph



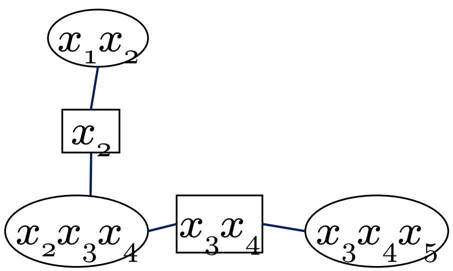
2. Triangulate graph



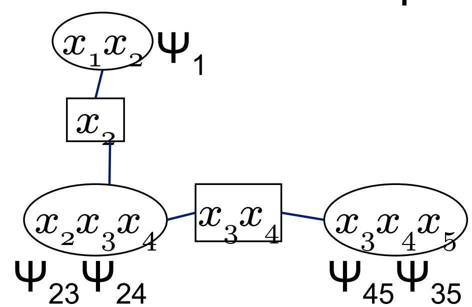
3. Create clique nodes



4. Create tree edges such that variables connected.



5) Assign potentials to exactly one subsumed clique node.



Belief propagation on junction trees

- Each node c
  - sends *belief* $B_{c \to c'}(.)$ to each of its neighbors c'
    - once it has beliefs from every other neighbor $N(c) - \{c'\}$.
  - $B_{c \to c'}(.)$ = belief that clique c has about the distribution of labels to common variables $s = c \cap c'$

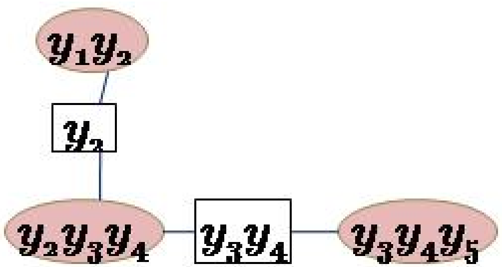
$$B_{c \to c'}(\mathbf{x}_s) = \sum_{\mathbf{x}_{c-s}} \psi_c(\mathbf{x}_c) \prod_{d \in N(c) - \{c'\}} B_{d \to c}(\mathbf{x}_{d \cap c})$$

Replace "sum" with "max" for MAP queries.

Compute marginal probability of any variable x_i as

1. c = clique in JT containing x_i
2. $\Pr(x_i) \propto \sum_{\mathbf{x}_{c-x_i}} \psi_c(\mathbf{x}_c) \prod_{d \in N(c)} B_{d \to c}(\mathbf{x}_{d \cap c})$

Example



$$\psi_{234}(\mathbf{y}_{234}) = \psi_{23}(\mathbf{y}_{23})\psi_{34}(\mathbf{y}_{34})$$
$$\psi_{345}(\mathbf{y}_{345}) = \psi_{35}(\mathbf{y}_{35})\psi_{45}(\mathbf{y}_{45})$$
$$\psi_{234}(\mathbf{y}_{12}) = \psi_{12}(\mathbf{y}_{12})$$

1. Clique "12" sends belief $B_{12\to 234}(y_2) = \sum_{y_1} \psi_{12}(\mathbf{y}_{12})$ to its only neighbor.
2. Clique "345" sends belief $B_{345\to 234}(\mathbf{y}_{34}) = \sum_{y_5} \psi_{234}(\mathbf{y}_{345})$ to "234"
3. Clique "234" sends belief $B_{234\to 345}(\mathbf{y}_{34}) = \sum_{y_2} \psi_{234}(\mathbf{y}_{234})B_{12\to 234}(y_2)$ to "345"
4. Clique "234" sends belief $B_{234\to 12}(y_2) = \sum_{y_4} \psi_{234}(\mathbf{y}_{234})B_{345\to 234}(\mathbf{y}_{34})$ to "12"

$\Pr(y_1) \propto \sum_{y_2} \psi_{12}(\mathbf{y}_{12})B_{234\to 12}(y_2)$

Part I: Outline

1. Representation
   - Directed graphical models: Bayesian networks
   - Undirected graphical models
2. Inference Queries
   - Exact inference on chains
   - Exact inference on general graphs
3. Constructing a graphical model
   - Graph Structure
   - Parameters in Potentials
4. Approximate inference
   - Generalized belief propagation
   - Sampling: Gibbs, Particle filters

Graph Structure

1. Manual: Designed by domain expert
   - Used in applications where dependency structure is well-understood
   - Example: QMR systems, Kalman filters, Vision (Grids), HMM for speech recognition and IE.
2. Learnt: from examples
   - NP hard to find the optimal structure.
   - Widely researched, mostly posed as a branch and bound search problem.
   - Useful in dyanmic situations
   - Example: Selectivity estimation over attributes of arbitrary tables.

Parameters in Potentials

1. Manual: Provided by domain expert
   - Used in infrequently constructured graphs, example QMR systems
   - Also where potentials are an easy function of the attributes of connected graphs, example: vision networks.
2. Learnt: from examples
   - More popular since difficult for humans to assign numeric values
   - Many variants of parameterizing potentials.
     1. Each potential entry a parameter, example, HMMs
     2. Potentials: combination of shared parameters and data attributes: example, CRFs. (Discussed in later with extraction)

Part I: Outline

1. Representation
   - Directed graphical models: Bayesian networks
   - Undirected graphical models
2. Inference Queries
   - Exact inference on chains
   - Exact inference on general graphs
3. Constructing a graphical model
   - Graph Structure
   - Parameters in Potentials
4. Approximate inference
   - Generalized belief propagation
   - Sampling: Gibbs, Particle filters

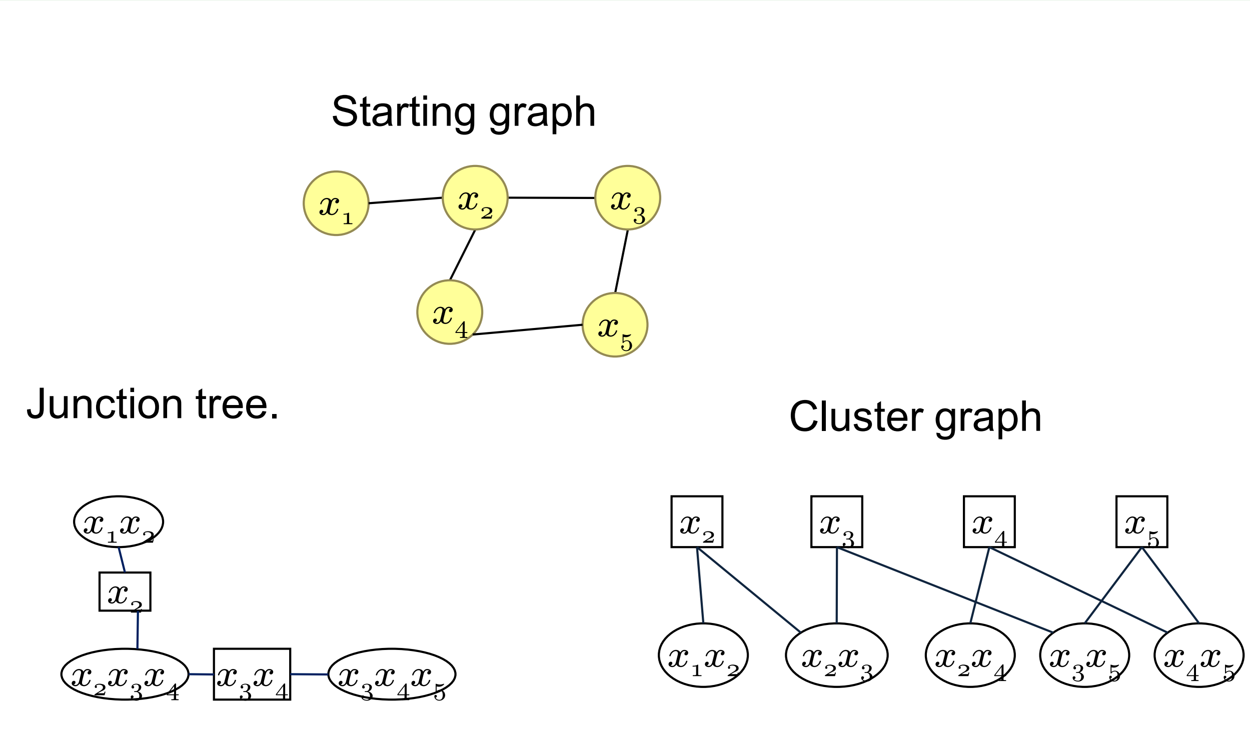
Why approximate inference

- Exact inference is NP hard. Complexity: $O(w^m)$
  - w= tree width = size of the largest clique in (triangulated) graph-1,
  - m = number of values of each discrete variable in the clique.
- Many real-life graphs produce large cliques on triangulation
  - A $n \times n$ grid has a tree width of n
  - A Kalman filter on K parallel state variables influencing a common observation variable, has a tree width of size $K + 1$

Generalized belief propagation

- Approximate junction tree with a cluster graph where
  1. Nodes = arbitrary clusters, not cliques in triangulated graph. Only ensure all potentials subsumed.
  2. Separator nodes on edges = *subset* of intersecting variables.

Example cluster graph



Belief propagation in cluster graphs

- Graph can have loops, tree-based two-phase method not applicable.
- Many variants on scheduling order of propagating beliefs.
  - Simple loopy belief propagation [Pea88]
  - Tree-reweighted message passing [Kol04]
  - Residual belief probagation [EMK06]
- Most have no guarantees of convergence
- Works well in practice, default method of choice.
  - Success story: Error correction using Turbo code

MCMC (Gibbs) sampling

- Useful when all else failes, guaranteed to converge to the optimal over infinite number of samples.
- Basic premise: easy to compute conditional probability $\Pr(x_i | \text{fixed values of remaining variables})$

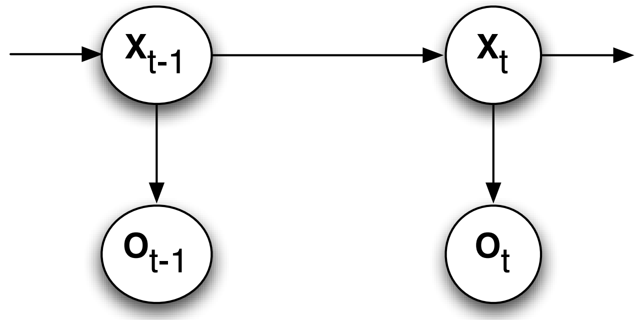
Algorithm

- Start with some initial assignment, say $\mathbf{x}^1 = [x_1, \ldots, x_n] = [0, \ldots, 0]$
- For several iterations
  - For each variable x_i
    Get a new sample $\mathbf{x}^{t+1}$ by replacing value of x_i with a new value sampled according to probability $\Pr(x_i | x_1^t, \ldots x_{i-1}^t, x_{i+1}^t, \ldots, x_n^t)$

Others

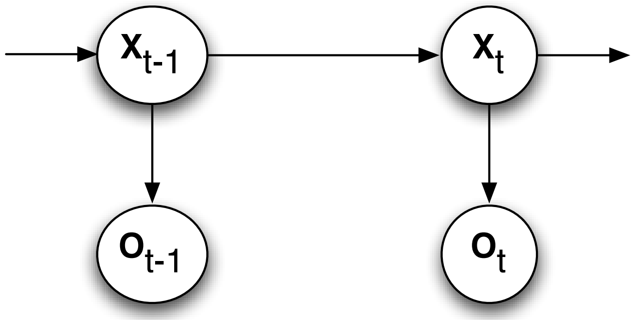
- Combinatorial algorithms for MAP [BVZ01, DTEK07, GDS07]
- Greedy algorithms: relaxation labeling
- Variational methods
- LP and QP based approaches

Inference Task in DBNs



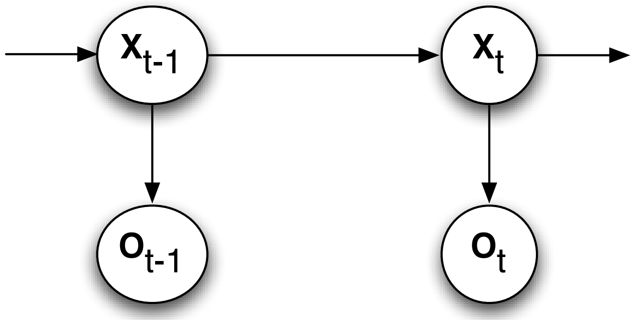
- Simplied representation of a dynamic Bayesian network
  - Hidden *state* variables: $\mathbf{x}$; Observed variables: $\mathbf{o}$
  - Assumed to be vector valued
- Given:
  - Prior on the initial state: $p(\mathbf{x}_0)$
  - How state evolves: $p(\mathbf{x}_t|\mathbf{x}_{t-1})$
  - How obsevations depend on state: $p(\mathbf{o}_t|\mathbf{x}_t)$
- Estimate the *state at time t* given *observations till time t*
  - The posterior distribution: $p(\mathbf{x}_t|\mathbf{o}_{1:t})$

Alternative Inference Tasks in DBNs



- Estimate the most likely sequence of states (for discrete $\mathbf{x}$)
  - $\text{argmax}_{\mathbf{x}_{1:t}} p(\mathbf{o}_{1:t} | \mathbf{x}_{1:t})$ (*Cf. Viterbi Algorithm*)
- Estimate the distribution of all states till time t
  - $p(\mathbf{x}_{1:t} | \mathbf{o}_{1:t})$
- Estimate the state at time t given measurements till time $t + l$ (*fixed-lag smoothing*)
  - $p(\mathbf{x}_t | \mathbf{o}_{1:t+l})$
  - Why ? Belief about the state at time t may change drastically given future observations

Exact Inference in DBNs

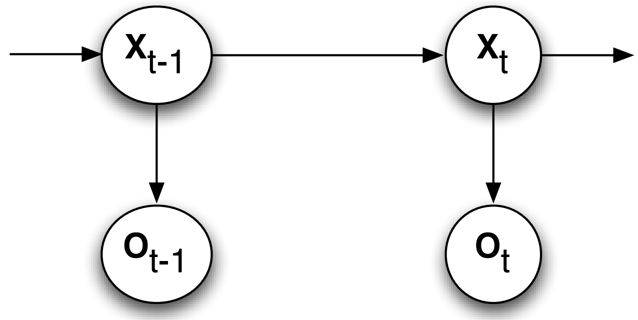


- Easy to write down
  - Using Bayes rule and Chain rule, we get:

$$p(\mathbf{x}_t|\mathbf{o}_{1:t}) = \frac{p(\mathbf{o}_t|\mathbf{x}_t)\int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{o}_{1:t-1})d\mathbf{x}_{t-1}}{p(\mathbf{o}_t|\mathbf{o}_{1:t-1})}$$

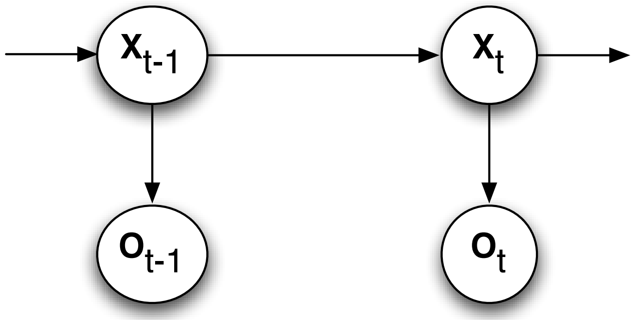
  - Where:
    - ★ $p(\mathbf{o}_t|\mathbf{x}_t)$ and $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ are known model parameters
    - ★ $p(\mathbf{x}_{t-1}|\mathbf{o}_{1:t-1})$ is available from the previous time
    - ★ $p(\mathbf{o}_t|\mathbf{o}_{1:t-1}) = \int p(\mathbf{o}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{o}_{1:t-1})d\mathbf{x}_t$ is a normalization constant (so may not need to be evaluated)

Exact Inference in DBNs



- However, can solve exactly in very few cases:
  - Kalman filters: if the system is linear Gaussian
    - If $p(\mathbf{x}_{t-1}|\mathbf{o}_{1:t-1})$ is Gaussian and the system is linear Gaussian, $p(\mathbf{x}_t|\mathbf{o}_{1:t})$ is Gaussian
    - Very efficient
    - Backward smoothing also easily doable
  - Grid-based method: if the state space is discrete and finite
    - Can compute the integral as a sum exactly

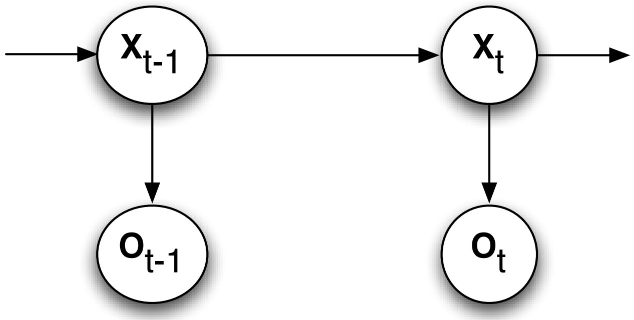
Approximate Inference in DBNs



- Extended Kalman Filter
  - Approximate the process as a linear Gaussian system
  - Will fail if the posterior density not close to a Gaussian (e.g. if it is bimodal or heavily skewed)

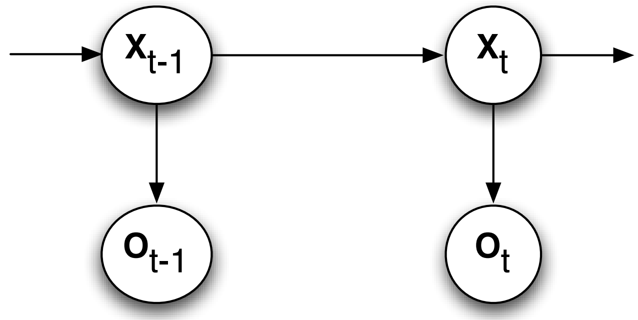
- Approximate Grid-based methods
  - Discretize the continuous state space using a grid
  - Need sufficiently dense grid for good approximation
  - Suffers from "curse of dimensionality"

Approximate Inference in DBNs: Particle Filters



- Approximate the state using a set of weighted samples, called *particles*
- At time $t-1$, approximate $p(x_{t-1}|o_{1:t-1})$ using n particles:
  - $\{\mathbf{x}_{t-1}^1, w_{t-1}^1\}, \{\mathbf{x}_{t-1}^2, w_{t-1}^2\}, \cdots, \{\mathbf{x}_{t-1}^n, w_{t-1}^n\}$
- Can estimate any statistic using these particles
  - e.g. $E(\mathbf{x}_{t-1}|\mathbf{o}_{1:t-1}) \approx \Sigma_{i=1}^n w_{t-1}^i \mathbf{x}_{t-1}^i$
- Inference Task: Generate a set of particles corresponding to $p(\mathbf{x}_t|\mathbf{o}_{1:t})$ given $\mathbf{o}_t$

Approximate Inference in DBNs: Particle Filters

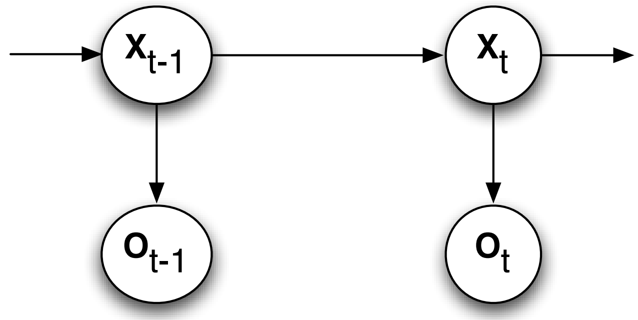


- Generate one sample each from: $p(\mathbf{x}_t | \mathbf{x}^i_{t-1}, \mathbf{o}_t)$
- Assign weights as:

$$w^i_t \propto w^i_{t-1} p(\mathbf{o}_t | \mathbf{x}^i_{t-1}) = w^i_{t-1} \int p(\mathbf{o}_t | \mathbf{x}'_t) p(\mathbf{x}'_t | \mathbf{x}^i_{t-1}) d\mathbf{x}'_t$$

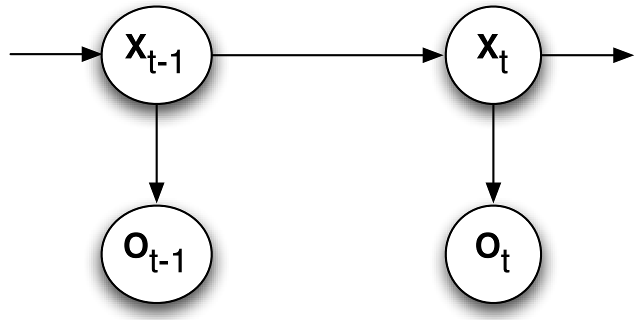
- Problems:
  - Requires sampling from $p(\mathbf{x}_t | ...)$ and computing $p(\mathbf{o}_t | ...)$
  - Requires evaluating complex integrals
- Can solve in very few cases:
  - $\mathbf{x}_t$ is discrete, *or*
  - $p(\mathbf{x}_t | \mathbf{x}^i_{t-1}, \mathbf{o}_t)$ is Gaussian (evolution can still be non-linear)

Approximate Inference in DBNs: Particle Filters



- Must use *importance sampling*
  - Use an *importance density* $q()$ to generate samples from
  - ...that closely approximates the true density $p()$
  - No magic bullet for choosing $q()$
- Degeneracy issues
  - After a while, a single particle has all the weight
  - Need to resample periodically

Approximate Inference in DBNs: Particle Filters



- Many other extensions/variations have been considered
  - A lot more art than science at this point

- For an approachable introduction, see "A Tutorial on Particle Filters for On-line Nonlinear/Non-Gaussian Bayesian Tracking"; Arulampalam et al.; IEEE Trans. Signal Processing; 2002
  - Our discussion heavily borrows from it

More on graphical models

- Koller and Friedman book (Structured Probabilistic Models) not published yet but you could request authors for a draft.
- Kevin Murphy's brief online introduction (`http://www.cs.ubc.ca/~murphyk/Bayes/bnintro.html`)
- Graphical models. M. I. Jordan. Statistical Science (Special Issue on Bayesian Statistics), 19, 140-155, 2004. (`http://www.cs.berkeley.edu/~jordan/papers/statsci.ps.gz`)
- Other text books:
  - R. G. Cowell, A. P. Dawid, S. L. Lauritzen and D. J. Spiegelhalter. "Probabilistic Networks and Expert Systems". Springer-Verlag. 1999.
  - J. Pearl. "Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference." Morgan Kaufmann. 1988.
  - Graphical models by Lauritzen, Oxford science publications F. V. Jensen. "Bayesian Networks and Decision Graphs". Springer. 2001.

Yuri Boykov, Olga Veksler, and Ramin Zabih.
Fast approximate energy minimization via graph cuts.
IEEE Trans. Pattern Anal. Mach. Intell., 23(11):1222–1239, 2001.

J. Duchi, D. Tarlow, G. Elidan, and D. Koller.
Using combinatorial optimization within max-product belief propagation.
In *Advances in Neural Information Processing Systems (NIPS 2006)*, 2007.

G. Elidan, I. McGraw, and D. Koller.
Residual belief propagation: Informed scheduling for asynchronous message passing.
In *Proceedings of the Twenty-second Conference on Uncertainty in AI (UAI)*, Boston, Massachussetts, July 2006.

Rahul Gupta, Ajit A. Diwan, and Sunita Sarawagi.
Efficient inference with cardinality-based clique potentials.
In *Proceedings of the 24th International Conference on Machine Learning (ICML), USA*, 2007.

Vladimir Kolmogorov.
Convergent tree-reweighted message passing for energy minimization.
Technical Report MSR-TR-2004-90, Microsoft Research (MSR), September 2004.

Judea Pearl.
Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.
Morgan Kaufmann, 1988.

Part II: Applications

Part II: Outline

- Selectivity Estimation and Query Optimization
- Probabilistic Relational Models
- Probabilistic Databases
- Sensor/Stream Data Management
- References

Selectivity Estimation

- Estimating the intermediate result sizes that may be generated during query processing
  - Equivalently, selectivities of predicates over tables
- Key to obtaining good plans during optimization

Customer

SSN	..	Income	..	Homeowner?
..	..	100000	..	Yes
..	..	11000	..	Yes

Purchases

SSN	Store	..	Amount

Single-table predicates:

income > 90000 and homeowner = yes
(on customer)

Multi-table predicates:

p.ssn = c.ssn and c.homeowner = "no"
and p.amount > 10000
(over Customer c and Purchases p)

Selectivity Estimation

- Optimizers make several independent assumptions
- *Attribute value independence* assumption
  - Attributes assumed to be independently distributed
  - Rarely true in practice

Customer

SSN	..	Income	..	Homeowner?
..	..	100000	..	Yes
..	..	11000	..	Yes
..	..	50000	..	No
..	..	30000	..	No
..	..	200000	..	Yes

Estimate

p(income > 90000 and homeowner = yes)

as

*p(income > 900000) * p(homeowner = yes)*

Can result in severe underestimation

In reality:

p(income > 900000, homeowner = yes)

≈ p(homeowner = yes)

Selectivity Estimation

- *Join uniformity* assumption
  - Tuples from one relation assumed equally likely to join with tuples from other relation
  - Real datasets exhibit large skews

Customer

SSN	..	Income	..	Homeowner?
..	..	100000	..	Yes
..	..	11000	..	Yes
..	..	50000	..	No
..	..	30000	..	No
..	..	200000	..	Yes

Purchases

SSN	Store	..	Amount

Selectivity Estimation

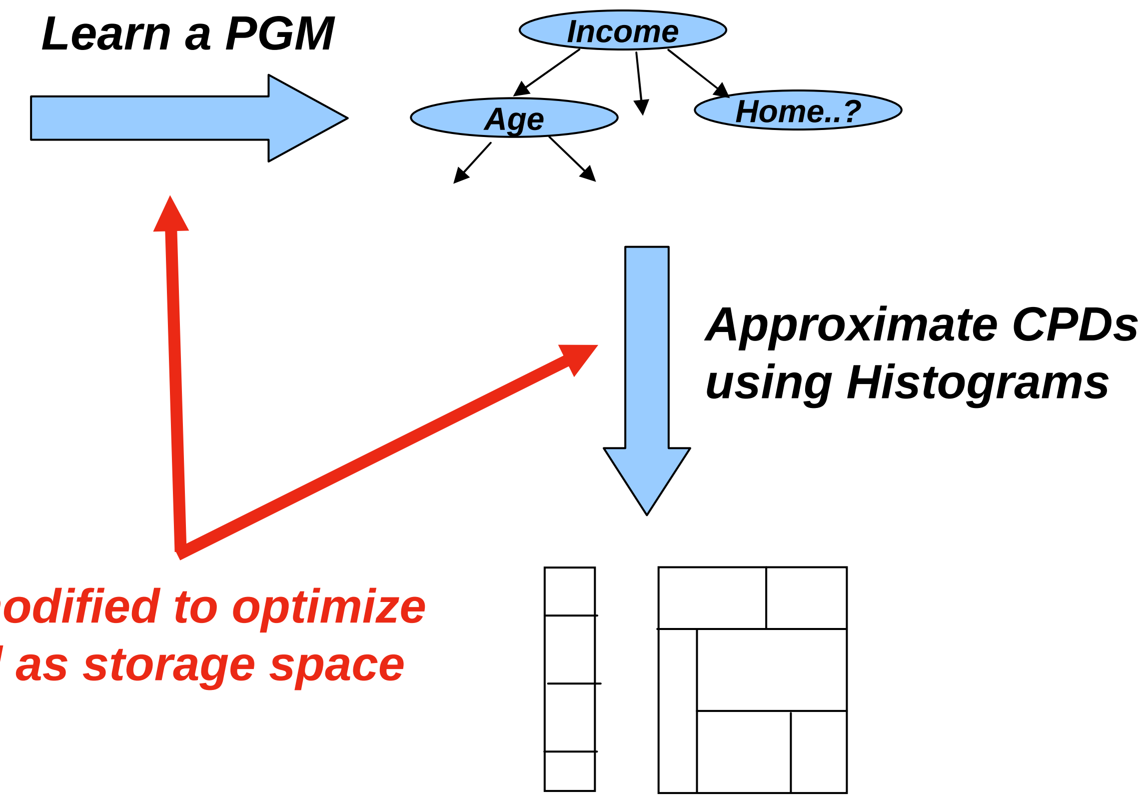
- Errors propagate exponentially [IC'91]
- Optimizers highly sensitive to underestimation
  - May choose nested-loop joins
- Proposed solutions:
  - Multi-dimensional histograms, wavelets [PI'97,MVW'98, GKTD'00]
    - Expensive to build and maintain
    - Suffer from "curse of dimensionality" in high dimensions
  - Random sampling [CDN'07]
    - Not as storage efficient
    - Few matching tuples for high dimensional queries
    - Need different sampling techniques for joins [AGPR'99]

Selectivity Estimation using PGMs

- Eliminating attribute value independence assumption [GTK'01,DGR'01,LWV'03,PMW'03]

Customer

SSN	age	Income	zipcode	Home owner?
..	..	100000	..	Yes
..	..	11000	..	Yes
..	..	50000	..	No
..	..	30000	..	No
..	..	200000	..	Yes



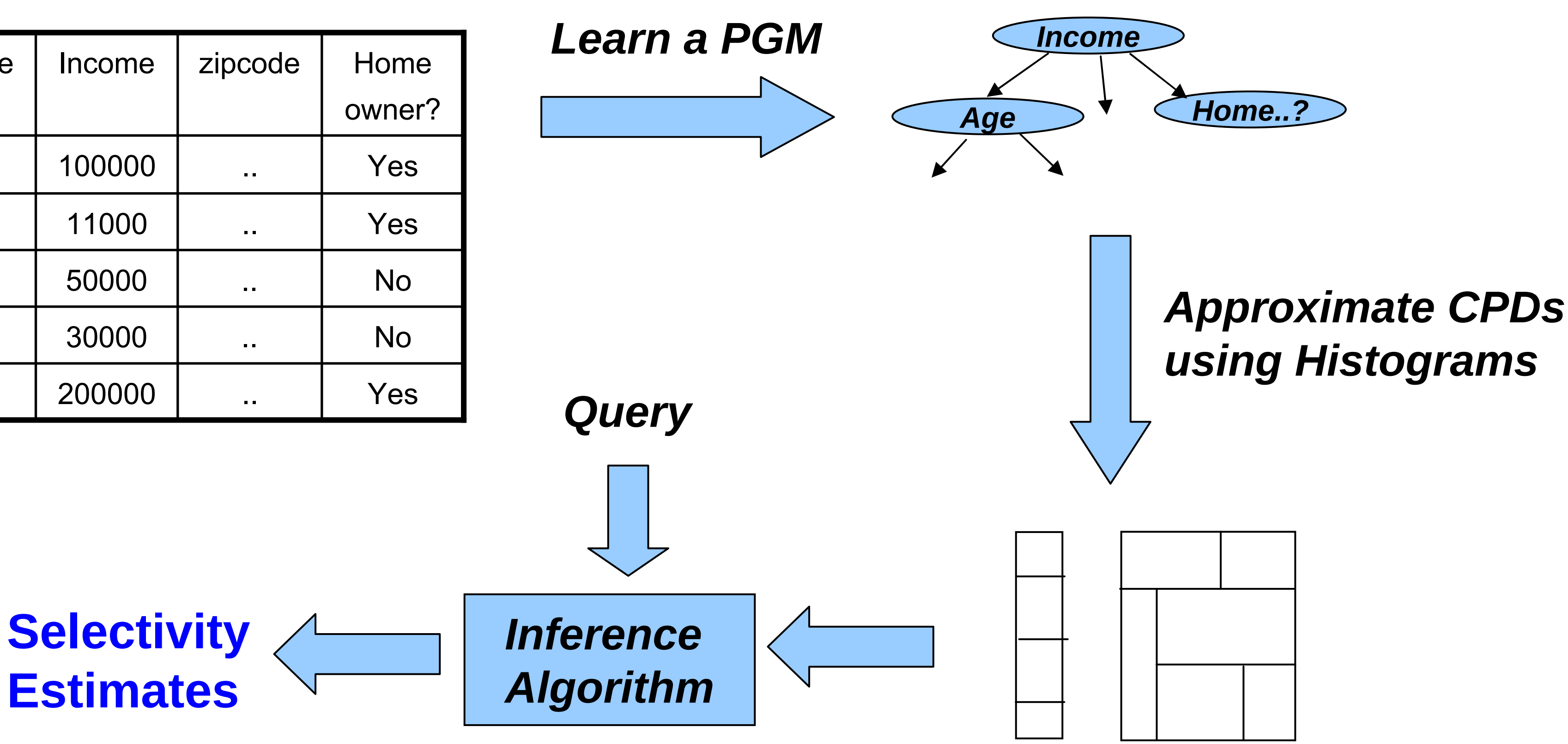
Learning process modified to optimize for accuracy as well as storage space

Selectivity Estimation using PGMs

- Eliminating attribute value independence assumption [GTK'01,DGR'01,LWV'03,PMW'03]

Customer

SSN	age	Income	zipcode	Home owner?
..	..	100000	..	Yes
..	..	11000	..	Yes
..	..	50000	..	No
..	..	30000	..	No
..	..	200000	..	Yes



Selectivity Estimation using PGMs

- Eliminating join uniformity assumption ??

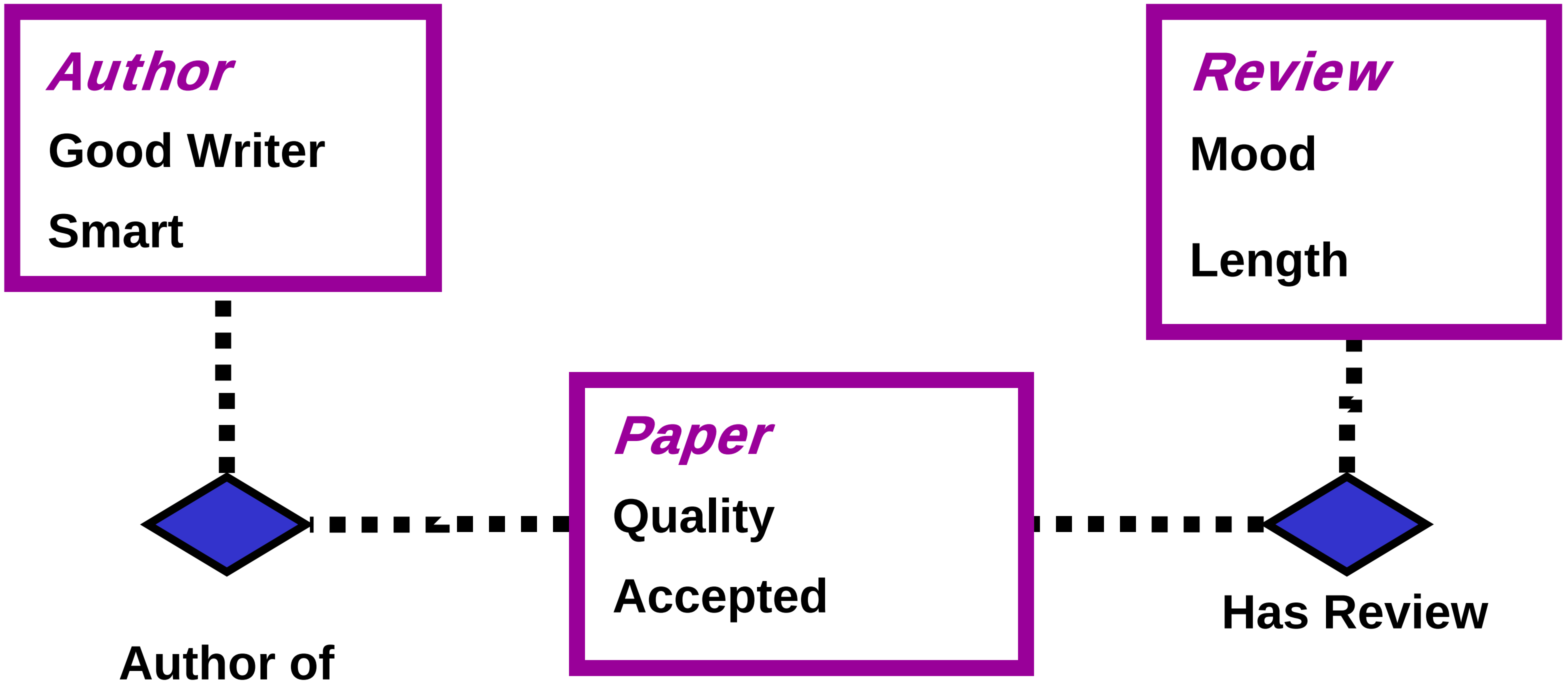
Part II: Outline

- Selectivity Estimation and Query Optimization
- Probabilistic Relational Models
- Probabilistic Databases
- Sensor/Stream Data Management
- References

Probabilistic Relational Models

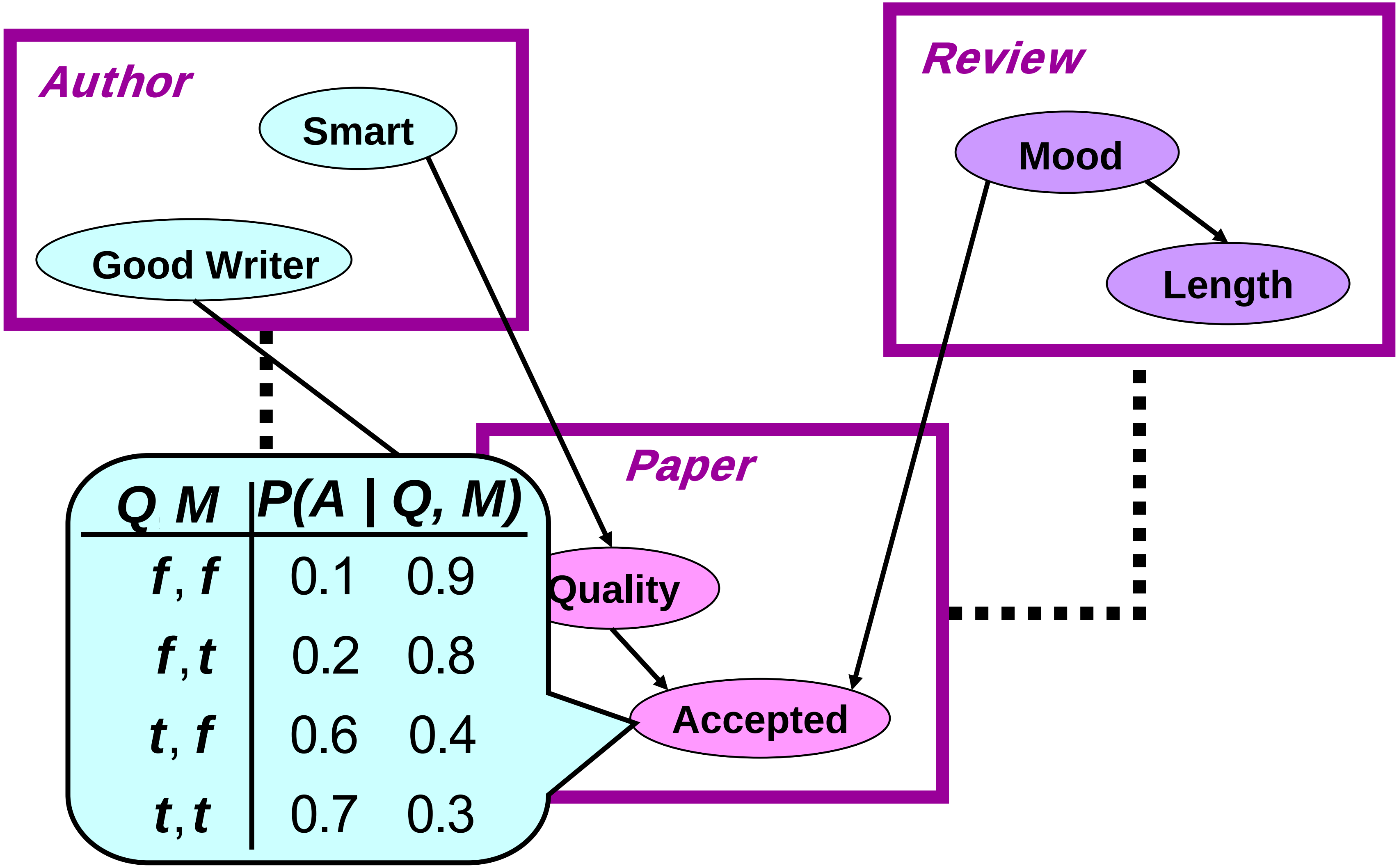
- Real-world data often has highly relational structure
  - There are *entities* and *relationships* between them etc
  - Bayesian networks treat each one individually
  - Will need a huge Bayesian network if we want to represent the uncertainties in such data
- PRMs: Generalization of PGMs to relational framework [FGKP'99]
  - Allows dependence over attributes in different relations through joins
- Significantly enrich both Bayesian networks and relational model

Relational Schema



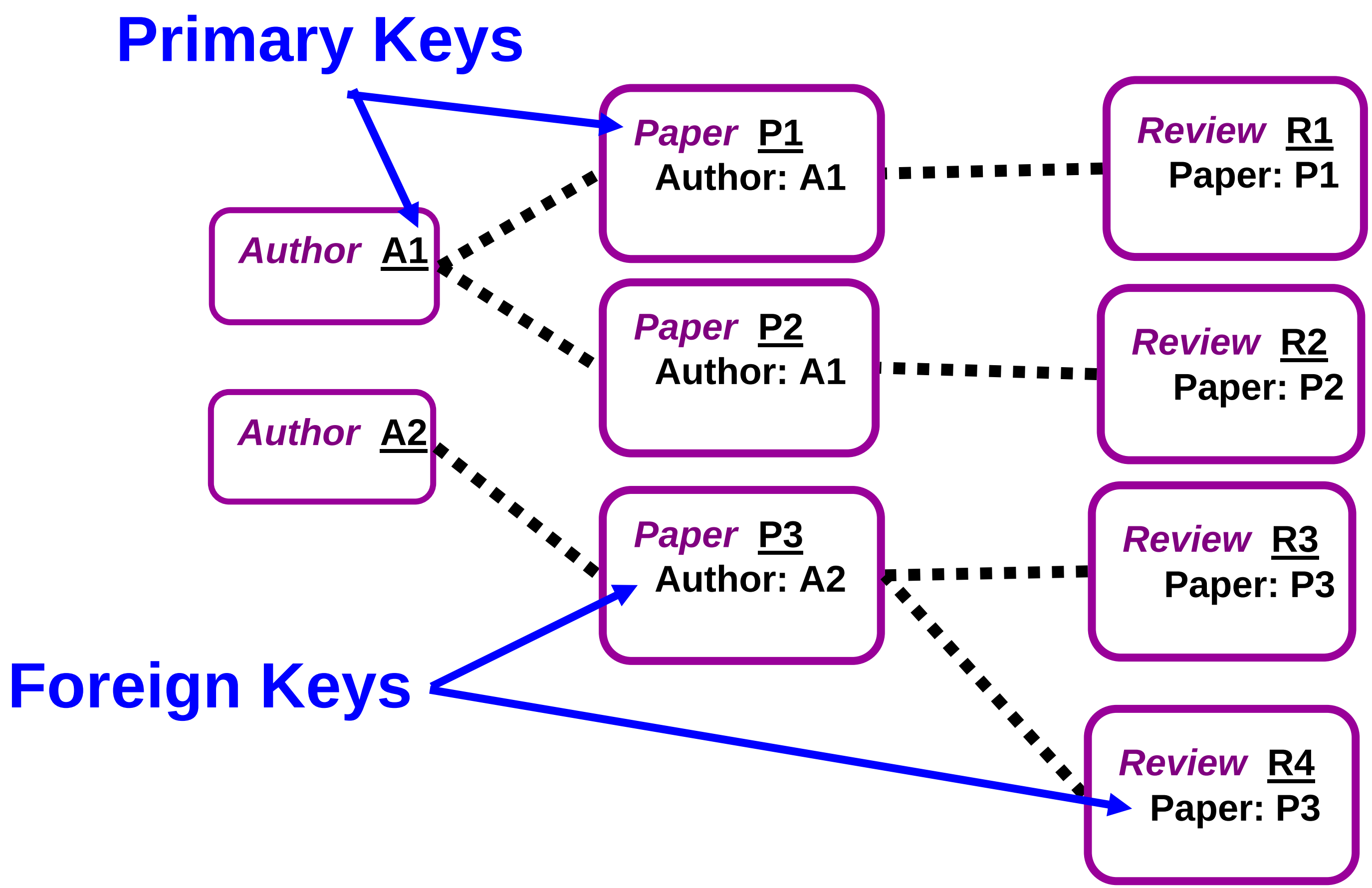
- Describes the types of objects and relations in the database

Probabilistic Relational Model



Q, M	P(A | Q, M)	
f, *f*	0.1	0.9
f, *t*	0.2	0.8
t, *f*	0.6	0.4
t, *t*	0.7	0.3

PRM: Semantics

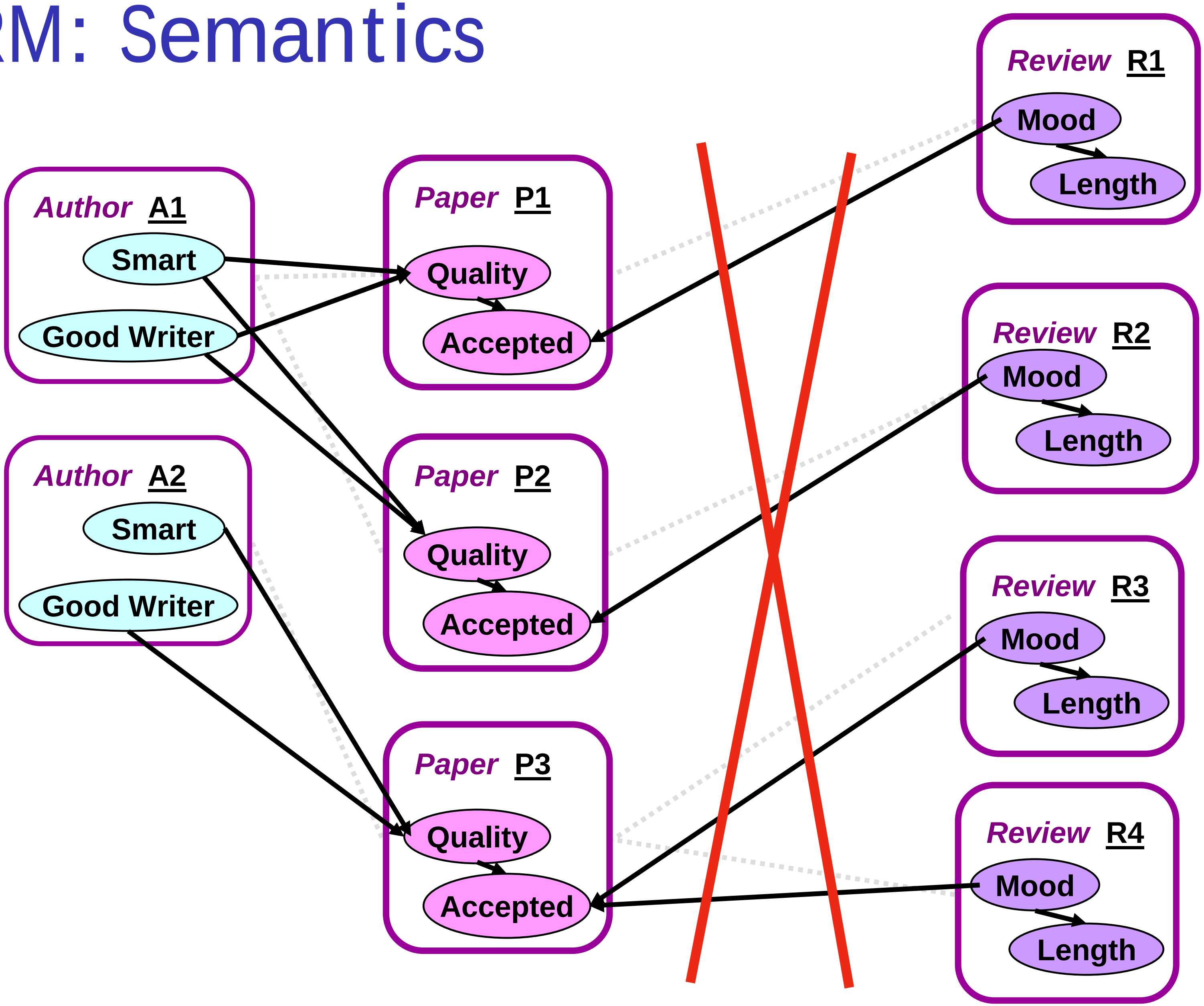


Fixed relational skeleton

- Objects and links between them

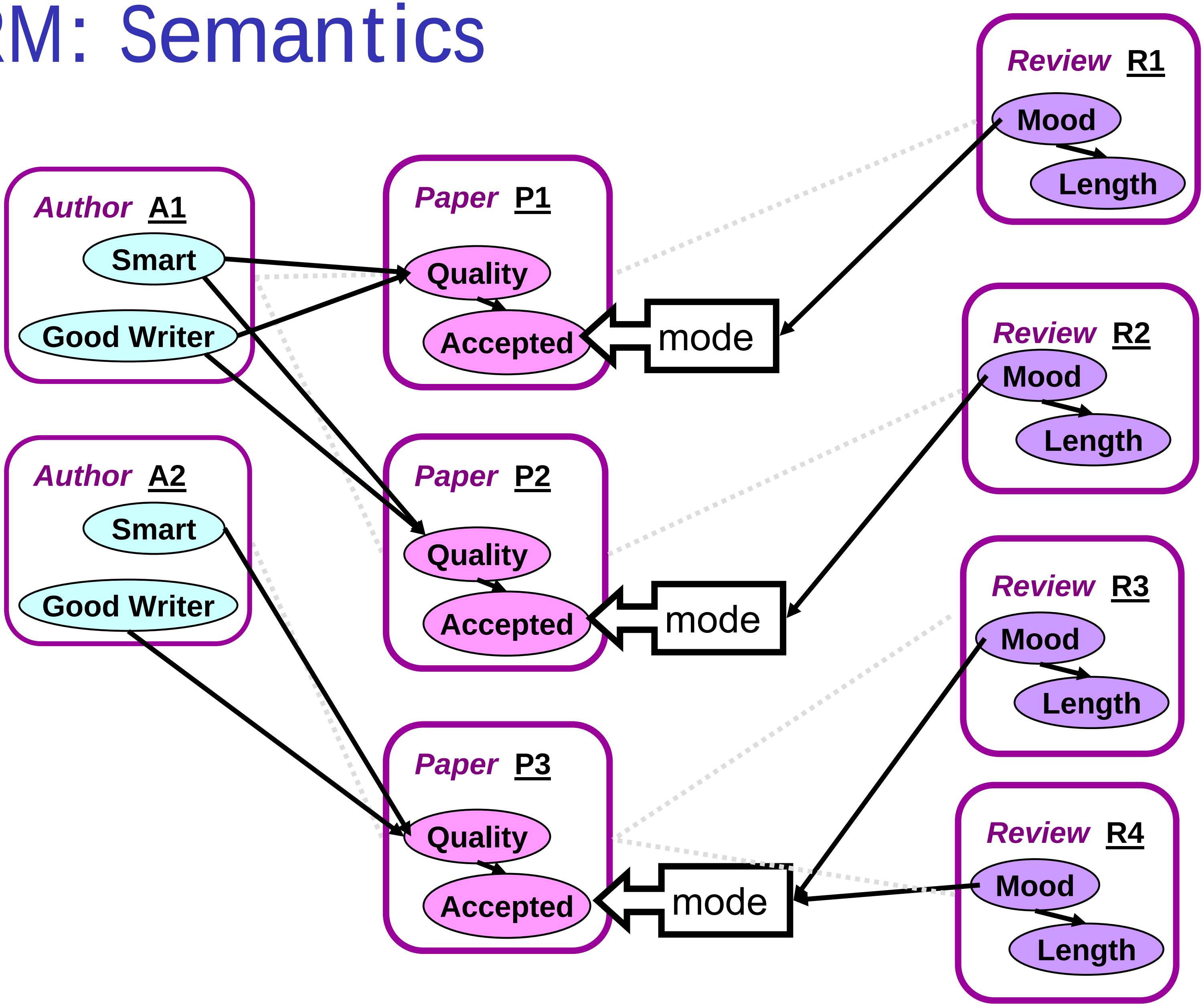
Non-key (descriptive) attributes *uncertain*

PRM: Semantics



PRM defines distribution over instantiations of attributes

PRM: Semantics



Any aggregate function can be used: sum, min, max, avg, mode, count

PRMs: Inference/Generalizations

- Inference
  - Option 1: Construct and use the *ground* Bayesian network
    - Allows exact inference
    - Too large for any reasonable dataset
  - Option 2: Approximate inference
    - E.g. using *loopy belief propagation*
- Generalizations
  - Link uncertainty [GGFKT'02]
  - Finer granularity dependencies using class hierarchies [dGK'00]
  - Undirected dependencies
    - Relational Markov networks [TAK'02]
    - Relational dependency networks [ND'04]
- Exciting research area with huge potential impact in databases !!

Part II: Outline

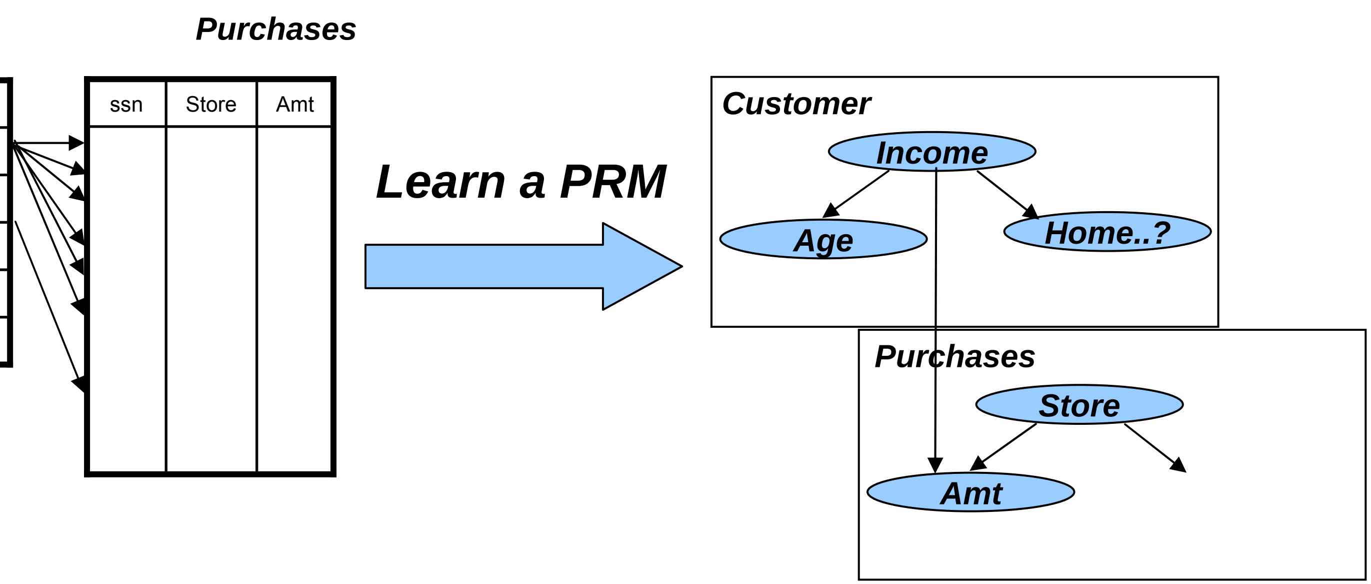
- Selectivity Estimation and Query Optimization (continued)
- Probabilistic Relational Models
- Probabilistic Databases
- Sensor/Stream Data Management
- References

Selectivity Estimation using PGMs

- Eliminating join uniformity assumption [GTK'01]
- Using a *Probabilistic Relational Model*

Customer

SSN	..	Income	..	Homeowner?
..	..	100000	..	Yes
..	..	11000	..	Yes
..	..	50000	..	No
..	..	30000	..	No
..	..	200000	..	Yes



- Can estimate selectivities of joint predicates across relations

Selectivity Estimation using PGMs

- Eliminating join uniformity assumption [GTK'01]
- Using a *Probabilistic Relational Model*

- Caveat:
  - Should not use them blindly
  - Need to add and reason about a new *join indicator* variable

- Called a *Statistical Relational Model*

- Details in Getoor, Tasker, Koller; SIGMOD 2001.

Discussion and Open Problems

- Approximate query processing ?
  - Can use the proposed techniques as they are
  - However, no guarantees on the accuracy of results
    - Optimize accuracy for a given storage
  - To obtain guarantees, optimize for accuracy alone
    - May result in large CPDs
- Using learned PGMs during optimization
  - Optimizers get better selectivity estimates, but otherwise unaware of the modeling
  - May be beneficial to explore tighter integration

Discussion and Open Problems

- Can exploit new types of query plans
  - Based on horizontal partitioning of the relations [BBDW'05,DGHM'05,P'05]
  - Use different plans for different partitions of relations based on attribute values
- Adaptive query processing
  - PGMs ideal for learning the distribution properties
  - Significantly fewer parameters → easier to learn
  - Many research challenges

Part II: Outline

- Selectivity Estimation and Query Optimization
- Probabilistic Relational Models
- Probabilistic Databases
- Sensor/Stream Data Management
- References

Probabilistic Databases

- Motivation: Increasing amounts of *uncertain* data
  - From sensor networks
    - Imprecise data, data with confidence/accuracy bounds
    - Human-observed data
  - Statistical modeling/machine learning
    - Many models provide a distribution over a set of labels (e.g. classification models, HMMs)
  - Approximate/vague queries
  - Information extraction

- "Probability theory" provides a strong foundation to reason about this
  - Caveat: It is not always clear if the underlying uncertainty measure follows probability theory semantics

Probabilistic Databases

- Goal: Managing and querying data annotated with probabilities using databases
- Types of uncertainties
  - Existence uncertainty
    - Don't know if a tuple exists in the database for sure
    - E.g. a sensor may detect a bird, but not 100% sure
  - Attribute-value uncertainty
    - The value of an attribute is not known for sure
    - Instead a distribution over the possible values is provided
    - E.g. a sensor detects a bird for sure, but it may be a sparrow or a dove or something else
- Much work in recent years on both [DS'07]

Correlations in Probabilistic Databases

- Much of the probabilistic data is naturally correlated
  - E.g. sensor data, data integration [AFM'06]
- Even if not..
  - Correlations get introduced during query processing

Example Probabilistic Database

- Example from Dalvi and Suciu [2004]

S

	A	B	*prob*
s1	m	1	*0.6*
s2	n	1	*0.5*

T

	C	D	*prob*
t1	1	p	*0.4*

Possible worlds

instance	probability
{s1, s2, t1}	0.12
{s1, s2}	0.18
{s1, t1}	0.12
{s1}	0.18
{s2, t1}	0.08
{s2}	0.12
{t1}	0.08
{}	0.12

Correlations during query processing

- Example from Dalvi and Suciu [2004]

S

	A	B	*prob*
s1	m	1	*0.6*
s2	n	1	*0.5*

T

	C	D	*prob*
t1	1	p	*0.4*

$\bowtie$

$S \bowtie_{B=C} T$

	A	B	C	D	*prob*
i1	m	1	1	p	*0.24*
i2	n	1	1	p	*0.20*

Not independent !!!

$\pi_D(S \bowtie_{B=C} T)$

	C	D	*prob*
r	1	p	~~*0.392*~~

Correlations in Probabilistic Databases

- Much of the probabilistic data is naturally correlated
  - E.g. sensor data, data integration
- Even if not..
  - Correlations get introduced during query processing
- Can use PGMs to capture such correlations

Example: Mutual Exclusivity

S

	A	B	*prob*
s1	m	1	*0.6*
s2	n	1	*0.5*

T

	C	D	*prob*
t1	1	p	*0.4*

Possible worlds

instance	probability
{s1, s2, t1}	0
{s1, s2}	0.3
{s1, t1}	0
{s1}	0.3
{s2, t1}	0.2
{s2}	0
{t1}	0.2
{}	0

X_{s1}	X_{t1}	*f1()*
0	0	0
0	1	0.4
1	0	0.6
1	1	0

X_{s2}	*f2()*
0	0.5
1	0.5

Possible worlds (if desired) computed using inference

Query Evaluation [SD'07]

- Introduce new factors as new tuples generated

X_{s1}	X_{t1}	*f1()*
0	0	0
0	1	0.4
1	0	0.6
1	1	0

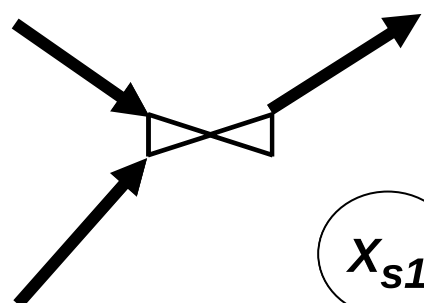
S

	A	B
s1	m	1
s2	n	1

X_{s2}	*f2()*
0	0.5
1	0.5

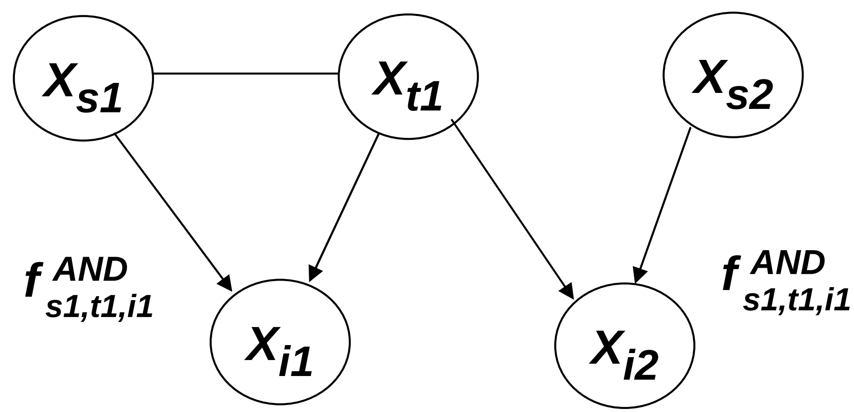
T

	C	D
t1	1	p



$S \bowtie_{B=C} T$

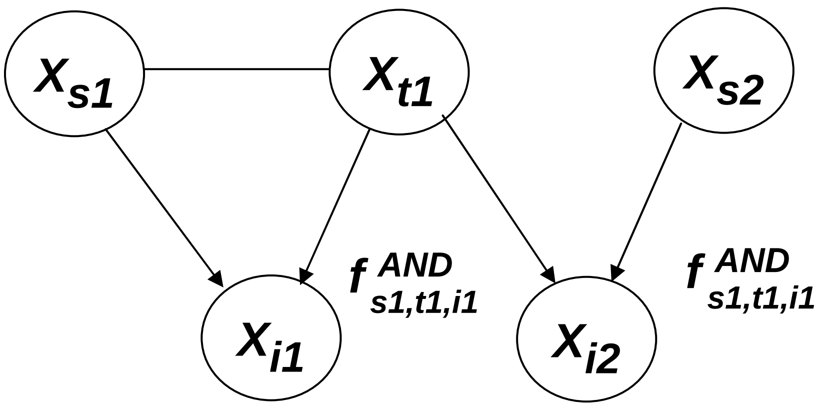
	A	B	C	D
i1	m	1	1	p
i2	n	1	1	p



Query Evaluation [SD'07]

- Introduce new factors as new tuples generated

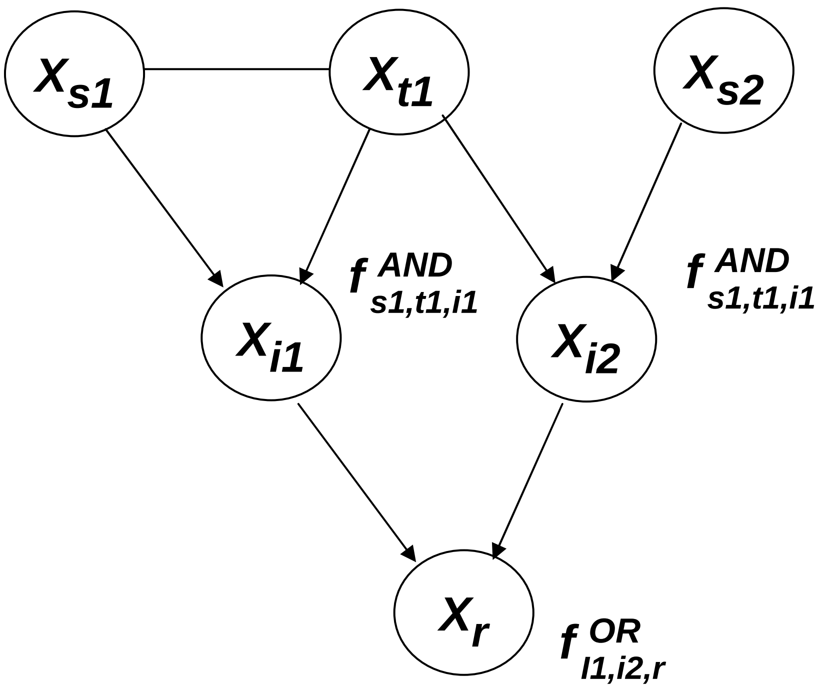
X_{s1}	X_{t1}	X_{i1}	f^{AND}
0	0	0	1
0	1	0	1
1	0	0	1
1	1	0	0
0	0	1	0
0	1	1	0
1	0	1	0
1	1	1	1



Query Evaluation [SD'07]

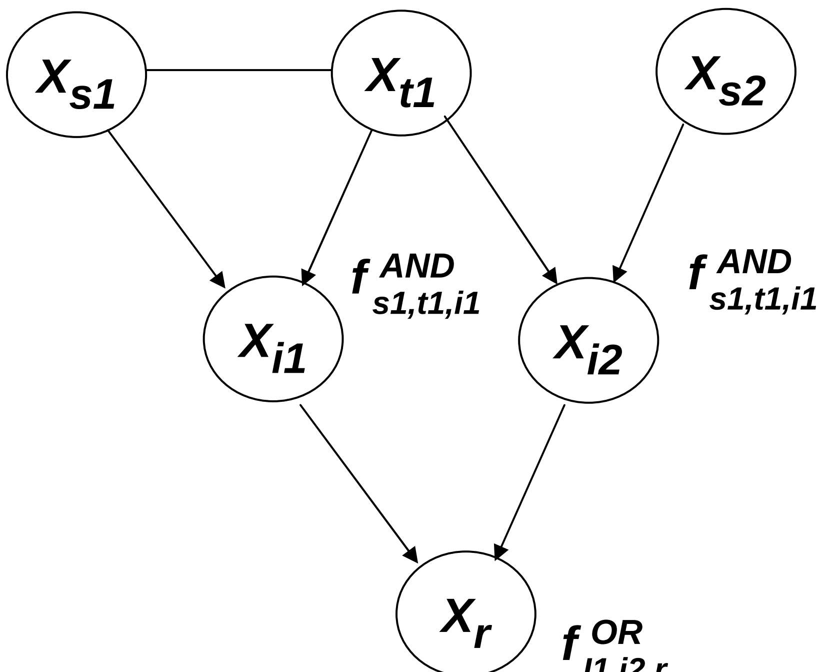
- Introduce new factors as new tuples generated

X_{i1}	X_{i2}	X_r	f^{OR}
0	0	0	1
0	1	0	0
1	0	0	0
1	1	0	0
0	0	1	0
0	1	1	1
1	0	1	1
1	1	1	1



Query Evaluation [SD'07]

- Query evaluation ≡ Inference !!
- Can use variable elimination or junction tree..
- Can also use approximate inference algorithms



Discussion

- Similar to intensional semantics [FR'97,DS'04]
  - Except this exposes the structure of the problem
  - Can exploit for more efficient execution
- "Safe plans" on independent tuples generate tree-structured models
  - Highly efficient inference

Part II: Outline

- Selectivity Estimation and Query Optimization
- Probabilistic Relational Models
- Probabilistic Databases
- Sensor/Stream Data Management
- References

Motivation

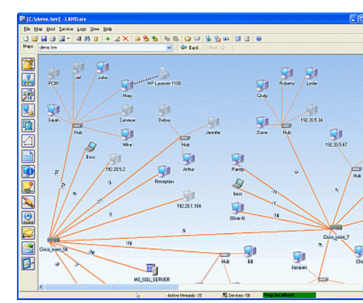
- Unprecedented, and rapidly increasing, instrumentation of our every-day world

Distributed measurement networks (e.g. GPS)

RFID

Wireless sensor networks

Industrial Monitoring



Network Monitoring

Sensor Data Management: Challenges

- Data streams generated at very high rates
- Strong spatio-temporal correlations in the data
- In-network, distributed processing tasks
  - Global inference needed to achieve consistency
- Need for higher-level modeling over the data
  - Typically imprecise, unreliable and incomplete data
    - Measurement noises, failures, biases …
  - Application often need higher-level, *hidden* variables
  - Pattern recognition, forming stochastic descriptions..

Sensor Data Management

- A statistical/probabilistic model of the data must be incorporated in the sensor data processing
- Probabilistic graphical models are a natural
  - Can capture and exploit the spatial and temporal nature of the underlying process
  - Minimize the number of parameters
  - Amenable to distributed processing

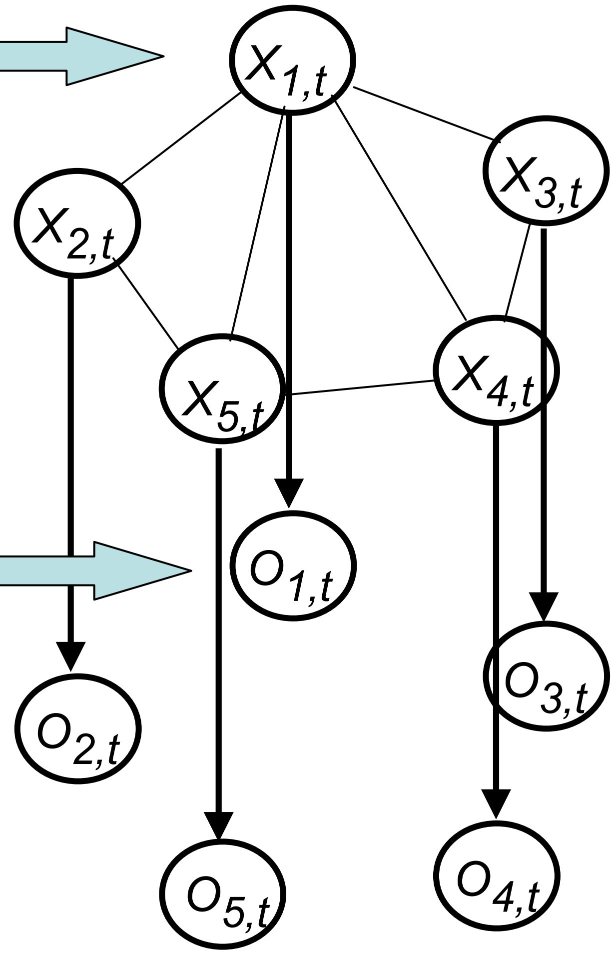
Outline

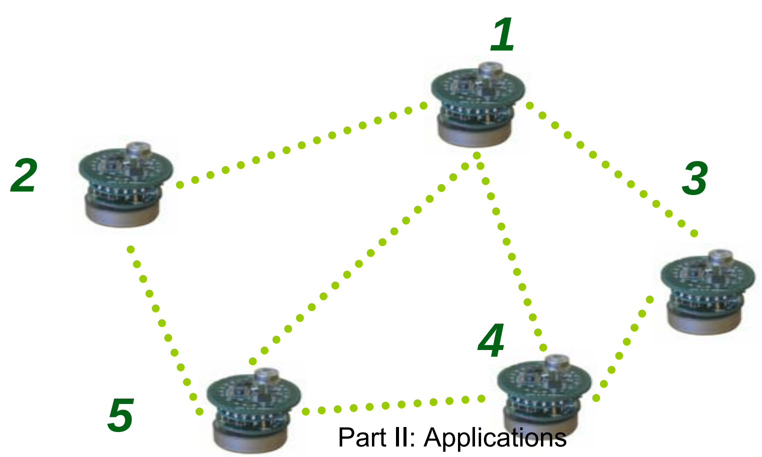
- A generic temporal model for sensor stream data
- Applications
  - Online estimation and filtering
  - Inferring hidden variables
  - Model-based query processing
  - In-network inference
  - Miscellaneous

True *temperature at X_1 at time t*

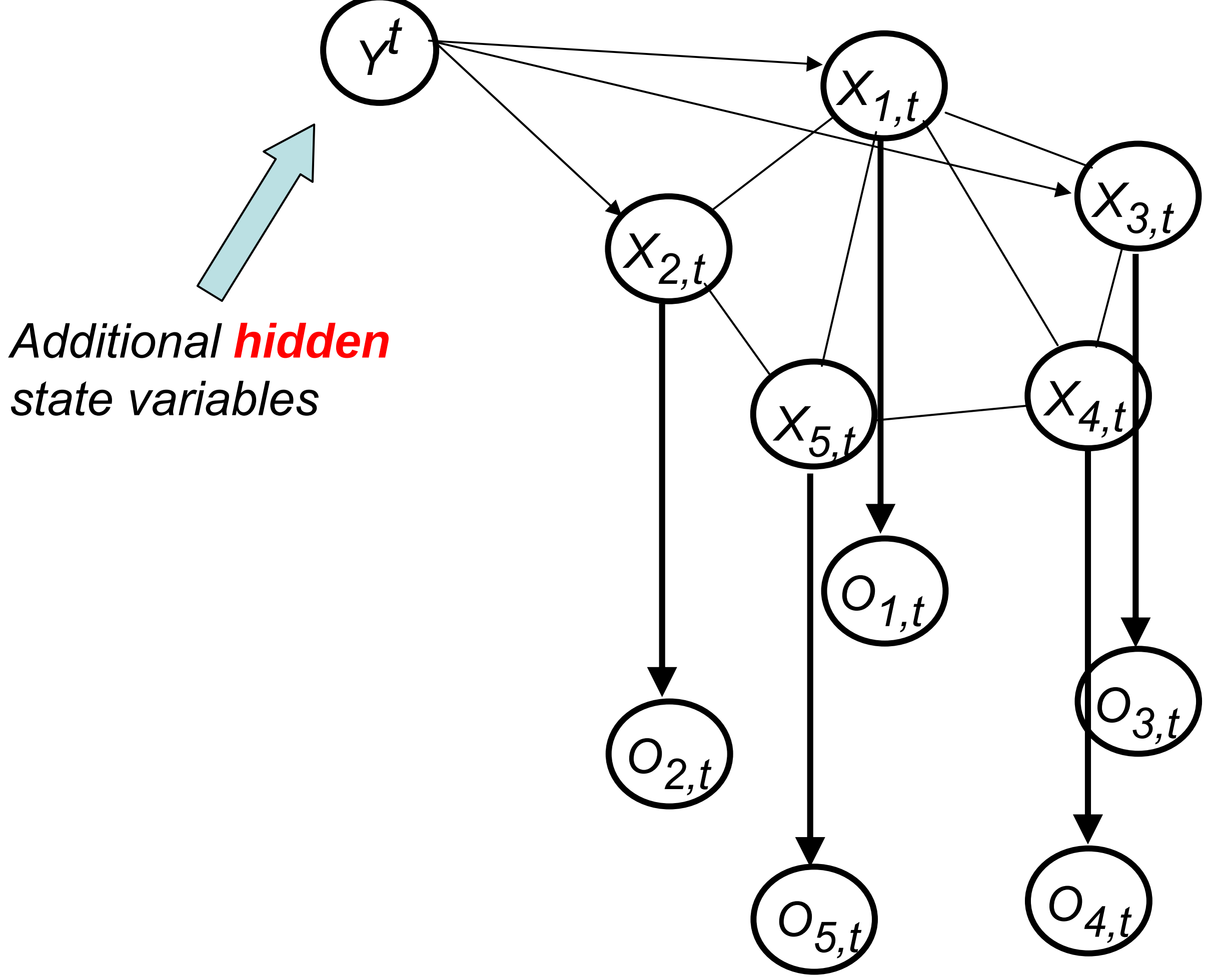
Interpretation: $X_{4,t}$ independent of $X_{2,t}$ given $X_{1,t}$ and $X_{5,t}$

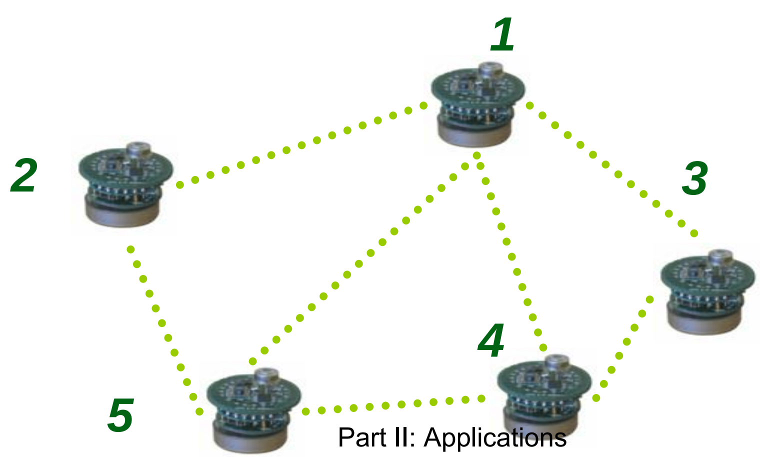
Observed *temperature at X_1 at time t*



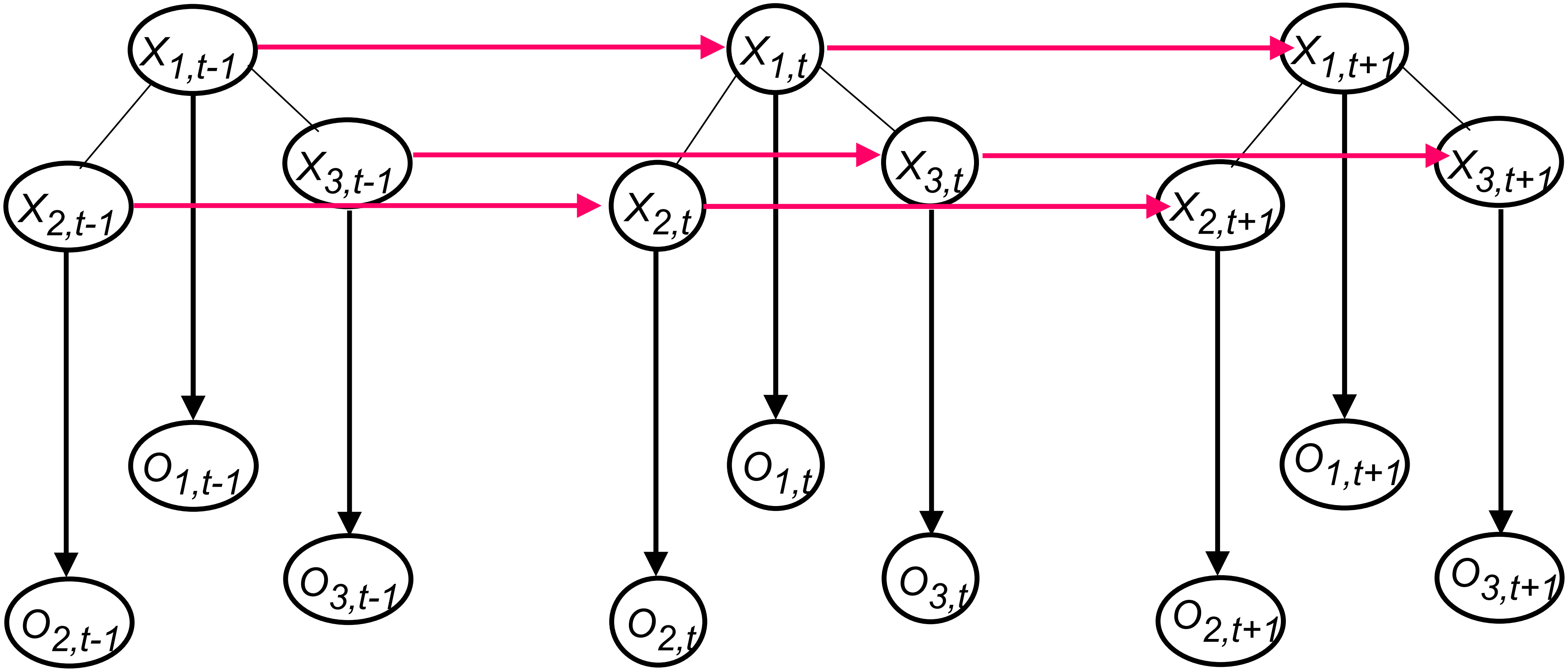


SENSOR NETWORK



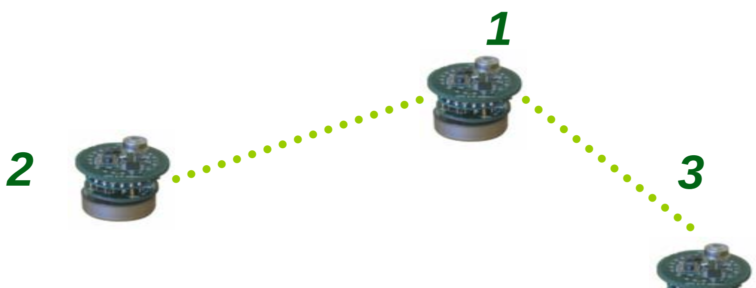


SENSOR NETWORK

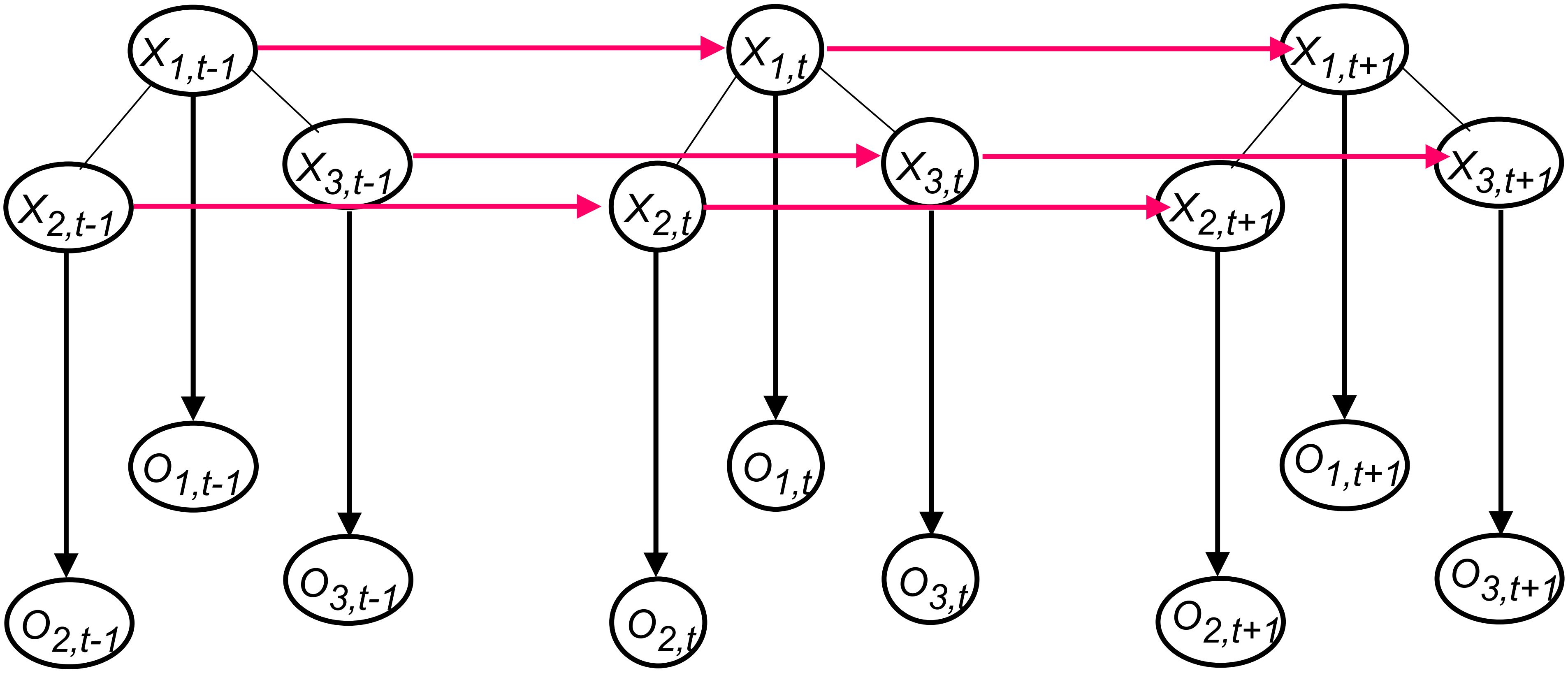


Markov Property

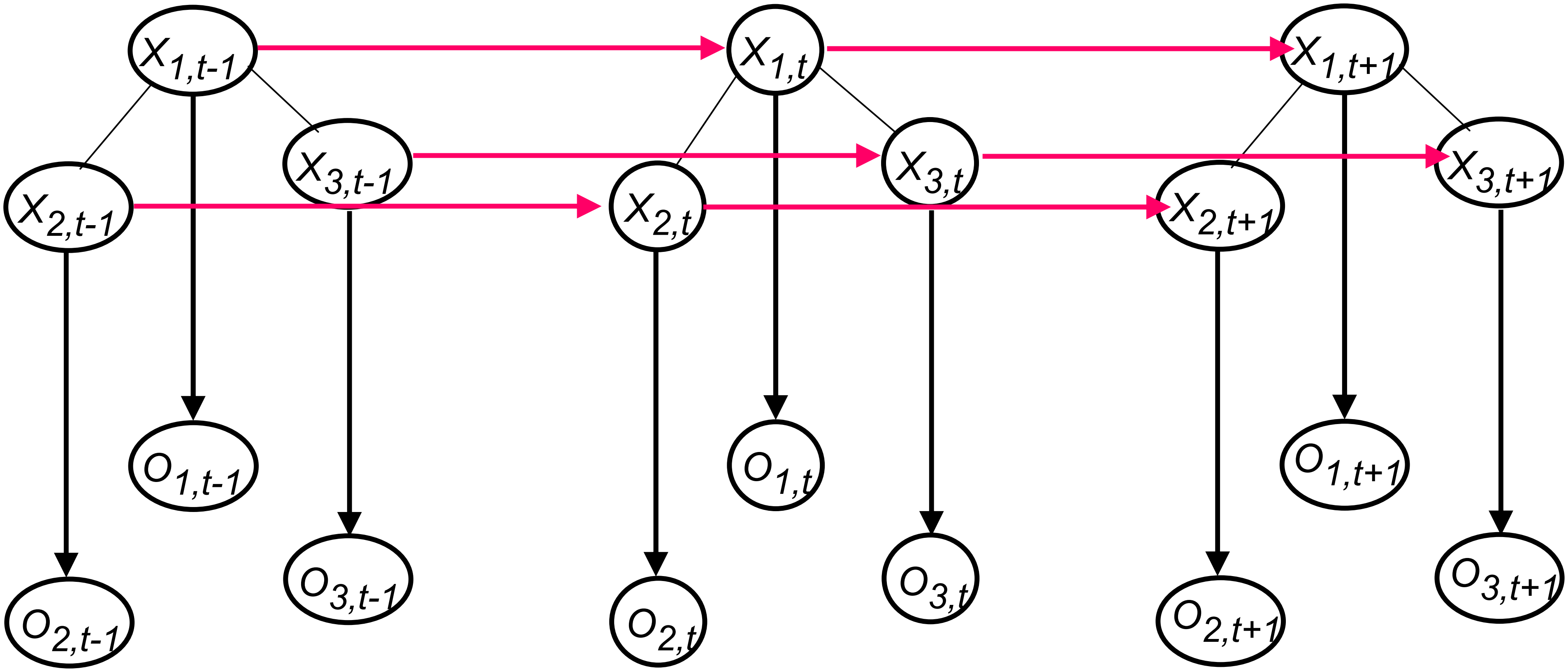
Interpretation: $\{X_{i,t+1}\}$ independent of $\{X_{i,t-1}\}$ given $\{X_{i,t}\}$



SENSOR NETWORK



State evolution can be modeled as a Dynamic Bayesian Network

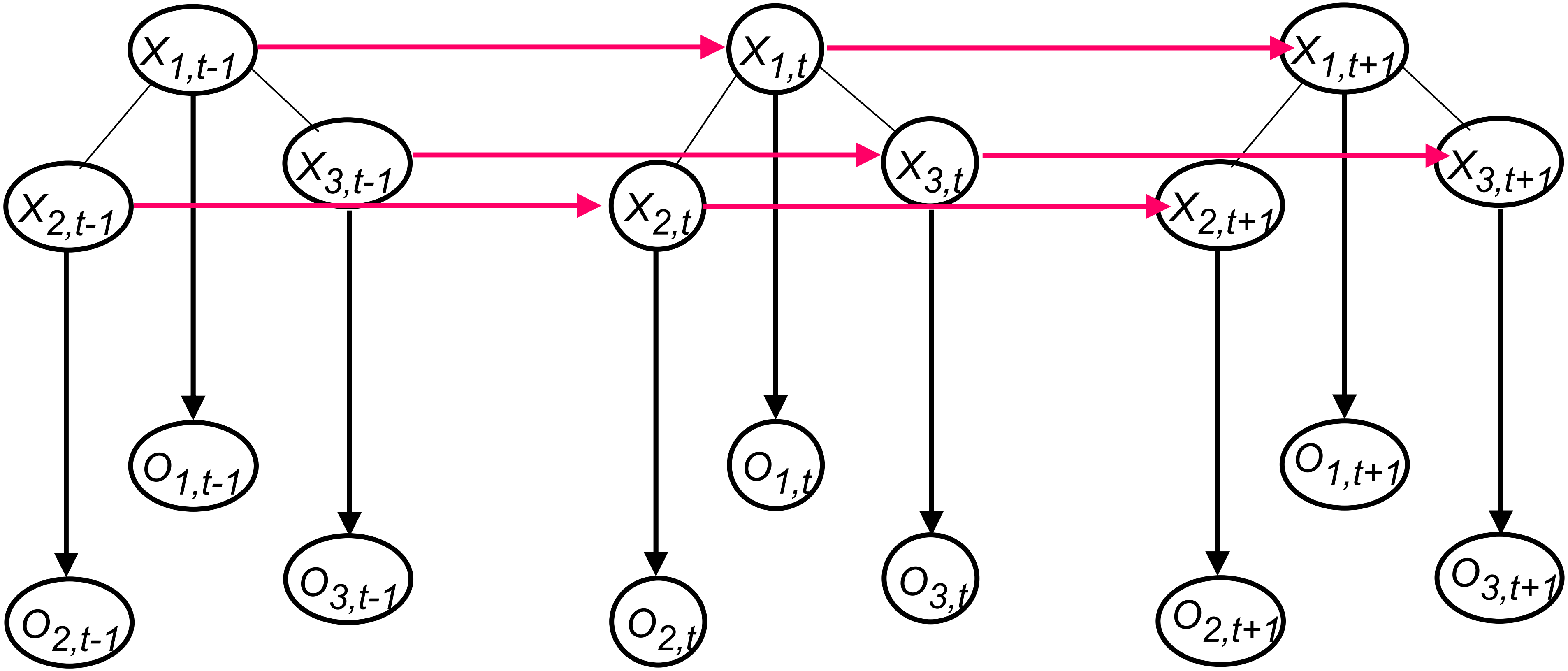


Parameters ?

(1) System model

Prior: $p(X_{1,0},X_{2,0},X_{3,0})$

Evolution: $p(X_{1,t},X_{2,t},X_{3,t} \mid X_{1,t-1},X_{2,t-1},X_{3,t-1})$



Parameters ?

(2) Measurement model

$$p(O_{1,t}, O_{2,t}, O_{3,t} \mid X_{1,t}, X_{2,t}, X_{3,t})$$

Outline

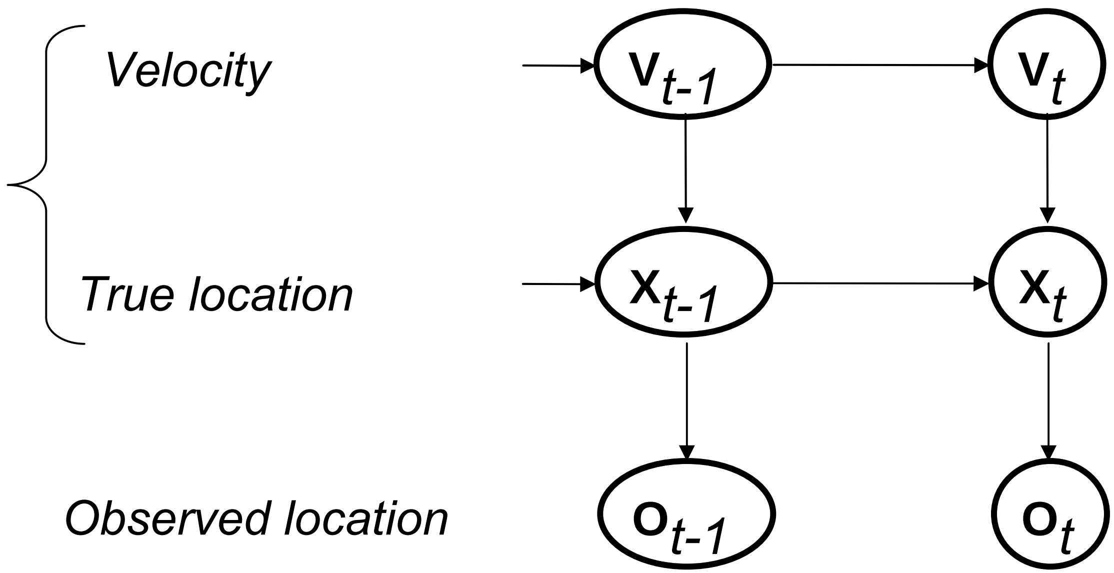
- A generic temporal model for sensor stream data
- Applications
  - Online estimation and filtering
  - Inferring hidden variables
  - Model-based query processing
  - In-network inference
  - Miscellaneous

Application: Online Estimation and Filtering

- Using *linear Gaussian dynamical systems*
  - E.g. Kalman Filters
- Task: Estimating *velocity* and *location* from noisy GPS readings

Hidden Variables

Assumed to have Gaussian Distributions



Application: Online Estimation and Filtering

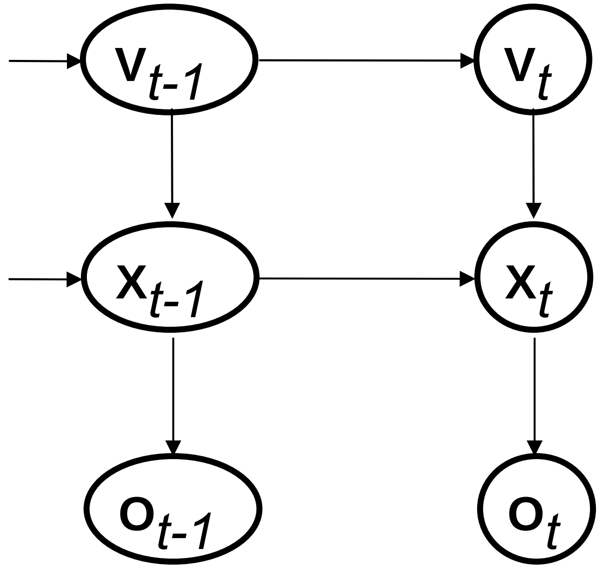
- Using *linear Gaussian dynamical systems*
  - E.g. Kalman Filters
- Task: Estimating *velocity* and *location* from noisy GPS readings

$$p(v_t | v_{t-1}) = N(v_{t-1}, \sigma_v)$$

$$p(x_t | x_{t-1}, v_t) = N(x_{t-1} + v_t, \sigma_x)$$

$$p(o_t | x_t) = N(x_t, \sigma_o)$$

$$Prior: \; p(v_0), p(x_0)$$



Application: Online Estimation and Filtering

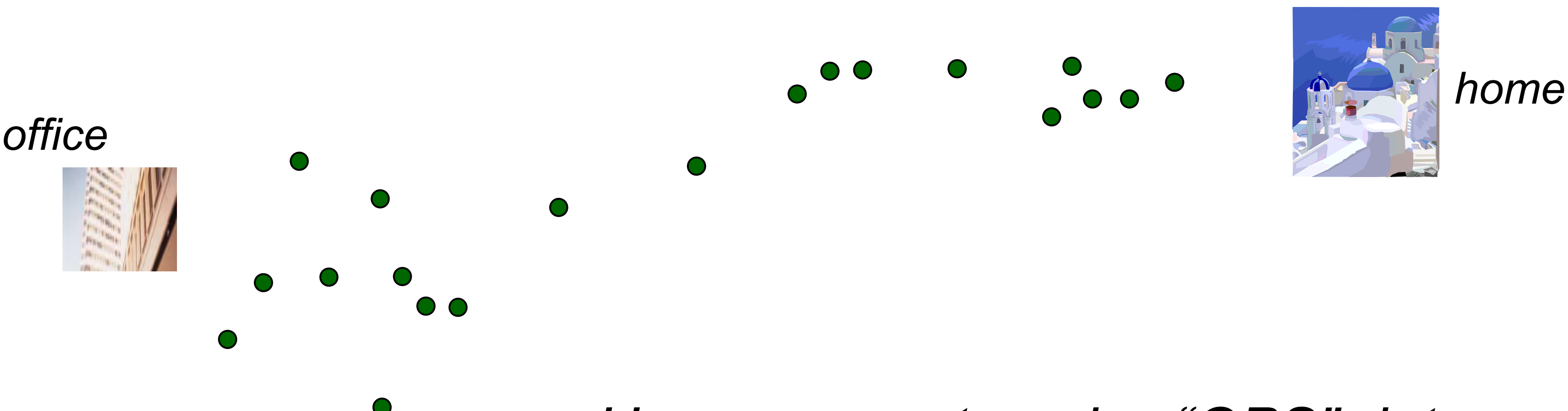
- Using *linear Gaussian dynamical systems*
  - E.g. Kalman Filters
- Closed-form equations for state estimation [Kalman'60]
  - Because of the *linear Gaussian* assumption
- LDS Applications:
  - Autopilot
  - Inertial guidance systems
  - Radar tracker
  - Economics…
- In databases:
  - Adaptive stream resource management [JCW'04]
  - Approximate querying in sensor networks [DGHHM'04]

Outline

- A generic temporal model for sensor stream data
- Applications
  - Online estimation and filtering
  - Inferring hidden variables
  - Model-based query processing
  - In-network inference
  - Miscellaneous

Application: Inferring Hidden Variables

- Inferring "transportation mode"/ "activities" [P+04]
  - Using easily obtainable sensor data (GPS, RFID proximity data)
  - Can do much if we can infer these automatically



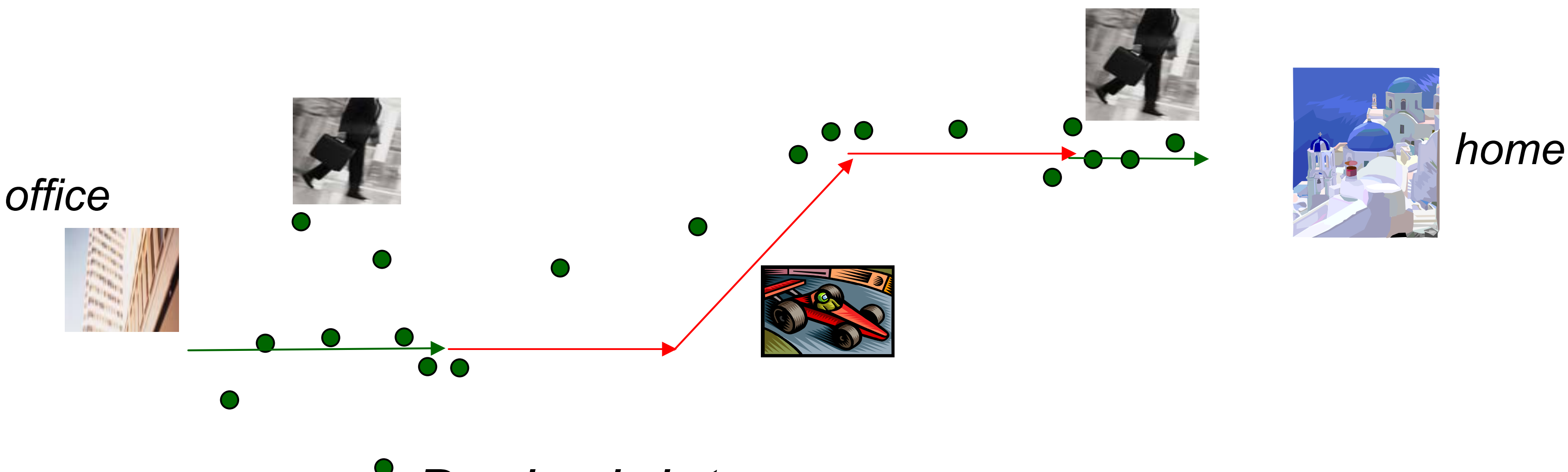
Have access to noisy "GPS" data

<u>Infer</u> the transportation mode:

walking, running, in a car, in a bus

Application: Inferring Hidden Variables

- Inferring “transportation mode”/ “activities” [P+04]
  - Using easily obtainable sensor data (GPS, RFID proximity data)
  - Can do much if we can infer these automatically



Desired data:

Clean path annotated with transportation mode

Online, in real-time

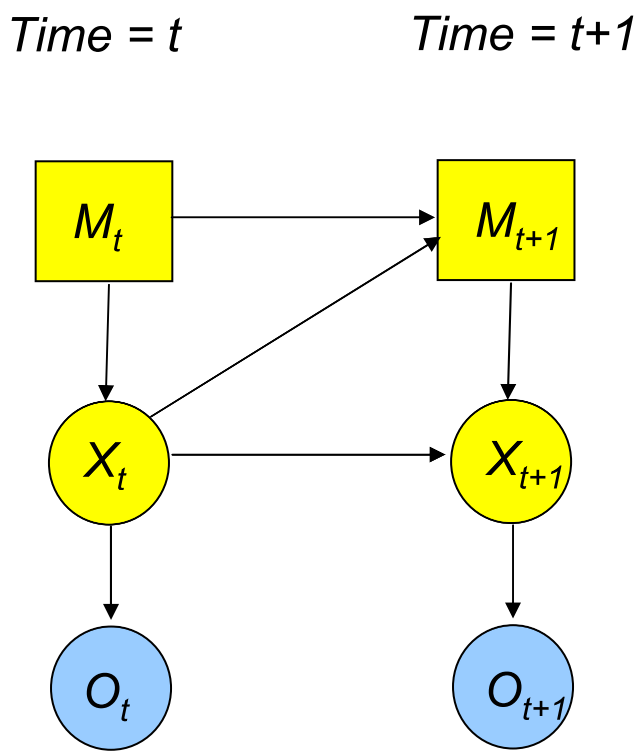
Application: Inferring Hidden Variables

Use a dynamic Bayesian network to model the system state

Transportation Mode:
Walking, Running, Car, Bus

True velocity and location

Observed location



Application: Inferring Hidden Variables

Given a sequence of observations (O_t), infer most likely M_t's that explain it.

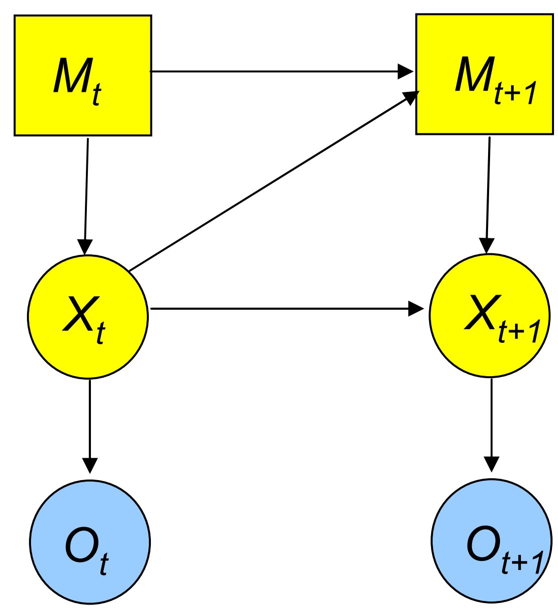
Alternatively, could provide a probability distribution on the possible M_t's.

Time = t *Time = t+1*

Transportation Mode:
Walking, Running, Car, Bus

True velocity and location

Observed location



Outline

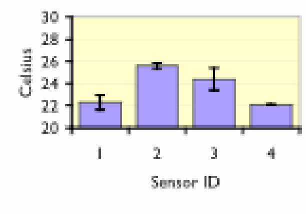
- A generic temporal model for sensor stream data
- Applications
  - Online estimation and filtering
  - Inferring hidden variables
  - Model-based query processing
  - In-network inference
  - Miscellaneous

Application: Model-based Query Processing [DGMHH'04,SBEMY'06]

USER

Declarative Query
Select nodeID,
temp ± .1C, conf(.95)
Where nodeID in {1..6}

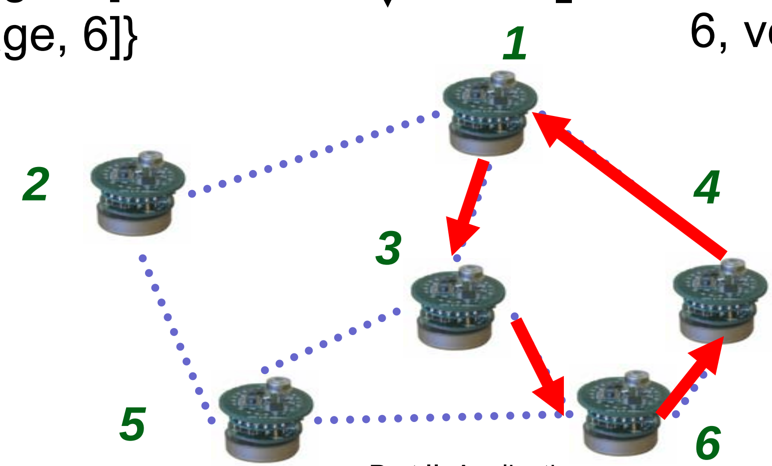
Query Results
1, 22.73, 100%
…
6, 22.1, 99%



Probabilistic Model ⟷ *Query Processor*

Observation Plan
{[temp, 1],
[voltage, 3],
[voltage, 6]}

Data
1, temp = 22.73,
3, voltage = 2.73
6, voltage = 2.65



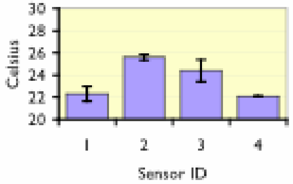
SENSOR NETWORK

Application: Model-based Query Processing [DGMHH'04,SBEMY'06]

<u>Declarative Query</u>
Select nodeID,
temp ± .1C, conf(.95)
Where nodeID in {1..6}

<u>USER</u>

<u>Query Results</u>
1, 22.73, 100%
…
6, 22.1, 99%

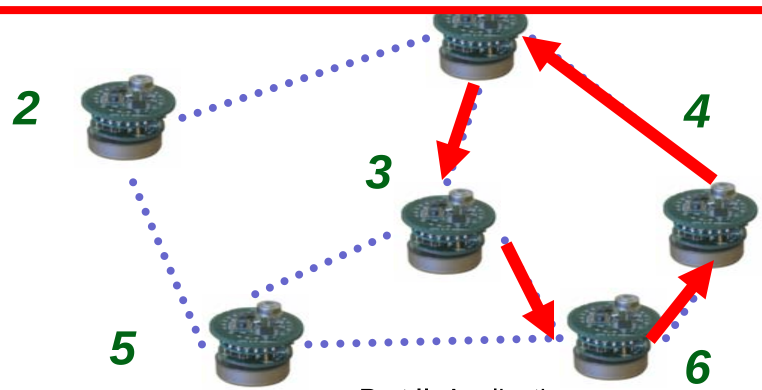


<u>Advantages</u>:

Exploit correlations for efficient approximate query processing

Handle noise, biases in the data

Predict missing or future values



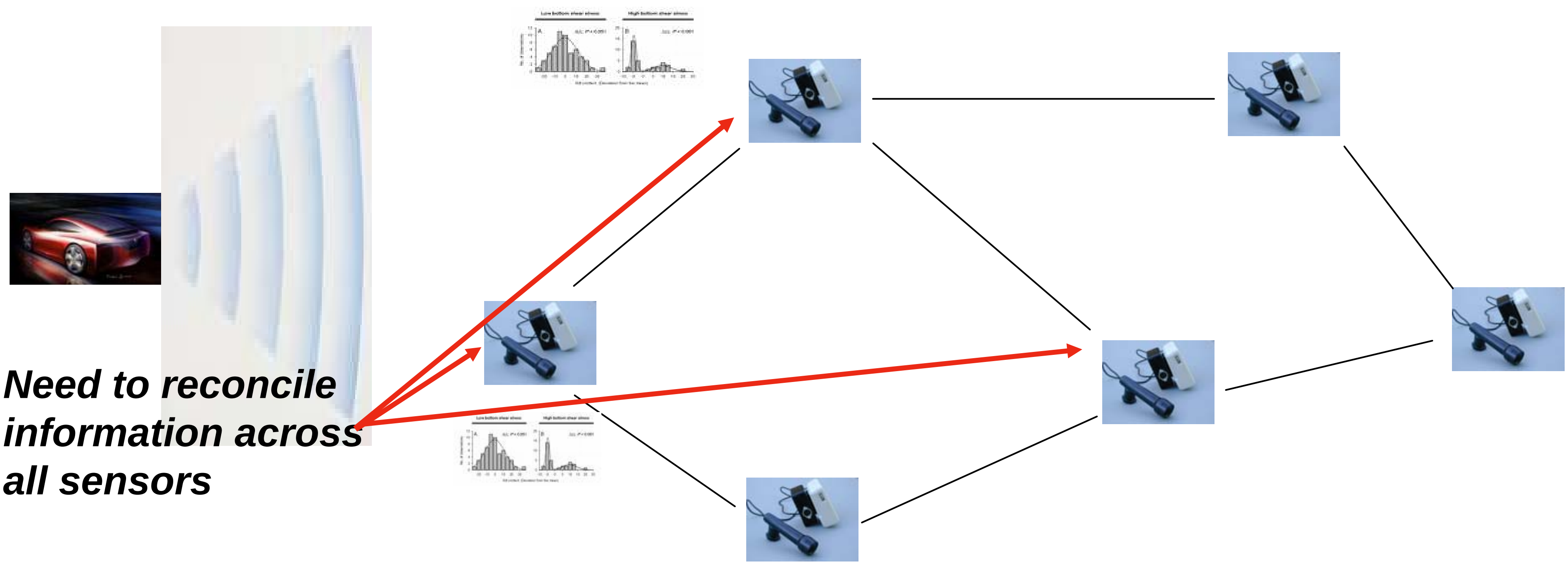
<u>SENSOR NETWORK</u>

Outline

- A generic temporal model for sensor stream data
- Applications
  - Online estimation and filtering
  - Inferring hidden variables
  - Model-based query processing
  - In-network inference
  - Miscellaneous

Application: In-network Inference [PGM'05]

- Often need to do in-network, distributed inference
  - Target tracking through information fusion
  - Optimal control (for actuation)
  - Distributed sensor calibration (using neighboring sensors)
  - In-network regression or function fitting

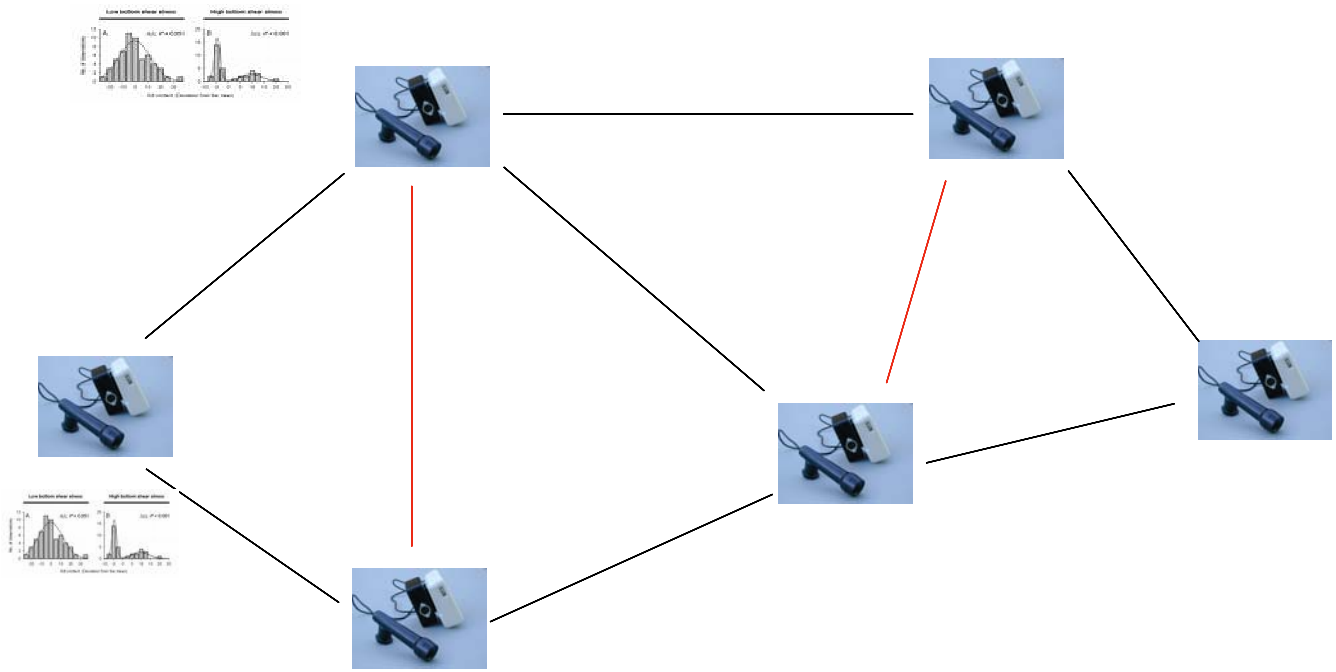


Application: In-network Inference [PGM'05]

- Often need to do in-network, distributed inference
  - Target tracking through information fusion
  - Optimal control (for actuation)
  - Distributed sensor calibration (using neighboring sensors)
  - In-network regression or function fitting
- Obey a common structure:
  - Each sensor has/observes some *local* information
  - Information across sensors is correlated
  - The information must be combined together to form a *global picture*
  - The global picture (or relevant part thereof) should be sent to each sensor

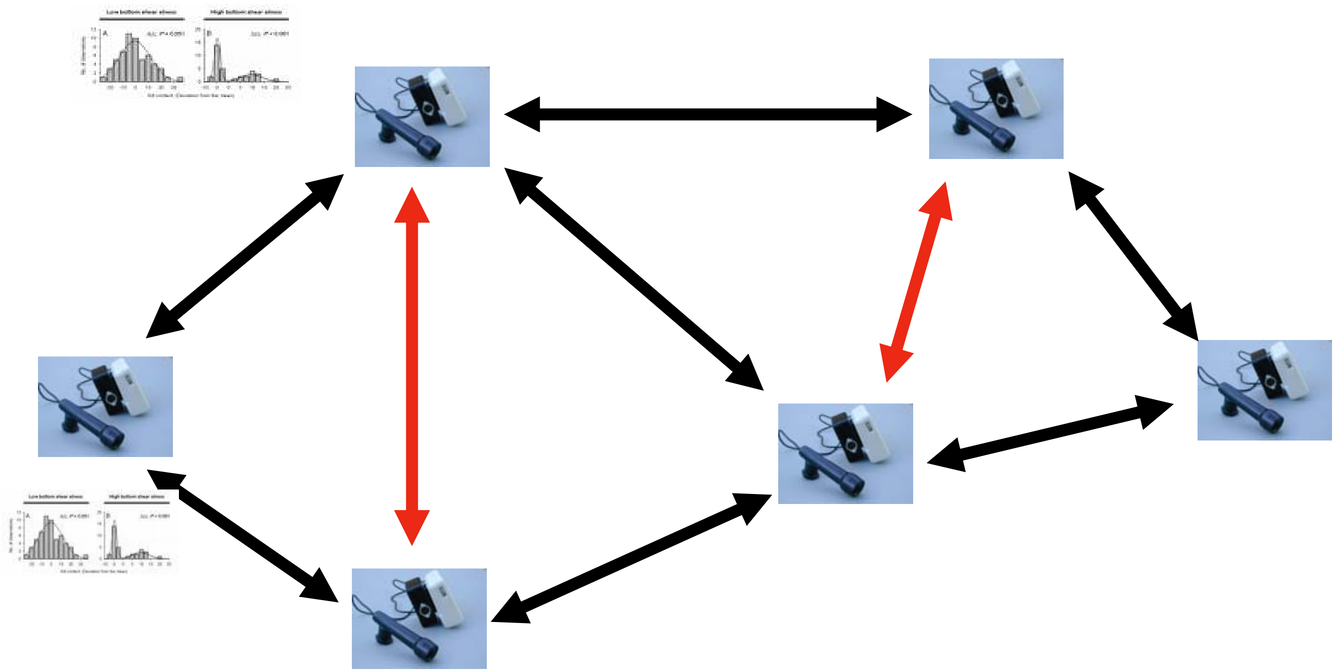
Application: In-network Inference [PGM'05]

- Naïve option:
  - Collect all data at the centralized base station – too expensive
- Using graphical models
  - Form a junction tree on the nodes directly
  - Use message passing (or loopy propagation [CP'03]) to form a global consistent view



Application: In-network Inference [PGM'05]

- Naïve option:
  - Collect all data at the centralized base station – too expensive
- Using graphical models:
  - Form a junction tree on the nodes directly
  - Use message passing (or loopy propagation [CP'03]) to form a global consistent view



Outline

- A generic temporal model for sensor stream data
- Applications
  - Online estimation and filtering
  - Inferring hidden variables
  - Model-based query processing
  - In-network inference
  - Miscellaneous

Applications: Miscellaneous

- Data compression [CDHH'06]
  - Central task in sensor networks
    - Collect all observed data at the base station at specified frequency
  - Challenge: How to exploit the correlations
  - Probabilistic graphical models ideally suited:
    - Can capture the correlations/pattern
    - Allow for local checking of constraints/correlations
- Fault/anomaly detection
- Distributed regression
- Sensor calibration

Part II: Outline

- Selectivity Estimation and Query Optimization
- Probabilistic Relational Models
- Probabilistic Databases
- Sensor/Stream Data Management
- References

Bibliography

- [AFM'06] Periklis Andritsos and Ariel Fuxman and Renee J. Miller; Clean Answers over Dirty Databases; ICDE 2006
- [BBDW'05] Pedro Bizarro, Shivnath Babu, David J. DeWitt, Jennifer Widom: Content-Based Routing: Different Plans for Different Data. VLDB 2005: 757-768
- [BDHW'06] O. Benjelloun, A. Das Sarma, A. Halevy, and J. Widom. ULDBs: Databases with uncertainty and lineage. In VLDB, 2006.
- [BGP'92] D. Barbara, H. Garcia-Molina, and D. Porter; The management of probabilistic data; In IEEE Trans. of Knowledge Data Eng.; 1992.
- [BGR'01] Shivnath Babu, Minos N. Garofalakis, Rajeev Rastogi: SPARTAN: A Model-Based Semantic Compression System for Massive Data Tables. SIGMOD Conference 2001: 283-294
- [CDN'07] Surajit Chaudhuri and Gautam Das and Vivek Narasayya; "Optimized stratified sampling for approximate query processing"; TODS 2007.
- [CP'03] Christopher Crick and Avi Pfeffer; Loopy Belief Propagation as a Basis for Communication in Sensor Networks; UAI 2003.
- [dGK'00] Marie desJardins, Lise Getoor, Daphne Koller: Using Feature Hierarchies in Bayesian Network Learning. SARA 2000: 260-270
- [DFG'05] Arnaud Doucet, Nando de Freitas, and Neil Gordon. Sequential Monte Carlo methods in practice. Springer, 2005.
- [DGHM'05] Amol Deshpande, Carlos Guestrin, Wei Hong, Samuel Madden: Exploiting Correlated Attributes in Acquisitional Query Processing. ICDE 2005: 143-154
- [DGMHH'04] Amol Deshpande, Carlos Guestrin, Samuel Madden, Joseph M. Hellerstein, Wei Hong: Model-Driven Data Acquisition in Sensor Networks. VLDB 2004: 588-599

Bibliography

- [DGMHH'04] Amol Deshpande, Carlos Guestrin, Samuel Madden, Joseph M. Hellerstein, Wei Hong: Model-Driven Data Acquisition in Sensor Networks. VLDB 2004: 588-599
- [DGR'01] Amol Deshpande, Minos N. Garofalakis, Rajeev Rastogi: Independence is Good: Dependency-Based Histogram Synopses for High-Dimensional Data. SIGMOD Conference 2001: 199-210
- [DS'07] Nilesh Dalvi and Dan Suciu; Management of Probabilistic Data: Foundations and Challenges; PODS 2007.
- [FGKP'99] Nir Friedman, Lise Getoor, Daphne Koller, Avi Pfeffer; Learning Probabilistic Relational Models; IJCAI 1999: 1300-1309.
- [FT'97] N. Fuhr and T. Rolleke; A probabilistic relational algebra for the integration of information retrieval and database systems; ACM Trans. on Info. Syst.; 1997
- [G'06] Lise Getoor; An Introduction to Probabilistic Graphical Models for Relational Data; IEEE Data Engineering Bulletin; March 2006.
- [G'98] Zoubin Ghahramani. Learning dynamic Bayesian networks. Lecture Notes in Computer Science, 1387, 1998.
- [GGFKT'02] Lise Getoor, Nir Friedman, Daphne Koller, Benjamin Taskar: Learning Probabilistic Models of Link Structure. Journal of Machine Learning Research 3: 679-707 (2002)
- [GKTD'00] D.Gunopulos, G.Kollios, V.J.Tsotras and C.Domeniconi. Approximating Multi-Dimensional Aggregate Range Queries Over Real Attributes. SIGMOD 2000.
- [GTK'01] Lise Getoor, Benjamin Taskar, Daphne Koller: Selectivity Estimation using Probabilistic Models. SIGMOD Conference 2001: 461-472
- [IC'93] Y. E.Ioannidis and S.Christodoulakis. "Optimal Histograms for Limiting Worst-Case Error Propagation in the Size of Join Results"; TODS 1993.

Bibliography

- [IL'84] T. Imielinski and W. Lipski, Jr; Incomplete information in relational databases; Journal of the ACM; 1984.
- [JCW'04] Jain, E. Change, and Y. Wang. Adaptive stream resource management using kalman filters. In SIGMOD, 2004.
- [KG'06] Daniel Kifer and Johannes Gehrke; Injecting utility into anonymized datasets; SIGMOD 2006.
- [LWV'03] Lipyeow Lim, Min Wang, Jeffrey Scott Vitter: SASH: A Self-Adaptive Histogram Set for Dynamically Changing Workloads. VLDB 2003: 369-380.
- [M'01] Kevin Murphy. The Bayes net toolbox for matlab. Computing Science and Statistics, 33, 2001.
- [M'02] Kevin Murphy. Dynamic Bayesian Networks: Representation, Inference and Learnig. PhD thesis, UC Berkeley, 2002.
- [MP'01] V. Mihajlovic and M. Petkovic. Dynamic bayesian networks: A state of the art. University of Twente Document Repository 2001.
- [MVW'98] Y. Matias, J. S. Vitter, and M. Wang. Wavelet-Based Histograms for Selectivity Estimation. SIGMOD 1998.
- [ND'04] J. Neville and D. Jensen; Dependency Networks for Relational Data; ICDM 2004.
- [P+'04] Matthai Philipose et al; Inferring Activities from Interactions with Objects; IEEE Pervasive Computing, October 2004
- [PGM'05] Mark A. Paskin, Carlos Guestrin, Jim McFadden: A robust architecture for distributed inference in sensor networks. IPSN 2005.

Bibliography

- [PLFK'03] D. Patterson, L. Liao, D. Fox, and H. Kautz. Inferring high level behavior from low level sensors. In UBICOMP, 2003.
- [R'89] L Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. 1989.
- [SBEMY'06] Adam Silberstein, Rebecca Braynard, Carla Ellis, Kamesh Munagala and Jun Yang. A Sampling-Based Approach to Optimizing Top-k Queries in Sensor Networks. ICDE 2006.
- [SD'07] Prithviraj Sen, Amol Deshpande: Representing and Querying Correlated Tuples in Probabilistic Databases. ICDE 2007: 596-605
- [SKGGM'05] Vipul Singhvi, Andreas Krause, Carlos Guestrin, James H. Garrett Jr., H. Scott Matthews: Intelligent light control using sensor networks. SenSys 2005: 218-229
- [TAK'02] Benjamin Taskar, Pieter Abbeel, Daphne Koller; Discriminative Probabilistic Models for Relational Data; UAI 2002: 485-492
- [W'05] J. Widom. Trio: A system for integrated management of data, accuracy, and lineage. In CIDR, 2005.
- [Y'00] Jie Ying. A hidden markov model-based algorithm for fault diagnosis with partial and imperfect tests. IEEE Trans. on Systems, Man, and Cybernetics, Part C, 2000.

Part III: Graphical models for Information extraction and data integration

Graphical models for Information extraction and data integration

Sunita Sarawagi
IIT Bombay
http://www.cse.iitb.ac.in/~sunita

Information Extraction (IE) & Integration

The Extraction task: Given,

- E: a set of structured elements
- S: unstructured source S

extract all instances of E from S

The integration task: Given

– database of existing inter-linked entities

Resolve which entities are the same.

- Many versions involving many source types
- Actively researched in varied communities
- Several tools and techniques
- Several commercial applications

IE from free format text

- Classical Named Entity Recognition
  - Extract person, location, organization names

According to Robert Callahan, president of Eastern's flight attendants union, the past practice of Eastern's parent, Houston-based Texas Air Corp., has involved ultimatums to unions to accept the carrier's terms

- Several applications
  - –News tracking
    - – Monitor events
  - –Bio-informatics
    - – Protein and Gene names from publications
  - –Customer care
    - •Part number, problem description from emails in help centers

Text segmentation

House number | Building | Road | City | State | Zip

4089 | Whispering Pines | Nobel Drive | San Diego | CA | 92122

Author | Year | Title | Journal | Volume | Page

P.P.Wangikar, T.P. Graycar, D.A. Estell, D.S. Clark, J.S. Dordick (1993) Protein and Solvent Engineering of Subtilising BPN' in Nearly Anhydrous Organic Media J.Amer. Chem. Soc. 115, 12231-12237.

Information Extraction (IE)

- Many different uses
  - Disease outbreaks from news articles
  - Addresses/Qualifications from resumes for HR DBs
  - Titles/Authors/Venue/Year from citations
  - Room attributes from hotel websites
- Many approaches
  - Rules-based -------- Statistical learners
- Varying levels of difficulty
  - Wrappers for machine generated pages
  - ..
  - Fact extraction from speech transcripts

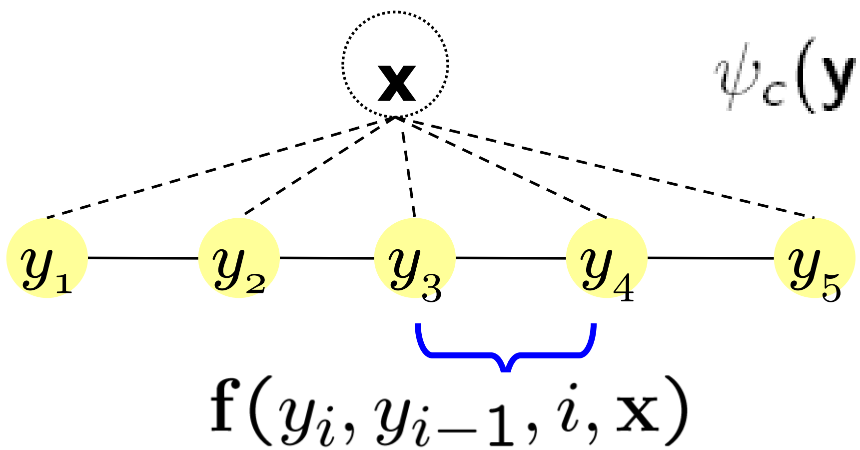
Graphical models in Extraction & Dedup

- State of the art: Conditional Random Fields
- IE Models
  - Basic IE model (Chain)
  - IE with collective labeling of repeated words
- De-duplication models
  - Basic pair-wise model
  - Collective de-duplication of relational data
  - Collective de-duplication of multiple networked entities

Conditional Random Fields

Special undirected graphical model

1. Conditional distribution Pr (**y|x**) where **y** = $y_1 y_2 \ldots y_n$
2. Graph: over the interdependent components of **y**
3. Potentials: weighted sum of features over **x**



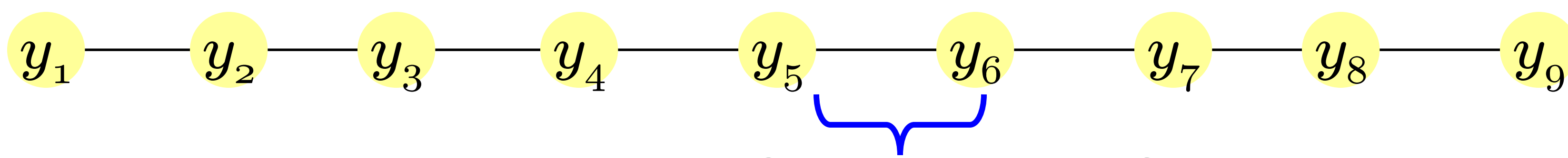
$$\psi_c(\mathbf{y}_c) = \exp(\sum_{i=1}^{K} w_k f_k(\mathbf{y}_c, \mathbf{x}, c))$$

[Lafferty et al 2001]

Chain model

My review of Fermat's last theorem by S. Singh

t	1	2	3	4	5	6	7	8	9
x	My	review	of	Fermat's	last	theorem	by	S.	Singh
y	**Other**	**Other**	**Other**	**Title**	**Title**	**Title**	**other**	**Author**	**Author**



$$\mathbf{f}(y_i, y_{i-1}, i, \mathbf{x})$$

Features

- Feature vector for each position

$$\mathbf{f}(y_i, \mathbf{x}, i, y_{i-1})$$

User provided

i-th label

Word i & neighbors

previous label

- Examples

$$f_2(y_i, \mathbf{x}, i, y_{i-1}) = 1 \text{ if } y_i \text{ is Person } \& \ x_i \text{ is Douglas}$$

$$f_3(y_i, \mathbf{x}, i, y_{i-1}) = 1 \text{ if } y_i \text{ is Person } \& \ y_{i-1} \text{ is Other}$$

- Parameters: weight for each feature (vector)

$$\mathbf{W} = W_1 W_2 \ldots W_{|\mathbf{f}|}$$

Machine learnt

Features in typical extraction tasks

- Words
- Orthographic word properties
  - Capitalized? Digit? Ends-with-dot?
- Part of speech
  - Noun?
- Match in a dictionary
  - Appears in a dictionary of people names?
  - Appears in a list of stop-words?

- Fire these for each label and
  - The token,
  - W tokens to the left or right, or
  - Concatenation of tokens.

Examples: features with weights (publications).

#	Name	Person	Location	Other
1	x_i is noun	1.2	1.2	-0.5
4	"at" in $\{x_{i-1}, x_{i-2}\}$	-0.3	3	0.2
7	$x_{i-1}x_i$ in people names dictionary	3	-0.4	0
10	x_{i-1} is single caps & dot.	2.1	-1.0	-0.1
13	y_{i-1} is Location	-1.5	0.3	1.0
.	..			
.				
100000	..			

A large number

Typical numbers

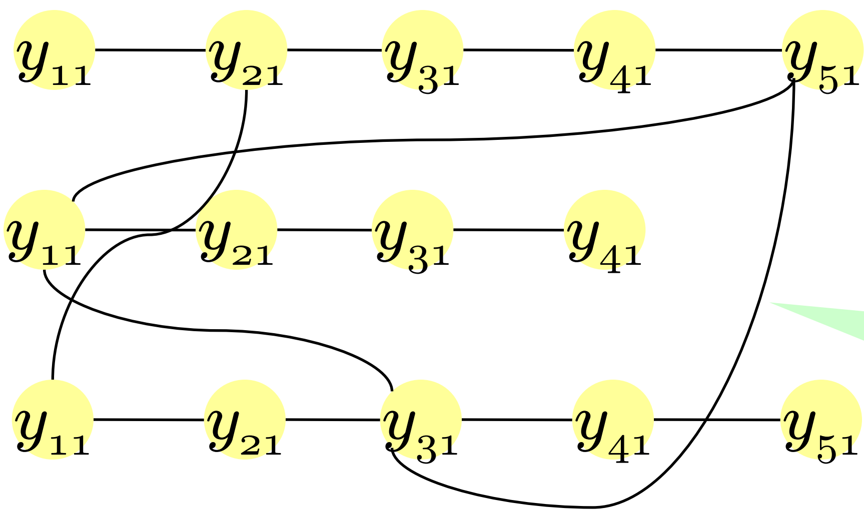
- Seminars announcements (CMU):
  - speaker, location, timings
  - SVMs for start-end boundaries
  - 250 training examples
  - F1: 85% speaker, location, 92% timings (Finn & Kushmerick ’04)
- Jobs postings in news groups
  - 17 fields: title, location, company,language, etc
  - 150 training examples
  - F1: 84% overall (LP2) (Lavelli et al 04)

Graphical models in Extraction & Dedup

- State of the art: Conditional Random Fields
- IE Models
  - Basic IE model (Chain)
  - IE with collective labeling of repeated words
- De-duplication models
  - Basic pair-wise model
  - Collective de-duplication of relational data
  - Collective de-duplication of multiple networked entities

Collective labeling

- Y has character.
- Mr. X lives in Y.
- X buys Y Times daily.



Associative potentials
$\phi_e(i,i) > \phi_e(i,j)$

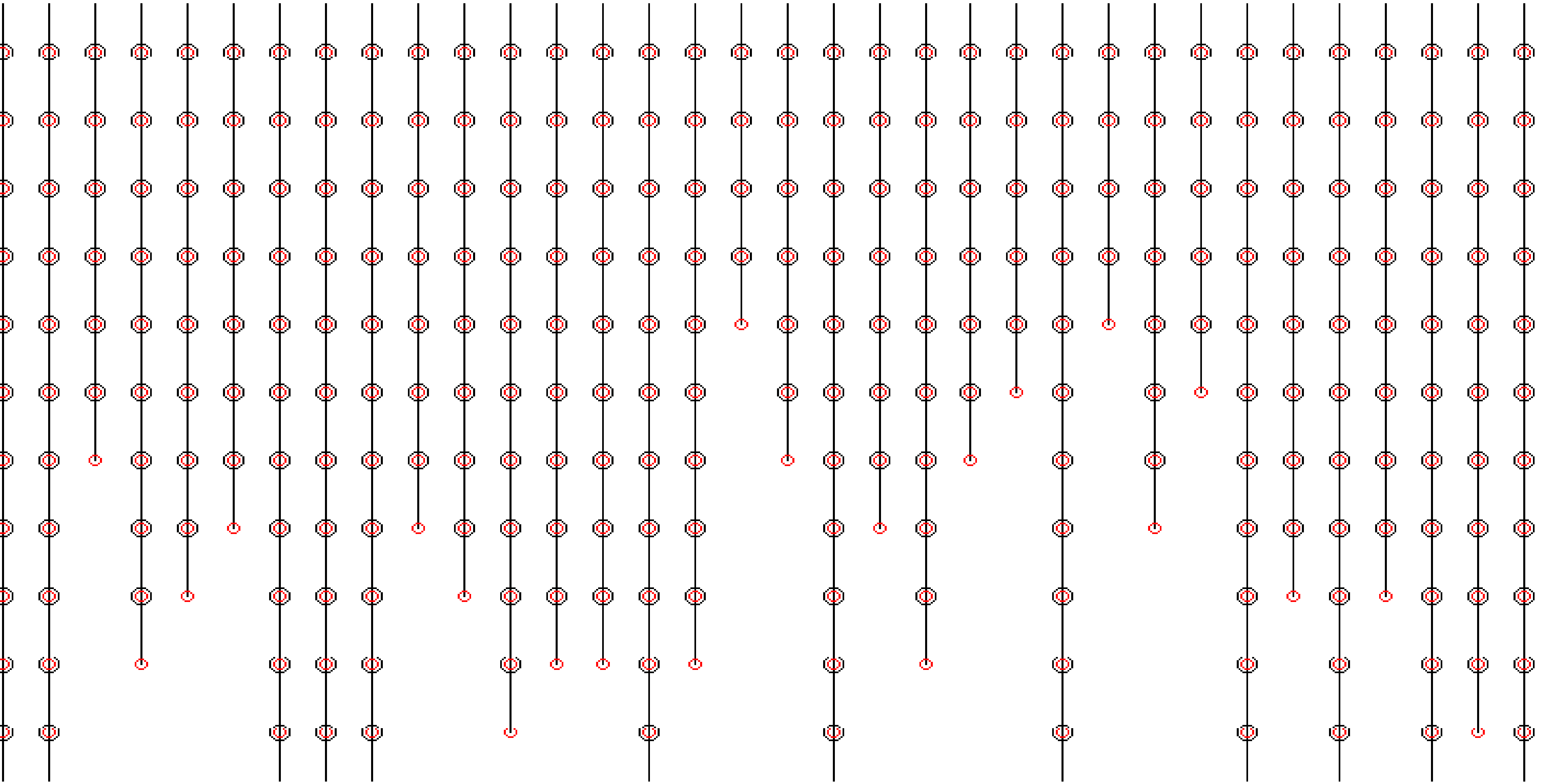
Other applications of associative potentials

Social network analysis: "friends of smokers are smokers"

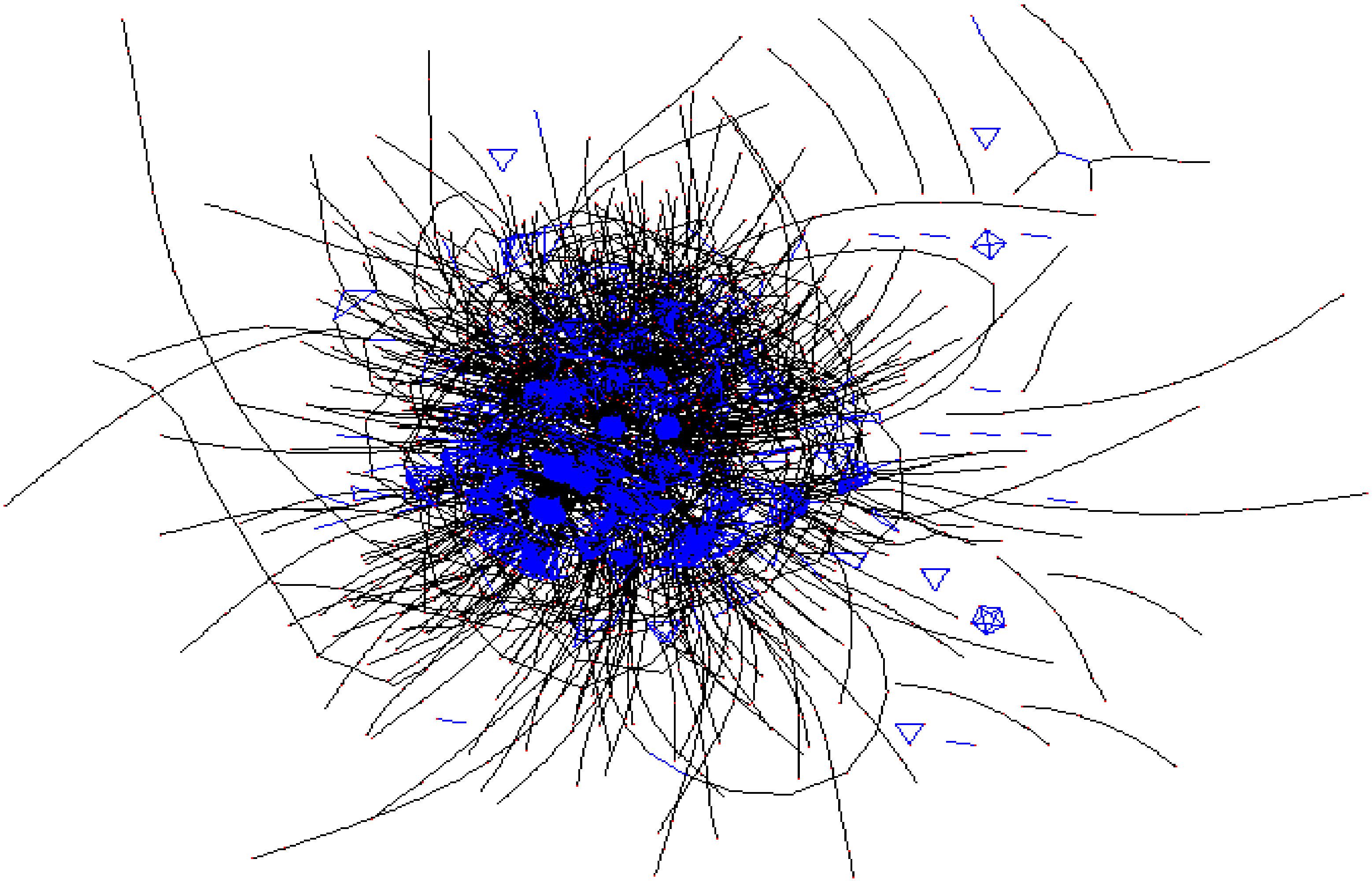
Image segmentation: "nearby pixels get the same label"

Spam detection: "spam pages are pointed to by spams"

Starting graphs (..of an extraction task from addresses)



Graph after collective edges



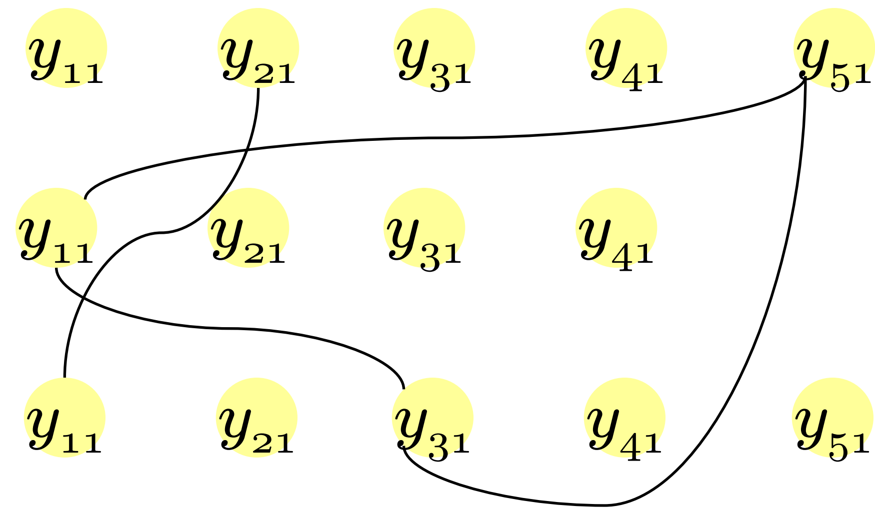
Algorithms for collective inference

- Exact: intractable
- Approximate
  - Loopy Belief Propagation
    - Message passing (MP) on edges of the graph
    - [Bunescu & Mooney ’04], [Sutton & McCallum ‘04]
  - Gibbs Sampling
    - [Finkel & Manning. ’05]
  - Greedy local search (ICM)
    - [Lu & Getoor’03]
    - A special two-pass variant: [Krishnan & Manning ’06]

- Generic techniques, no guarantees

Associative Markov Networks

- Graph with only associative edge potentials and node potentials

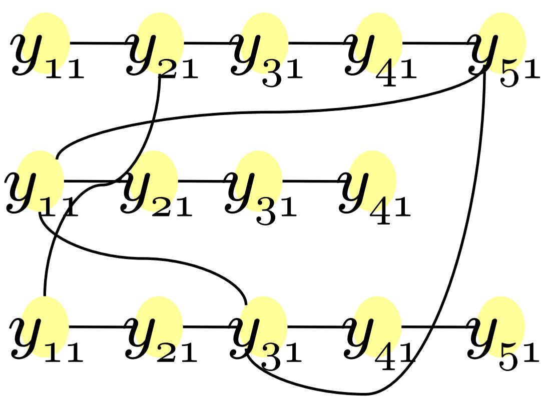


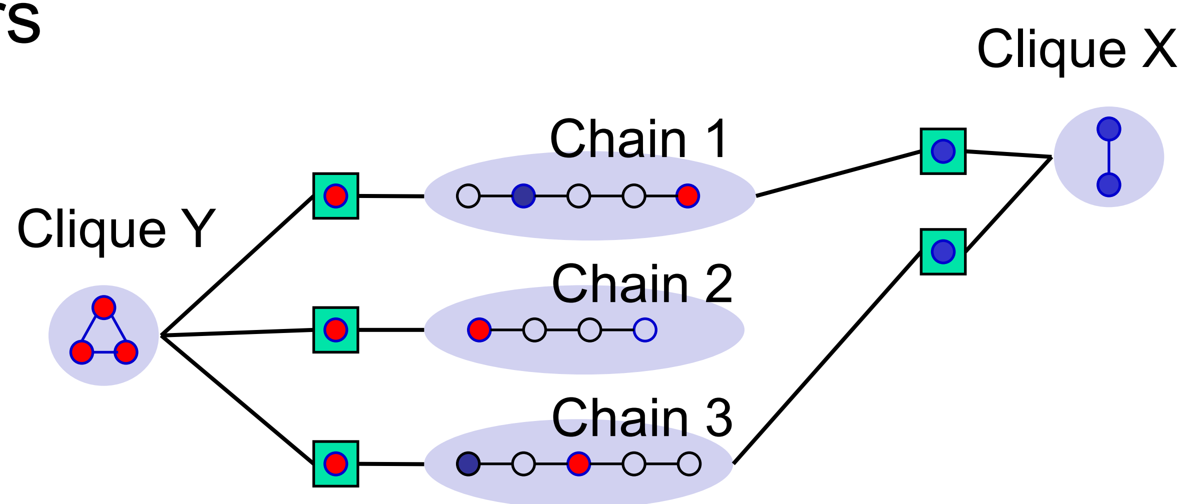
- Optimal for m=2. ½ approximation for m > 2
  - Min-cut with α-expansion (Boykov '99)
  - LP-based metric labeling algorithms (Klienberg & Tardos '02)
  - BP with TRW-S message schedules (Kolmogorov & Wainwright, '05)

Not directly usable Slow Worse guarantees

Generalized Belief propagation

BP on clusters of cliques and chains with single node separators





- Basic MP step: Compute max-marginals for a separator node ➔ MAP for each label of the node.
- MAP algorithms for chains ➔ easy and efficient.
- MAP algorithms for cliques➔ combinatorial algorithms can be used for this. (Gupta et al 2007)

Graphical models in Extraction & Dedup

- State of the art: Conditional Random Fields
- IE Models
  - Basic IE model (Chain)
  - IE with collective labeling of repeated words
- De-duplication models
  - Basic pair-wise model
  - Collective de-duplication of relational data
  - Collective de-duplication of multiple networked entities

Basic dedup problem

- Given a pair of records x_1, x_2, predict "y" to denote if they are the same or not.

X_1 `Johnson Laird, Philip N. (1983). Mental models. Cambridge, Mass.: Harvard University Press.`

X_2 `P. N. Johnson-Laird. Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness. Cambridge University Press, 1983`

- CRF: $Pr(y| x_1, x_2)$ where
  - Features: list of similarity functions between record pairs.
  - Graph: trivial single node graph

Multi Attribute Similarity

Record 1 D
Record 2

Record 1 N
Record 3

Record 4 D
Record 5

f_1	f_2	...f_n	
1.0	0.4	... 0.2	1
0.0	0.1	... 0.3	0
0.3	0.4	... 0.4	1

Mapped examples

Unlabeled list

Record 6
Record 7
Record 8
Record 9
Record 10
Record 11

0.0	0.1	... 0.3	?
1.0	0.4	... 0.2	?
0.6	0.2	... 0.5	?
0.7	0.1	... 0.6	?
0.3	0.4	... 0.4	?
0.0	0.1	... 0.1	?
0.3	0.8	... 0.1	?
0.6	0.1	... 0.5	?

All-Ngrams*0.4 + AuthorTitleNgram*0.2 – 0.3YearDifference + 1.0*AuthorEditDist + 0.2*PageMatch – 3 > 0

Learners:

- Support Vector Machines (SVM)
- Logistic regression,
- Linear regression,
- Perceptron

0.6	0.2	... 0.5	0
0.7	0.1	... 0.6	0
0.3	0.4	... 0.4	1
0.0	0.1	... 0.1	0
0.3	0.8	... 0.1	1
0.6	0.1	... 0.5	1

Graphical models in Extraction & Dedup

- State of the art: Conditional Random Fields
- IE Models
  - Basic IE model (Chain)
  - IE with collective labeling of repeated words
- De-duplication models
  - Basic pair-wise model
  - Collective de-duplication of relational data
  - Collective de-duplication of set-oriented data

De-duplication of relational records

Collectively de-duplicate entities and its many attributes

Record	Title a^1	Author a^2	Venue a^3
b_1	"Record Linkage using CRFs"	"Linda Stewart"	"KDD-2003"
b_2	"Record Linkage using CRFs"	"Linda Stewart"	"9th SIGKDD"
b_3	"Learning Boolean Formulas"	"Bill Johnson"	"KDD-2003"
b_4	"Learning of Boolean Expressions"	"William Johnson"	"9th SIGKDD"

Associate variables for predictions

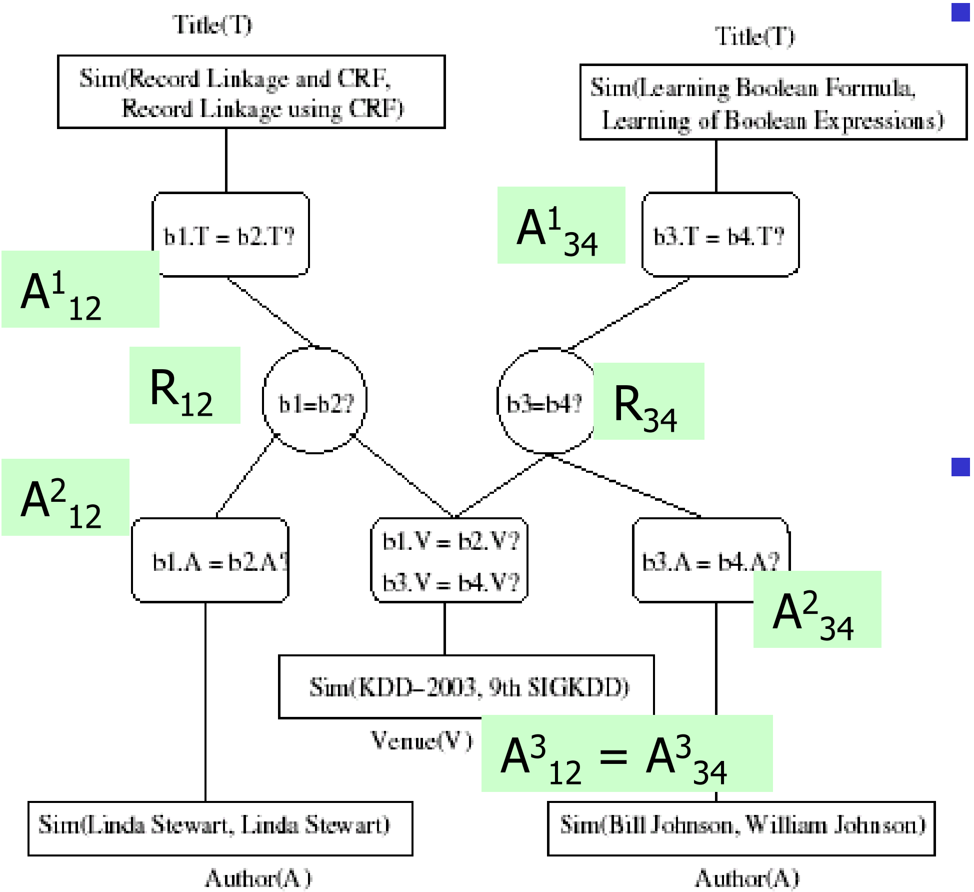
for each attribute k

each record pair (i,j) A^k_{ij}

for each record pair R_{ij}

from Parag & Domingos 2005

Graphical model



Potentials

- Independent scores
  - $s_k(A^k,a_i,a_j)$ Attribute-level
    - Any classifier on various text similarities of attribute pairs
  - $s(R,b_i,b_j)$ Record-level
    - Any classifier on various similarities of all k attribute pairs
- Dependency scores
  - $d_k(A^k, R)$: record pair, attribute pair

	0	1
0	4	2
1	1	7

Joint de-duplication steps

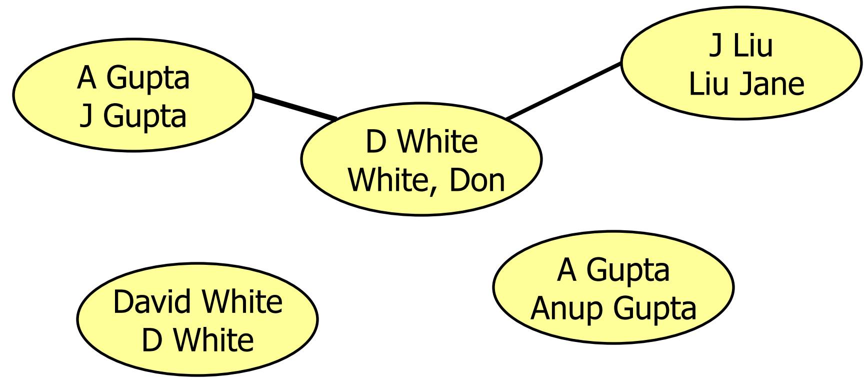
- Jointly pick 0/1 labels for all record pairs Rij and all K attribute pairs A^k_{ij} to maximize sum of potentials
- Typical graphical model inference problem
- Efficient algorithm possible because of special forms of potentials
  - dependency scores associative
  - dk(1,1) + dk(0,0) >= dk(1,0)+dk(0,1)

Graphical models in Extraction & Dedup

- State of the art: Conditional Random Fields
- IE Models
  - Basic IE model (Chain)
  - IE with collective labeling of repeated words
- De-duplication models
  - Basic pair-wise model
  - Collective de-duplication of relational data
  - Collective de-duplication of set-oriented data

Collective linkage: set-oriented data

P1	D White, J Liu, A Gupta
P2	Liu, Jane & J Gupta & White, Don
P3	Anup Gupta
P4	David White



Scoring functions

- $S(A_{ij})$ Attribute-level
  - Text similarity
- $S(A_{ij}, N_{ij})$ Dependency with labels of co-author set
  - Fraction of co-author set assigned label 1.
- Score: $\alpha\, s(A_{ij}) + (1-\alpha)\, s(A_{ij}, N_{ij})$

Inference Algorithm

- Exact inference hard
- MCMC algorithm in (Bhattacharya and Getoor, 2007)

Concluding remarks

- Graphical models provide a unified and flexible modeling of many extraction and integration tasks
- Much work is still needed in converting these to methods of choice in commercial systems
  - Scalable algorithms
  - Skillful integration of manual rules with statistical methods
  - Feedback on when the statistical method failes
  - Robust feature design so as to not overfit on the training data.