Graphical models

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

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 marginals (sum or max) over the joint distribution.

Examples of Joint Distributions So far

- Naive Bayes: $P(x_1, \ldots x_d | y)$, d is large. Assume conditional independence.
- Multivariate Gaussian
- Recurrent Neural Networks for Sequence labeling and prediction

Example

- Variables are attributes are people.

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 fraction 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: conditional independencies
  - income $\not\perp\!\!\!\perp$ age, but income $\perp\!\!\!\perp$ age $|$ experience
  - experience $\perp\!\!\!\perp$ degree, but experience $\not\perp\!\!\!\perp$ degree $|$ income

Graphical models make explicit an efficient joint distribution from these independencies

More examples of CIs

- The grades of a student in various courses are correlated but they become CI given attributes of the student (hard-working, intelligent, etc?)
- Health symptoms of a person may be correlated but are CI given the latent disease.
- Words in a document are correlated, but may become CI given the topic.
- Pixel color in an image become CI of distant pixels given near-by pixels.

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.

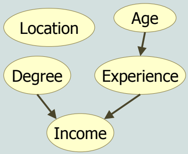
Representation

Structure of a graphical model: Graph + Potential

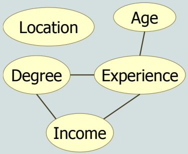
Graph

- Nodes: variables $\mathbf{x} = x_1, \ldots x_n$
  - Continuous: Sensor temperatures, income
  - Discrete: Degree (one of Bachelors, Masters, PhD), Levels of age, Labels of words
- 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

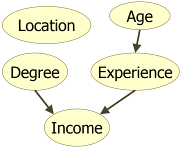
- Graph G: directed acyclic
  - Parents of a node: $\text{Pa}(x_i)$ = set of nodes in G pointing to x_i
- Potentials: defined at each node in terms of its parents.

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

- 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

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

Conditional Independencies

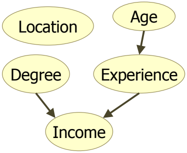
- Given three sets of variables X, Y, Z, set X is conditionally independent of Y given Z ($X \perp\!\!\!\perp Y|Z$) iff

$$\Pr(X|Y,Z) = \Pr(X|Z)$$

- Local conditional independencies in BN: for each x_i

$$x_i \perp\!\!\!\perp ND(x_i)|Pa(x_i)$$

- $L \perp\!\!\!\perp E, D, A, I$
- $A \perp\!\!\!\perp L, D$
- $E \perp\!\!\!\perp L, D|A$
- $I \perp\!\!\!\perp A|E, D$



CIs and Fractorization

Theorem

Given a distribution $P(x_1, \ldots, x_n)$ and a DAG G, if P satisfies Local-CI induced by G, then P can be factorized as per the graph.
Local-CI$(P, G) \implies$ Factorize(P, G)

Proof.

- $x_1, x_2, \ldots, x_n$ topographically ordered (parents before children) in G.
- Local CI(P, G): $P(x_i|x_1, \ldots, x_{i-1}) = P(x_i|Pa_G(x_i))$
- Chain rule:
  $P(x_1, \ldots, x_n) = \prod_i P(x_i|x_1, \ldots, x_{i-1}) = \prod_i P(x_i|Pa_G(x_i))$
- $\implies$ Factorize(P, G)

$\square$

CIs and Fractorization

Theorem

Given a distribution $P(x_1, \ldots, x_n)$ and a DAG G, if P can be factorized as per G then P satisfies Local-CI induced by G.
Factorize(P, G) $\implies$ Local-CI(P, G)

Proof skipped. (Refer book.)

Drawing a BN starting from a distribution

Given a distribution $P(x_1, \ldots, x_n)$ to which we can ask any CI of the form "Is $X \perp\!\!\!\perp Y|Z$?" and get a yes, no answer.
Goal: Draw a minimal, correct BN G to represent P.

Why minimal

Theorem

G constructed by the above algorithm is minimal, that is, we cannot remove any edge from the BN while maintaining the correctness of the BN for P

Proof.

By construction. A subset of ND of each x_i were available when parent of U were chosen minimally. □

Why Correct

Theorem

G constructed by the above algorithm is correct, that is, the local-CIs induced by G hold in P

Proof.

The construction process makes sure that the factorization property holds. Since factorization implies local-CIs, the constructed BN satisfied the local-CIs of P $\square$

Order is important

Examples of CIs that hold in BN but not covered by local-CI

Global CIs in a BN

Three sets of variables X, Y, Z. If Z d-separates X from Y in BN then, $X \perp\!\!\!\perp Y | Z$.
In a directed graph H, Z d-separates X from Y if all paths P from any X to Y is blocked by Z.
A path P is blocked by Z when

1. $x_1 \rightarrow x_2 \rightarrow \ldots x_k$ and $x_i \in Z$
2. $x_1 \leftarrow x_2 \leftarrow \ldots x_k$ and $x_i \in Z$
3. $x_1 \ldots \leftarrow x_i \rightarrow \ldots x_k$ and $x_i \in Z$
4. $x_1 \ldots \rightarrow x_i \leftarrow \ldots x_k$ and $x_i \not\in Z$ and $Desc(x_i) \not\in Z$

Theorem

The d-separation test identifies the complete *set of conditional independencies that hold in* all *distributions that conform to a given Bayesian network.*

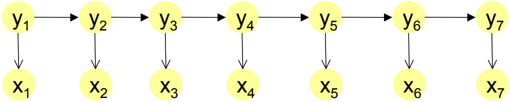
Global CIs Examples

Global CIs and Local-CIs

In a BN, the set of CIs combined with the axioms of probability can be used to derive the Global-CIs.
Proof is long but easy to understand. Sketch of a proof available in the supplementary.

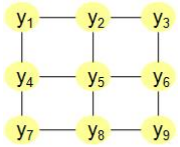
Popular Bayesian networks

- 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
- Topic models for text data
  1. Principled mechanism to categorize multi-labeled text documents while incorporating priors in a flexible generative framework
  2. Application: news tracking
- QMR (Quick Medical Reference) system
- PRMs: Probabilistic relational networks:

Undirected graphical models

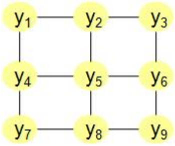


- Graph G: arbitrary undirected graph
- Useful when variables interact symmetrically, no natural parent-child relationship
- Example: labeling pixels of an image.
- Potentials $\psi_C(\mathbf{y}_C)$ defined on arbitrary cliques C of G.
- $\psi_C(\mathbf{y}_C)$: Any arbitrary non-negative value, cannot be interpreted as probability.
- Probability distribution

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

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

Example



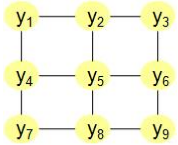
$y_i = 1$ (part of foreground), 0 otherwise.

- 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: $\Pr(y_1 \dots y_9) \propto \prod_{k=1}^{9} \psi_k(y_k) \prod_{(i,j) \in E(G)} \psi(y_i, y_j)$

Conditional independencies (CIs) in an undirected graphical model

Let $V = \{y_1, \ldots, y_n\}$.
Let distribution P be represented by an undirected graphical model G. If Z separates X and Y in G, then $X \perp\!\!\!\perp Y | Z$ in P.
The set of all such CIs are called Global-CI of the UGM.
Example:

1. $y_1 \perp\!\!\!\perp y_3, y_5 y_6, y_7, y_8, y_9 | y_2, y_4$
2. $y_1 \perp\!\!\!\perp y_3 | y_2, y_4, y_5, y_6, y_7, y_8, y_9$
3. $y_1, y_2, y_3 \perp\!\!\!\perp y_7, y_8, y_9 | y_4, y_5, y_6$



Factorization implies Global-CI

Theorem

Let G be a undirected graph over $V = x_1, \ldots, x_n$ nodes and $P(x_1, \ldots, x_n)$ be a distribution. If P is represented by G that is, if it can be factorized as per the cliques of G, then P will also satisfy the global-CIs of G

$\textit{Factorize}(P, G) \implies \textit{Global-CI}(P, G)$

Factorization implies Global-CI (Proof)

Available as proof of Theorem 4.1 in KF book.

Global-CI does not imply factorization.

(Taken from example 4.4 of KF book)
But global-CI does not imply factorization. Consider a distribution over 4 binary variables: $P(x_1, x_2, x_3, x_4)$
Let G be

$$\begin{tikzcd} x_1 \arrow[r, dash] \arrow[d, dash] & x_2 \arrow[d, dash] \\ x_4 \arrow[r, dash] & x_3 \end{tikzcd}$$

Let $P(x_1, x_2, x_3, x_4) = 1/8$ when x_1, x_2, x_3, x_4 takes values from this set $=\{0000,1000,1100,1110,1111,0111,0011,0001\}$. In all other cases it is zero. One can painfully check that all four globals CIs in the graph: e.g. $x_1 \perp\!\!\!\perp \{x_3\}|x_2, x_4$ etc hold in the graph.
Now let us look at factorization. The factors correspond to the edges in $\psi(x_1, x_2)$. Each of the four possible assignment of each factor will get a positive value. But that cannot represent the zero probability for cases like $x_1, x_2, x_3, x_4 = 0101$.

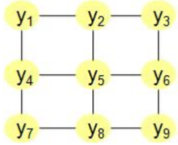
Other Conditional independencies (CIs) in an undirected graphical model

Let $V = \{y_1, \ldots, y_n\}$.

1. Local CI: $y_i \perp\!\!\!\perp V - ne(y_i) - \{y_i\} | ne(y_i)$
2. Pairwise CI: $y_i \perp\!\!\!\perp y_j | V - \{y_i, y_j\}$ if edge (y_i, y_j) does not exist.
3. Global CI: $X \perp\!\!\!\perp Y | Z$ if Z separates X and Y in the graph.

Equivalent when the distribution $P(x)$ is positive, that is $P(x) > 0, \quad \forall x$

1. $y_1 \perp\!\!\!\perp y_3, y_5 y_6, y_7, y_8, y_9 | y_2, y_4$
2. $y_1 \perp\!\!\!\perp y_3 | y_2, y_4, y_5, y_6, y_7, y_8, y_9$
3. $y_1, y_2, y_3 \perp\!\!\!\perp y_7, y_8, y_9 | y_4, y_5, y_6$



Relationship between Local-CI and Global-CI

Let G be a undirected graph over $V = x_1, \ldots, x_n$ nodes and $P(x_1, \ldots, x_n)$ be a distribution. If P satisfies Global-CIs of G, then P will also satisfy the local-CIs of G but the reverse is not always true. We will show this with an example.

Consider a distribution over 5 binary variables: $P(x_1, \ldots, x_5)$ where $x_1 = x_2$, $x_4 = x_5$ and $x_3 = x_2$ AND x_4.

Let G be

$$x_1 \text{ --- } x_2 \text{ --- } x_3 \text{ --- } x_4 \text{ --- } x_5$$

All 5 local CIs in the graph: e.g. $x_1 \perp\!\!\!\perp \{x_3, x_4, x_5\} | x_2$ etc hold in the graph.

However, the global CI: $x_2 \perp\!\!\!\perp x_4 | x_3$ does not hold.

Relationship between Local-CI and Pairwise-CI

Let G be a undirected graph over $V = x_1, \ldots, x_n$ nodes and $P(x_1, \ldots, x_n)$ be a distribution. If P satisfies Local-CIs of G, then P will also satisfy the pairwise-CIs of G but the reverse is not always true. We will show this with an example.
Consider a distribution over 3 binary variables: $P(x_1, x_2, x_3)$ where $x_1 = x_2 = x_3$. That is, $P(x_1, x_2, x_3) = 1/2$ when all three are equal and 0 otherwise.
Let G be

$$x_1 \text{ --- } x_2 \qquad\qquad x_3$$

All 2 pairwise CIs in the graph: e.g. $x_1 \perp\!\!\!\perp \{x_3\}|x_2$ and $x_2 \perp\!\!\!\perp \{x_3\}|x_1$ hold in the graph.
However, the local CI: $x_1 \perp\!\!\!\perp x_3$ does not hold.

Factorization and CIs

Theorem

(Hammerseley Clifford Theorem) If a positive distribution $P(x_1, \ldots, x_n)$ confirms to the pairwise CIs of a UDGM G, then it can be factorized as per the cliques C of G as

$$P(x_1, \ldots, x_n) \propto \prod_{C \in G} \psi_C(\mathbf{y}_C)$$

Proof.

Theorem 4.8 of KF book (partially) $\square$

Summary

Let P be a distribution and H be an undirected graph of the same set of nodes.

Factorize(P, H) $\implies$ Global-CI(P, H) $\implies$ Local-CI(P, H) $\implies$ Pairwise-CI(P, H)

But only for positive distributions

Pairwise-CI(P, H) $\implies$ Factorize(P, H)

Constructing an UGM from a positive distribution

Given a positive distribution $P(x_1, \ldots, x_n)$ to which we can ask any CI of the form "Is $X \perp\!\!\!\perp Y|Z$?" and get a yes, no answer.

Goal: Draw a minimal, correct UGM G to represent P.

Two options: (1) Using pairwise CI (2) Using Local CI.

Constructing an UGM from a positive distribution using Local-CI

Definition: The Markov Blanket of a variable x_i, MB(x_i) is the smallest subset of variables V that makes x_i CI of others given the Markov blanket.

$$x_i \perp\!\!\!\perp V - MB(x_i) | MB(x_i)$$

The MB of a variable is always unique for a positive distribution.

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
- Social networks
- Bio-informatics: annotating active sites in a protein molecules.

Conditional Random Fields (CRFs)

Used to represent conditional distribution $P(\mathbf{y}|\mathbf{x})$ where $\mathbf{y} = y_1, \ldots, y_n$ forms an undirected graphical model.
The potentials are defined over subset of y variables, and the whole of $\mathbf{x}$.

$$\Pr(y_1, \ldots, y_n|\mathbf{x}, \theta) = \frac{\prod_C \psi_c(\mathbf{y}_c, \mathbf{x}, \theta)}{Z_\theta(\mathbf{x})} \quad = \frac{1}{Z_\theta(\mathbf{x})} \exp(\sum_c F_\theta(\mathbf{y}_c, c, \mathbf{x}))$$

where $Z_\theta(\mathbf{x}) = \sum_{\mathbf{y}'} \exp(\sum_c F_\theta(\mathbf{y}'_c, c, \mathbf{x}))$
clique potential $\psi_c(\mathbf{y}_c, \mathbf{x}) = \exp(F_\theta(\mathbf{y}_c, c, \mathbf{x}))$

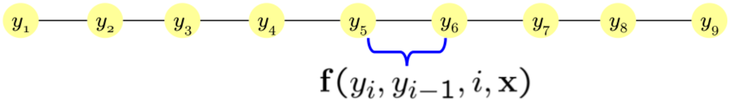
Potentials in CRFs

- Log-linear model over user-defined features. E.g. CRFs, Maxent models, etc.
  Let K be number of features. Denote a feature as $f_k(\mathbf{y}_c, c, \mathbf{x})$.
  Then,
  $$F_\theta(\mathbf{y}_c, c, \mathbf{x}) = \sum_{k=1}^{K} \theta_k f_k(\mathbf{y}_c, c, \mathbf{x})$$
- Arbitrary function, e.g. a neural network that takes as input $\mathbf{y}_c, c, \mathbf{x}$ and transforms them possibly non-linearly into a real value. θ are the parameters of the network.

Example: Named Entity Recognition

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**



Features decompose over adjacent labels.

$$\mathbf{f}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{|\mathbf{x}|} \mathbf{f}(y_i, y_{i-1}, i, \mathbf{x})$$

Named Entity Recognition: 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

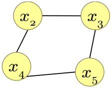
- Examples

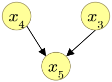
$$f_2(y_i, \mathbf{x}, i, y_{i-1}) = 1 \text{ if } y_i \text{ is Person } \& \ x_i \text{ is Douglas}$$

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

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.

Equivalent BNs

Two BN DAGs are said to be equivalent if they express the same set of CIs. (Examples)

Theorem

Two BNs G_1, G_2 are equivalent iff they have the same skeleton and the same set of immoralities. (An immorality is a structure of the form $x \rightarrow y \leftarrow z$ with no edge between x and z)

Converting BN to MRFs

Efficient: Using the Markov Blanket algorithm.

For which BN can we create perfect MRFs?

Converting MRFs to BNs

Which MRFs have perfect BNs

Chordal or triangulated graphs

A graph is chordal if it has no minimal cycle of length ≥ 4.

Theorem

A MRF can be converted perfectly into a BN iff it is chordal.

Proof.

Theorems 4.11 and 4.13 of KF book □

Algorithm for constructing perfect BNs from chordal MRFs to be discussed later.

BN and Chordality

A BN with a minimal undirected cycle of length ≥ 4 must have an immorality. A BN without any immorality is always chordal.

Inference queries

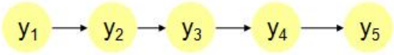
1. *Marginal probability queries over a small subset of variables:*
   - Find Pr(Income='High & Degree='PhD')
   - Find Pr(pixel $y_9 = 1$)

$$\begin{aligned}\Pr(x_1) &= \sum_{x_2 \ldots x_n} \Pr(x_1 \ldots x_n) \\ &= \sum_{x_2=1}^{m} \ldots \sum_{x_n=1}^{m} \Pr(x_1 \ldots x_n)\end{aligned}$$

   Brute-force requires $O(m^{n-1})$ time.
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

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

Exact inference on chains

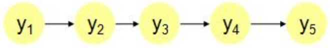
- Given,
  - Graph 

  - Potentials: $\psi_i(y_i, y_{i+1})$
  - $Pr(y_1, \ldots y_n) = \prod_i \psi_i(y_i, y_{i+1}), Pr(y_1)$
- 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')

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

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

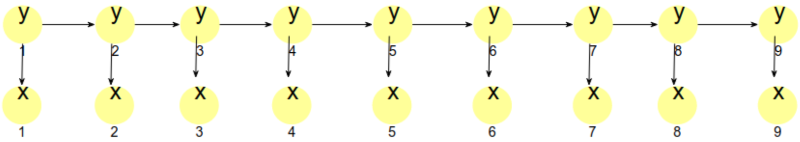
$$\mathbf{x}^* = \text{argmax}_{x_1 \dots x_m} \Pr(x_1 \dots x_n | \mathbf{x}_e)$$

Easy to add evidence, just change the potential.

Case study: HMMs for Information Extraction

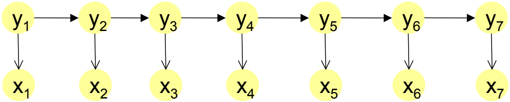
My review of Fermat's last theorem by S. Singh

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**



Inference in HMMs

- Given,
  - Graph

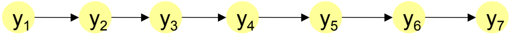


  - Potentials: $\Pr(y_i|y_{i-1}), \Pr(x_i|y_i)$
  - Evidence variables: $\mathbf{x} = x_1 \ldots x_n = o_1 \ldots o_n$.
- Find most likely values of the hidden state variables.

  $\mathbf{y} = y_1 \ldots y_n$

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

- 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

The Viterbi algorithm

Let observations x_t take one of k possible values, states y_t take one of m possible value.

Given n observations: $o_1, \ldots, o_n$
Given Potentials $\Pr(y_t|y_{t-1}) = P(y|y')$ (Table with m^2 values), $\Pr(x_t|y_t) = P(x|y)$ (Table with mk values), $\Pr(y_1) = P(y)$ start probabilities (Table with m values.)
Find $\max_{\mathbf{y}} \Pr(\mathbf{y}|\mathbf{x} = \mathbf{o})$
$B_n[y] = 1 \quad y \in [1, \ldots, m]$
for $t = n \ldots 2$ **do**
  $\psi(y, y') = P(y|y')P(x_t = o_t|y)$
  $B_{t-1}[y'] = \max_{y=1}^{n} \psi(y, y')B_t[y]$
end for
Return $\max_y B_1[y]P(y)P(x_t = o_t|y)$

Time taken: $O(nm^2)$

Numerical Example

$P(y|y') =$

y'	$P(y=0|y')$	$P(y=1|y')$
0	0.9	0.1
1	0.2	0.8

$P(x|y) =$

y	$P(x=0|y)$	$P(x=1|y)$
0	0.7	0.3
1	0.6	0.4

$P(y=1) = 0.5$

Observation $[x_0, x_1, x_2] = [0, 0, 0]$

Variable elimination on general graphs

- Given, arbitrary sets of potentials $\psi_C(x_C)$, C = cliques in a graph G.
- Find, $Z = \sum_{x_1,\ldots,x_n} \prod_C \psi_C(x_C)$

$x_1, \ldots x_n$ = good ordering of variables
$\mathcal{F} = \psi_C(x_C)$, C = cliques in a graph G.
for $i = 1 \ldots n$ **do**
  $\mathcal{F}_i$ = factors in $\mathcal{F}$ that contain x_i
  M_i = product of factors in $\mathcal{F}_i$
  $m_i = \sum_{x_i} M_i$
  $\mathcal{F} = \mathcal{F} - \mathcal{F}_i \cup \{m_i\}$
end for

Example: Variable elimination

- Given, $\psi_{12}(x_1, x_2)$, $\psi_{24}(x_2, x_4)$, $\psi_{23}(x_2, x_3)$, $\psi_{45}(x_4, x_5)$, , $\psi_{35}(x_3, x_5)$.
- Find, $Z = \sum_{x_1,\ldots,x_5} \psi_{12}(x_1, x_2)\psi_{24}(x_2, x_4)\psi_{23}(x_2, x_3)\psi_{45}(x_4, x_5)\psi_{35}(x_3, x_5)$.

1. x_1: $\prod\{\psi_{12}(x_1, x_2)\} \rightarrow M_1(x_1, x_2) \xrightarrow{\sum_{x_1}} m_1(x_2)$
2. x_2: $\prod\{\psi_{24}(x_2, x_4), \psi_{23}(x_2, x_3), m_1(x_2)\} \rightarrow M_2(x_2, x_3, x_4) \xrightarrow{\sum_{x_2}} m_2(x_3, x_4)$
3. x_3:$\prod\{\psi_{35}(x_3, x_5), m_2(x_3, x_4)\} \rightarrow M_3(x_3, x_4, x_5) \xrightarrow{\sum_{x_3}} m_3(x_4, x_5)$
4. x_4:$\prod\{\psi_{45}(x_4, x_5), m_3(x_4, x_5)\} \rightarrow M_4(x_4, x_5) \xrightarrow{\sum_{x_4}} m_4(x_5)$
5. x_5:$\prod\{m_5(x_5)\} \rightarrow M_5(x_5) \xrightarrow{\sum_{x_5}} Z$

Choosing a variable elimination order

- Complexity of VE $O(nm^w)$ where w is the maximum number of variables in any factor.
- Wrong elimination order can give rise to very large intermediate factors.
- Example: eliminating x_2 first will give a factor of size 4.
- Given an example where the penalty can be really severe (?)
- Choosing the optimal elimination order is NP hard for general graphs.
- Polynomial time algorithm exists for chordal graphs.
  - A graph is chordal or triangulated if all cycles of length greater than three have a shortcut.
- Optimal triangulation of graphs is NP hard. (Many heuristics)

Finding optimal order in a triangulated graph

Theorem

Every triangulated graph is either complete or has at least two simplicial vertices. A vertex is simplicial if its neighbors form a complete set.

Proof.

In supplementary. (not in syllabus) □

Goal: find optimal ordering for $P(x_1)$ inference. x_1 has to be last in the ordering.

Input: Graph G. $n =$ number of vertices of G
for $i = 2, \ldots, n$ **do**
  $\pi_i =$ pick any simplicial vertex in G other than 1.
  remove π_i from G
end for
Return ordering

Reusing computation across multiple inference queries

Given a chain graph with potentials $\psi_{i,i+1}(x_i, x_{i+1})$, suppose we need to compute all n marginals $P(x_1), \ldots, P(x_n)$.
Invoking variable elimination algorithm n times for each x_i will entail a cost of $n \times nm^2$. Can we go faster by reusing work across computations?

Junction tree algorithm

- 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

Junction tree

Junction tree JT of a triangulated graph G with nodes $x_1, \ldots, x_n$ is a tree where

- Nodes = maximal cliques of G
- Edges ensure that if any two nodes contain a variable x_i then x_i is present in every node in the unique path between them (Running intersection property).

Constructing a junction tree

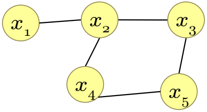
Efficient polynomial time algorithms exist for creating a JT from a triangulated graph.

1. Enumerate a covering set of cliques
2. Connect cliques to get a tree that satisfies the running intersection property.

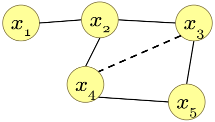
If graph is non-triangulated, triangulate first using heuristics, optimal triangulation is NP-hard.

Creating a junction tree from a graphical model

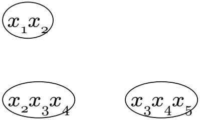
1. Starting graph



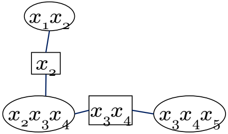
2. Triangulate graph



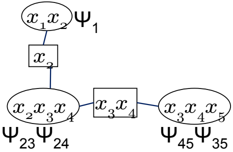
3. Create clique nodes



4. Create tree edges such that variables connected.



5) Assign potentials to exactly one subsumed clique node.



Finding cliques of a triangulated graph

Theorem

Every triangulated graph has a simplicial vertex, that is, a vertex whose neighbors form a complete set.

Input: Graph G. $n =$ number of vertices of G

for $i = 1, \ldots, n$ **do**

  $\pi_i =$ pick any simplicial vertex in G

  $C_i = \{\pi_i\} \cup Ne(\pi_i)$

  remove π_i from G

end for

Return maximal cliques from $C_1, \ldots C_n$

Connecting cliques to form junction tree

Separator variables = intersection of variables in the two cliques joined by an edge.

Theorem

A clique tree that satisfies the running intersection property maximizes the number of separator variables.

Proof: `https://people.eecs.berkeley.edu/~jordan/courses/281A-fall04/lectures/lec-11-16.pdf`

Input: Cliques: $C_1, \dots C_k$

Form a complete weighted graph H with cliques as nodes and edge weights = size of the intersection of the two cliques it connects.

T = maximum weight spanning tree of H

Return T as the junction tree.

Message passing on junction trees

- Each node c
  - sends *message* $m_{c \to c'}(.)$ to each of its neighbors c'
    - ★ once it has messages from every other neighbor $N(c) - \{c'\}$.
  - $m_{c \to c'}(.)$ = Message from c to c' is the result of sum-product elimination on side of the tree that contains clique c but not c' on the separator variables $s = c \cap c'$

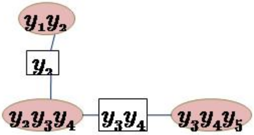
$$m_{c \to c'}(\mathbf{x}_s) = \sum_{\mathbf{x}_{c-s}} \psi_c(\mathbf{x}_c) \prod_{d \in N(c) - \{c'\}} m_{d \to c}(\mathbf{x}_{d \cap c})$$

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. $\Pr(x_i) \propto \sum_{\mathbf{x}_{c-x_i}} \psi_c(\mathbf{x}_c) \prod_{d \in N(c)} m_{d \to c}(\mathbf{x}_{d \cap c})$

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

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

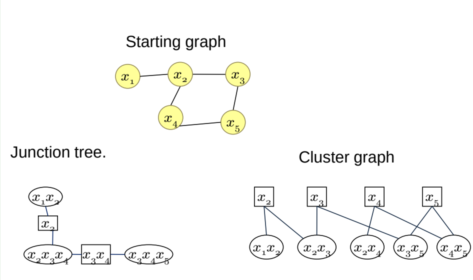
Why approximate inference

- Exact inference is NP hard. Complexity: $O(m^w)$
  - $w=$ 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 so as to satisfy running intersection property.
- Special case: Factor graphs.

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 [**?**]
  - Tree-reweighted message passing [**?**, **?**]
  - Residual belief probagation [**?**]
- Many have no guarantees of convergence. Specific tree-based orders do [**?**]
- Works well in practice, default method of choice.

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 [**?**].
- Greedy algorithms: relaxation labeling.
- Variational methods like mean-field and structured mean-field.
- LP and QP based approaches.

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. Learned: from examples
   - More popular since difficult for humans to assign numeric values
   - Many variants of parameterizing potentials.
     1. Table potentials: each entry a parameter, example, HMMs
     2. Potentials: combination of shared parameters and data attributes: example, CRFs.

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. Learned from examples
   - NP hard to find the optimal structure.
   - Widely researched, mostly posed as a branch and bound search problem.
   - Useful in dynamic situations

Learning potentials

Given sample $D = \{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$ of data generated from a distribution $P(\mathbf{x})$ represented by a graphical model with known structure G, learn potentials $\psi_C(\mathbf{x}_C)$.

Two settings:

1. All variables observed or not.
   1. Fully observed: each training sample $\mathbf{x}^i$ has all n variables observed.
   2. Partially observed: a subset of the variables are observed.
2. Potentials coupled with a log-partition function or not.
   1. No: Closed form solutions
   2. Yes: Potentials attached to arbitrary overlapping subset of variables in a UDGM. Example = edge potentials in a grid graph. iterative solution as in the case of learning with shared parameters Discussed later.

General framework for Parameter learning in graphical models

- Conditional distribution $\Pr(\mathbf{y}|\mathbf{x},\theta)$, potentials are function of $\mathbf{x}$ and parameters θ to be learned.
- $\mathbf{y} = y_1,\ldots,y_n$ forms a graphical model: directed or undirected.
- Undirected:

$$\Pr(y_1,\ldots,y_n|\mathbf{x},\theta) = \frac{\prod_C \psi_c(\mathbf{y}_c,\mathbf{x},\theta)}{Z_\theta(\mathbf{x})}$$
$$= \frac{1}{Z_\theta(\mathbf{x})}\exp(\sum_c F_\theta(\mathbf{y}_c,c,\mathbf{x}))$$

where $Z_\theta(\mathbf{x}) = \sum_{\mathbf{y}'}\exp(\sum_c F_\theta(\mathbf{y}'_c,c,\mathbf{x}))$
clique potential $\psi_c(\mathbf{y}_c,\mathbf{x}) = \exp(F_\theta(\mathbf{y}_c,c,\mathbf{x}))$

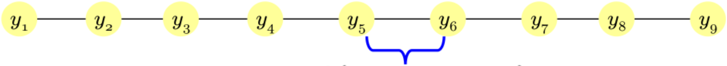
Forms of $F_\theta(\mathbf{y}_c, c, \mathbf{x})$

- Log-linear model over user-defined features. E.g. CRFs, Maxent models, etc.
  Let K be number of features. Denote a feature as $f_k(\mathbf{y}_c, c, \mathbf{x})$. Then,
  $$F_\theta(\mathbf{y}_c, c, \mathbf{x}) = \sum_{k=1}^{K} \theta_k f_k(\mathbf{y}_c, c, \mathbf{x})$$
- Arbitrary function, e.g. a neural network that takes as input $\mathbf{y}_c, c, \mathbf{x}$ and transforms them possibly non-linearly into a real value. θ are the parameters of the network.

Example: Named Entity Recognition

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 decompose over adjacent labels.

$$\mathbf{f}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{|\mathbf{x}|} \mathbf{f}(y_i, y_{i-1}, i, \mathbf{x})$$

Named Entity Recognition: 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

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

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

Training

Given

- N input output pairs $D = \{(\mathbf{x}^1, \mathbf{y}^1), (\mathbf{x}^2, \mathbf{y}^2), \ldots, (\mathbf{x}^N, \mathbf{y}^N)\}$
- Form of F_θ
- Learn parameters θ by maximum likelihood.

$$\max_\theta LL(\theta, D) = \max_\theta \sum_{i=1}^{N} \log \Pr(\mathbf{y}^i | \mathbf{x}^i, \theta)$$

Training undirected graphical model

$$
\begin{aligned}
LL(\theta, D) &= \sum_{i=1}^{N} \log \Pr(\mathbf{y}^i | \mathbf{x}^i, \theta) \\
&= \sum_{i=1}^{N} \log \frac{1}{Z_\theta(\mathbf{x}^i)} \exp(\sum_c F_\theta(\mathbf{y}^i_c, c, \mathbf{x}^i)) \\
&= \sum_i [\sum_c F_\theta(\mathbf{y}^i_c, c, \mathbf{x}^i) - \log Z_\theta(\mathbf{x}^i)
\end{aligned}
$$

The first part is easy to compute but the second term requires to invoke an inference algorithm to compute $Z_\theta(\mathbf{x}^i)$ for each i. Computing the gradient of the above objective with respect to θ also requires inference.

Training via gradient descent

Assume log-linear models like in CRFs where $F_\theta(\mathbf{y}^i_c, c, \mathbf{x}^i) = \theta \cdot \mathbf{f}(\mathbf{x}^i, \mathbf{y}^i_c, c)$ Also, for brevity write $\mathbf{f}(\mathbf{x}^i, \mathbf{y}^i) = \sum_c \mathbf{f}(\mathbf{x}^i, \mathbf{y}^i_c, c)$

$$LL(\theta) = \sum_i \log \Pr(\mathbf{y}^i | \mathbf{x}^i, \theta) = \sum_i (\theta \cdot \mathbf{f}(\mathbf{x}^i, \mathbf{y}^i) - \log Z_\theta(\mathbf{x}^i))$$

Add a regularizer to prevent over-fitting.

$$\max_\theta \sum_i (\theta \cdot \mathbf{f}(\mathbf{x}^i, \mathbf{y}^i) - \log Z_\theta(\mathbf{x}^i)) - \|\theta\|^2 / C$$

Concave in $\theta \implies$ gradient descent methods will work.

Gradient of the training objective

$$
\begin{aligned}
\nabla L(\theta) &= \sum_i \mathbf{f}(\mathbf{x}^i, \mathbf{y}^i) - \frac{\sum_{\mathbf{y}'} \mathbf{f}(\mathbf{y}', \mathbf{x}^i) \exp \theta \cdot \mathbf{f}(\mathbf{x}^i, \mathbf{y}')}{Z_\theta(\mathbf{x}^i)} - 2\theta/C \\
&= \sum_i \mathbf{f}(\mathbf{x}^i, \mathbf{y}^i) - \sum_{\mathbf{y}'} \mathbf{f}(\mathbf{x}^i, \mathbf{y}') \Pr(\mathbf{y}'|\theta, \mathbf{x}^i) - 2\theta/C \\
&= \sum_i \mathbf{f}(\mathbf{x}^i, \mathbf{y}^i) - E_{\Pr(\mathbf{y}'|\theta,\mathbf{x}^i)} \mathbf{f}(\mathbf{x}^i, \mathbf{y}') - 2\theta/C
\end{aligned}
$$

$$
\begin{aligned}
E_{\Pr(\mathbf{y}'|\theta,\mathbf{x}^i)} f_k(\mathbf{x}^i, \mathbf{y}') &= \textstyle\sum_{\mathbf{y}'} f_k(\mathbf{x}^i, \mathbf{y}') \Pr(\mathbf{y}'|\theta, \mathbf{x}^i) \\
&= \textstyle\sum_{\mathbf{y}'} \sum_c f_k(\mathbf{x}^i, \mathbf{y}'_c, c) \Pr(\mathbf{y}'|\theta, \mathbf{x}^i) \\
&= \textstyle\sum_c \sum_{\mathbf{y}'_c} f_k(\mathbf{x}^i, \mathbf{y}'_c, c) \Pr(\mathbf{y}'_c|\theta, \mathbf{x}^i)
\end{aligned}
$$

Computing $E_{\Pr(\mathbf{y}|\theta^t,\mathbf{x}^i)} f_k(\mathbf{x}^i, \mathbf{y})$

Three steps:

1. $\Pr(\mathbf{y}|\theta^t, \mathbf{x}^i)$ is represented as an undirected model where nodes are the different components of $\mathbf{y}$, that is $y_1, \ldots, y_n$.
   The potential $\psi_c(\mathbf{y}_c, \mathbf{x}, \theta)$ on clique c is $\exp(\theta^t \cdot \mathbf{f}(\mathbf{x}^i, \mathbf{y}_c^i, c))$
2. Run a sum-product inference algorithm on above UGM and compute for each $c, \mathbf{y}_c$ marginal probability $\mu(\mathbf{y}_c, c, \mathbf{x}^i)$.
3. Using these μs we compute
   $E_{\Pr(\mathbf{y}|\theta^t,\mathbf{x}^i)} f_k(\mathbf{x}^i, \mathbf{y}) = \sum_c \sum_{\mathbf{y}_c} \mu(\mathbf{y}_c, c, \mathbf{x}^i) f_k(\mathbf{x}^i, c, \mathbf{y}_c)$

Example

Consider a parameter learning task for an undirected graphical model on 3 variables $\mathbf{y} = [y_1 \ y_2 \ y_3]$ where each $y_i = +1$ or 0 and they form a chain. Let the following two features be defined for it.

$f_1(i, \mathbf{x}, y_i) = x_i y_i$ (where x_i=intensity of pixel i)

$f_2((i,j), \mathbf{x}, (y_i, y_j)) = [\![y_i \neq y_j]\!]$

where $[\![z]\!] = 1$ if $z =$ true and 0 otherwise.

Initial parameters $\theta = [\theta_1, \theta_2] = [3, -2]$

Examples: $\mathbf{x}^1 = [0.1, 0.7, 0.3], \mathbf{y}^1 = [1, 1, 0]$

Using these we can calculate:

1. Node potentials for y_i as $\exp(\theta_1 x_i y_i)$. For e.g. for y_1 it is $[\psi_1(0), \psi_1(1)] = [1, e^{3\times0.1}]$
2. Edge potentials $\psi_{12}(y_1, y_2) = \psi_{23}(y_2, y_3) = 1$ if $y_1 = y_2$ and e^{-2} if $y_1 \neq y_2$

Example (continued)

1. Use above potentials to run sum-product inference on a junction tree to calculate marginals $\mu(y_i, i)$ and $\mu(y_i, y_j, (i, j))$
2. Using these we calculate expected value of features as:

$$E[f_1(\mathbf{x}^1, \mathbf{y})] = \sum_i x_i \mu_i(1, i) = 0.1\mu(1,1) + 0.7\mu(1,2) + 0.3\mu(1,3)$$

$$E[f_2(\mathbf{x}^1, \mathbf{y})] = \mu(1,0,(1,2)) + \mu(0,1,(1,2)) + \mu(1,0,(2,3)) + \mu(0,1,(2,3))$$

3. The value of $\mathbf{f}(\mathbf{x}^1, \mathbf{y}^1)$ for each feature is (Note value of $\mathbf{y}^1 = [1, 1, 0]$):

$$f_1(\mathbf{x}^1, \mathbf{y}^1) = 0.1 * 1 + 0.7 * 1 + 0.3 * 0 = 0.8$$

$$f_2(\mathbf{x}^1, \mathbf{y}^1) = [\![y_1^1 \neq y_2^1]\!] + [\![y_2^1 \neq y_3^1]\!] = 1$$

4. The gradient of each parameter is then.

$$\nabla L(\theta_1) = 0.8 - E[f_1(\mathbf{x}^1, \mathbf{y})] - 2 * 3/C$$

$$\nabla L(\theta_2) = 1 - E[f_2(\mathbf{x}^1, \mathbf{y})] + 2 * 2/C$$

Another Example

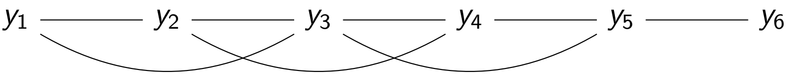
Consider a parameter learning task for an undirected graphical model on six variables $\mathbf{y} = [y_1 \;\; y_2 \;\; y_3 \;\; y_4 \;\; y_5 \;\; y_6]$ where each $y_i = +1$ or -1. Let the following eight features be defined for it.

$f_1(y_i, y_{i+1}) = [\![y_i + y_{i+1} > 1]\!], 1 \leq i < 5$ $\quad$ $\mathrm{f}_2(y_1, y_3) = -2y_1y_3$
$\mathrm{f}_3(y_2, y_3) = y_2y_3$ $\quad$ $\mathrm{f}_4(y_3, y_4) = y_3y_4$
$\mathrm{f}_5(y_2, y_4) = [\![y_2y_4 < 0]\!]$ $\quad$ $\mathrm{f}_6(y_4, y_5) = 2y_4y_5$
$\mathrm{f}_7(y_3, y_5) = -y_3y_5$ $\quad$ $\mathrm{f}_8(y_5, y_6) = [\![y_5 + y_6 > 0]\!]$.

where $[\![z]\!] = 1$ if $z =$ true and 0 otherwise. That is, $\mathbf{f}(\mathbf{y}) = [f_1 \;\; f_2 \;\; f_3 \;\; f_4 \;\; f_5 \;\; f_6 \;\; f_7 \;\; f_8]^T$. Assume the corresponding weight vector to be $\theta = [1 \; 1 \; 1 \; 2 \; 2 \; 1 \;\; -1 \; 1]^T$

Example

Draw the underlying graphical model corresponding to the 6 variables.



Draw an arc between any two y which appear together in any of the 8 features.

Example

Draw the junction tree corresponding to the graph above and assign potentials to each node of your junction tree so that you can run message passing on it to find $Z = \sum_{\mathbf{y}} \theta^T \mathbf{f}(\mathbf{x}, \mathbf{y})$, that is, define $\psi_c(\mathbf{y}_c)$ in terms of the above quantities for each clique node c in the JT.
For clique c, $\psi_c(\mathbf{y}_c) = \exp(\theta_c \cdot \mathbf{f}_c(\mathbf{x}, \mathbf{y}_c))$. *log* of the potentials are shown below

y_1, y_2, y_3	y_2, y_3, y_4	y_3, y_4, y_5	y_5, y_6
$1.f_1(y_1, y_2) +$ $1.f_1(y_2, y_3) +$ $1.f_2(y_1, y_3) +$ $1.f_3(y_2, y_3)$	$2.f_5(y_2, y_4) +$ $1.f_1(y_3, y_4) +$ $2.f_4(y_3, y_4)$	$-1.f_7(y_3, y_5) +$ $1.f_1(y_4, y_5) +$ $1.f_6(y_4, y_5)$	$1.f_1(y_5, y_6) +$ $1.f_8(y_5, y_6)$

Example

Suppose you use the junction tree above to compute the marginal probability for each pair of adjacent variables in the graph of part (a). Let $\mu_{ij}(-1,1), \mu_{ij}(1,1), \mu_{ij}(-1,-1), \mu_{ij}(1,-1)$ denote the marginal probability of variable pairs y_i, y_j taking values (-1,1), (1,1), (-1,-1) and (1,-1) respectively. Express the expected value of the following features in terms of the μ values.

1. $$f_1 = \sum_i \big(f_1(-1,-1)\mu_{i,i+1}(-1,-1) + f_1(-1,1)\mu_{i,i+1}(-1,1) + f_1(1,-1)\mu_{i,i+1}(1,-1) + f_1(1,1)\mu_{i,i+1}(1,1)\big)$$
2. $f_2 = 2\big(-\mu_{1,3}(-1,-1) + \mu_{1,3}(-1,1) + \mu_{1,3}(1,-1) - \mu_{1,3}(1,1)\big)$
3. $f_8 = \mu_{56}(1,1)$

Training algorithm

Initialize $\theta^0 = \mathbf{0}$
for $t = 1 \dots \mathrm{T}$ **do**
  for $i = 1 \dots N$ **do**
    $g_{k,i} = f_k(\mathbf{x}^i, \mathbf{y}^i) - E_{\Pr(\mathbf{y}'|\theta^t, \mathbf{x}^i)} f_k(\mathbf{x}^i, \mathbf{y}') \quad k = 1 \dots K$
  end for
  $g_k = \sum_i g_{k,i} \quad k = 1 \dots K$
  $\theta_k^t = \theta_k^{t-1} + \gamma_t(g_k - 2\theta_k^{t-1}/C)$
  Exit if $\|\mathbf{g}\| \approx zero$
end for

Running time of the algorithm is $O(INn(m^2 + K))$ where I is the total number of iterations.

Local conditional probability for BN

$$\Pr(y_1,\ldots,y_n|\mathbf{x},\theta) = \prod_j \Pr(y_j|\mathbf{y}_{\mathrm{Pa}(j)},\mathbf{x},\theta)$$

$$= \prod_j \frac{\exp(F_\theta(\mathbf{y}_{\mathrm{Pa}(j)},y,j,\mathbf{x}))}{\sum_{y'=1}^{m}\exp(F_\theta(\mathbf{y}_{\mathrm{Pa}(j)},y',j,\mathbf{x}))}$$

Training for BN

$$
\begin{aligned}
LL(\theta, D) &= \sum_{i=1}^{N} \log \Pr(\mathbf{y}^i | \mathbf{x}^i, \theta) \\
&= \sum_{i=1}^{N} \log \prod_{j} \Pr(y_j^i | \mathbf{y}_{\mathrm{Pa}}^i(j), \mathbf{x}^i, \theta) \\
&= \sum_{i} \sum_{j} \log \Pr(y_j^i | \mathbf{y}_{\mathrm{Pa}}^i(j), \mathbf{x}^i, \theta) \\
&= \sum_{i} \sum_{j} F_\theta(\mathbf{y}_{\mathrm{Pa}(j)}^i, y_j^i, j, \mathbf{x}^i)) - \log \sum_{y'=1}^{m} \exp(F_\theta(\mathbf{y}_{\mathrm{Pa}(j)}^i, y', j, \mathbf{x}^i))
\end{aligned}
$$

Like normal classification task. No challenge arising during training because of graphical model. Normalizer is easy to compute.

Table Potentials in the feature framework.

Assume $\mathbf{x}^i$ does not exist..(As in HMMs)

- $F_\theta(\mathbf{y}^i_{\text{Pa}(j)}, y^i_j, j)) = \log P(y^i_j | \mathbf{y}^i_{\text{Pa}(j)})$, normalizer vanishes.
- $\Pr(y_j | \mathbf{y}_{\text{Pa}(j)}) =$ Table of real values denoting the probability of each value of x_j corresponding to each combination of values of the parents (θ^j).
- If each variables takes m possible values, and has k parents, then each $\Pr(y_j | \mathbf{y}_{\text{Pa}(j)})$ will require $m^k(m)$ parameters in θ^j.

$$\theta^j_{vu_1,\ldots,u_k} = \Pr(y_j = v | \mathbf{y}_{pa(j)} = u_1, \ldots, u_k)$$

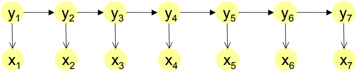
Maximum Likelihood estimation of parameters

$$
\begin{aligned}
& \max_\theta \sum_i \sum_j \log P(y_j^i | \mathbf{y}^i_{\mathrm{Pa}(j)}) \\
= \ & \max_\theta \sum_i \sum_j \log \theta^j_{y_j^i \mathbf{y}^i_{(j)}} \quad s.t. \sum_v \theta^j_{vu_1,\ldots,u_k} = 1 \ \forall j, u_1, \ldots, u_k \\
= \ & \max_\theta \sum_i \sum_j \log \theta^j_{y_j^i \mathbf{y}^i_{(j)}} - \sum_j \sum_{u_1,\ldots,u_k} \lambda^j_{u_1,\ldots,u_k} (\sum_v \theta^j_{vu_1,\ldots,u_k} - 1)
\end{aligned}
$$

Solve above using gradient descent to get

$$
\theta^j_{vu_1,\ldots,u_k} = \frac{\sum_{i=1}^N [[y_j^i == v, \mathbf{y}^i_{Pa(j)} = u_1, \ldots, u_k]]}{\sum_{i=1}^N [[\mathbf{y}^i_{Pa(j)} = u_1, \ldots, u_k]]} \tag{1}
$$

Partially observed, decoupled potentials



EM Algorithm

Input: Graph G, Data D with observed subset of variables $\mathbf{x}$ and hidden variables $\mathbf{z}$.

Initially ($t = 0$): Assign random variables of parameters $\Pr(x_j|pa(x_j))^t$

for $= 1, \ldots, T$ **do**

  E-step

  for $i = 1, \ldots, N$ **do**

    Use inference in G to estimate conditionals $\Pr_i(\mathbf{z}_c|\mathbf{x}^i)^t$ for all variable subsets $(i, pa(i))$ involving any hidden variable.

  end for

  M-step

$$\Pr(x_j|pa(x_j) = \mathbf{z}_c)^t = \frac{\sum_{i=1}^{N} \Pr_i(\mathbf{z}_c|\mathbf{x}^i)[[x_j^i == x_j]]}{\sum_{i=1}^{N} \Pr_i(\mathbf{z}_c|\mathbf{x}^i)^t}$$

end for

More on graphical models

- Koller and Friedman, Probabilistic Graphical Models: Principles and Techniques. MIT Press, 2009.
- Wainwright's article in FnT for Machine Learning. 2009.
- 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.