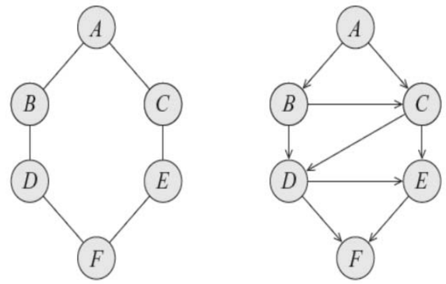
Lecture 7: February 4, 2015
cs 573: Probabilistic Reasoning
Professor Nevatia
Spring 2015

Review

- Assignment #2 due Feb 9
- Update on TA office hours: is one day per week enough?
- Last lecture:
  - Intro to Markov Networks (undirected graphs)
    - Factors and Factor product
    - Reduced Factors
    - Gibbs distribution
    - Local and Global Independences
    - BNs to MNs
- Today's objective
  - Chordal graphs
  - Log Linear Models
  - CRFs

MN => BN

- Much harder in this direction



(a) (b)

- Start with nodes in MN (in some order). Construct minimal I-map BN; example from one order is shown above
  - Consider A,B,C,D,E,F order
  - A,B and A,C obviously connected but we also need edge from B to C as in MN, (B not ⊥ C|A)
  - Consider D: in MN (D not ⊥ C|B) so edge from C to D in BN..
  - from C to D in BN
- Note that directed graph is chordal (triangulated)
- In general, minimal I-map, G (BN), for any MN, H, is necessarily chordal

Chordal Graph Properties

- If H is a non-chordal MN, there is no BN, G, that is a perfect map of H.
- If H is a chordal MN, there exists a BN, G, such that I(H) = I(G)
- Chordal graphs also map to *clique trees* that permit efficient inference algorithms (ch. 10); we will postpone definition of clique-trees until we make use of them in the inference algorithms
- Venn diagram
  – BN and MN subsets of P
  – Intersection of BN and MN is set of chordal graphs

Log Linear Model

- Rewrite Φ (D) = $e^{-\varepsilon(D)}$
- ε (D) = -ln Φ (D) , ε is called an *energy* function (Φ is called clique *potential*)

$$P(X_1, \ldots, X_n) \propto \exp\left[-\sum_{i=1}^{m} \epsilon_i(\boldsymbol{D}_i)\right]$$

(a) $\epsilon_1(A, B)$

a^0	b^0	−3.4
a^0	b^1	−1.61
a^1	b^0	0
a^1	b^1	−2.3

(b) $\epsilon_2(B, C)$

b^0	c^0	−4.61
b^0	c^1	0
b^1	c^0	0
b^1	c^1	−4.61

(c) $\epsilon_3(C, D)$

c^0	d^0	0
c^0	d^1	−4.61
c^1	d^0	−4.61
c^1	d^1	0

(d) $\epsilon_4(D, A)$

d^0	a^0	−4.61
d^0	a^1	0
d^1	a^0	0
d^1	a^1	−4.61

(a) $\phi_1(A, B)$

a^0	b^0	30
a^0	b^1	5
a^1	b^0	1
a^1	b^1	10

(b) $\phi_2(B, C)$

b^0	c^0	100
b^0	c^1	1
b^1	c^0	1
b^1	c^1	100

(c) $\phi_3(C, D)$

c^0	d^0	1
c^0	d^1	100
c^1	d^0	100
c^1	d^1	1

(d) $\phi_4(D, A)$

d^0	a^0	100
d^0	a^1	1
d^1	a^0	1
d^1	a^1	100

Log Linear Model

- Often, we will find that the logarithmic representation is easier to manipulate and numerically more stable
  - Adding rather than multiplying many numbers.
  - Minimizing energy is equivalent to maximizing probability.
- Feature Functions
  - Feature f (**D**) is a function from val (**D**) to R.
  - *Indicator* feature: has value 1 for some value, y in val (D) (*i.e.* for some assignment of values to variables in D)
  - Another feature: $\varepsilon(A_1, A_2) = -3$ if $A_1 = A_2$; 0 otherwise.
- $\varepsilon = \sum_k w_k f_k (D_k)$; w_k weight parameters; features typically but not necessarily binary, multiple features over same clique acceptable
- $P = (1/Z)\, e^{-\sum_k w_k f_k (D_k)}$

Feature Representation Example

- We can represent arbitrary factors using feature functions
- Consider ε_1 (a,b) from misconception example

$\epsilon_1(A, B)$		
a^0	b^0	-3.4
a^0	b^1	-1.61
a^1	b^0	0
a^1	b^1	-2.3

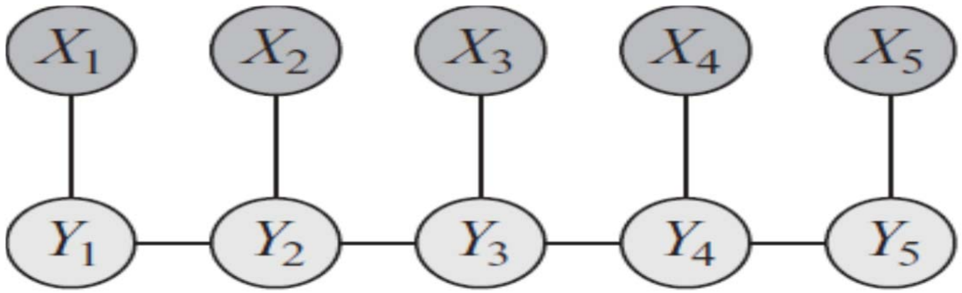
- Define f_1(A, B) = 1 when A=0 and B=0; zero otherwise
    - This provides first entry in the factor, set weight = -3.4
  - f_2(A, B) = 1 when A =0 and B=1, zero otherwise
    - Provides second entry into the table, set weight = -1.61
  - …
- Flexible methodology: can assign weights to arbitrary functions of variables in a clique
  - Setting of these weights may not always be intuitive and learning them may be more complex than learning conditional probabilities

Overparameterization

- It is easy to overparameterize a Markov Network
  - Nodes can be in multiple cliques, evidence for them can be distributed differently
- Canonical parameterization
  - Factors for each clique (not just the maximal cliques)
  - We skip details (see section 4.4.2.1)
- If we use feature functions, some linear relations between them apply (i.e. they are not linearly independent)
  - Can solve for a minimal set of features
  - Again, we skip details, see section 4.4.2.2

Conditional Random Fields (CRFs)

- Encodes conditional distribution P(**Y**|**X**); *Y* is a set of target variables (unknowns), *X* is a set of observed variables. Note that MRF encodes P(**Y**,**X**).
- First an example:



A conditional random field *is an undirected graph $\mathcal{H}$ whose nodes correspond to $\boldsymbol{X} \cup \boldsymbol{Y}$; the network is annotated with a set of factors $\phi_1(\boldsymbol{D}_1), \ldots, \phi_m(\boldsymbol{D}_m)$ such that each $\boldsymbol{D}_i \not\subseteq \boldsymbol{X}$. The network encodes a conditional distribution as follows:*

$$\begin{aligned} P(\boldsymbol{Y} \mid \boldsymbol{X}) &= \frac{1}{Z(\boldsymbol{X})}\tilde{P}(\boldsymbol{Y}, \boldsymbol{X}) \\ \tilde{P}(\boldsymbol{Y}, \boldsymbol{X}) &= \prod_{i=1}^{m} \phi_i(\boldsymbol{D}_i) \\ Z(\boldsymbol{X}) &= \sum_{\boldsymbol{Y}} \tilde{P}(\boldsymbol{Y}, \boldsymbol{X}). \end{aligned} \tag{4.11}$$

Two variables in $\mathcal{H}$ are connected by an (undirected) edge whenever they appear together in the scope of some factor.

CRFs: General Definition

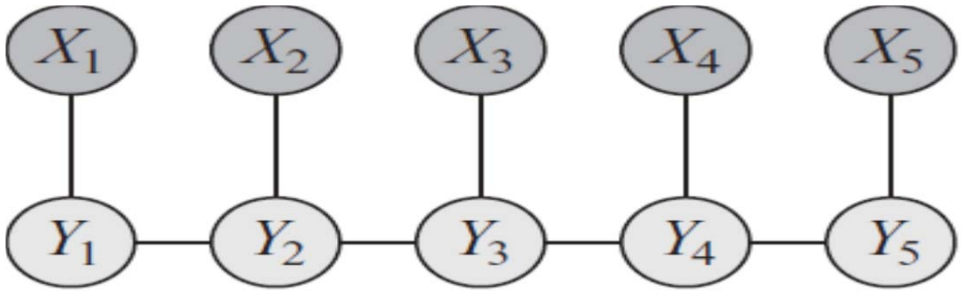
A conditional random field *is an undirected graph $\mathcal{H}$ whose nodes correspond to $\boldsymbol{X} \cup \boldsymbol{Y}$; the network is annotated with a set of factors $\phi_1(\boldsymbol{D}_1), \ldots, \phi_m(\boldsymbol{D}_m)$ such that each $\boldsymbol{D}_i \not\subseteq \boldsymbol{X}$. The network encodes a conditional distribution as follows:*

$$
\begin{aligned}
P(\boldsymbol{Y} \mid \boldsymbol{X}) &= \frac{1}{Z(\boldsymbol{X})} \tilde{P}(\boldsymbol{Y}, \boldsymbol{X}) \\
\tilde{P}(\boldsymbol{Y}, \boldsymbol{X}) &= \prod_{i=1}^{m} \phi_i(\boldsymbol{D}_i) \qquad (4.11) \\
Z(\boldsymbol{X}) &= \sum_{\boldsymbol{Y}} \tilde{P}(\boldsymbol{Y}, \boldsymbol{X}).
\end{aligned}
$$

Two variables in $\mathcal{H}$ are connected by an (undirected) edge whenever they appear together in the scope of some factor. ■

Note: connections do not have to be between adjacent nodes in a chain only, one target variable node may be connected to multiple evidence nodes. Flexible due to easy use of feature functions and log-linear models. Linear chain graphs are natural for many problems, e.g. text, activity analysis, robot navigation….

Conditional Random Fields (CRFs)



$$P(\boldsymbol{Y} \mid \boldsymbol{X}) = \frac{1}{Z(\boldsymbol{X})} \tilde{P}(\boldsymbol{Y}, \boldsymbol{X})$$

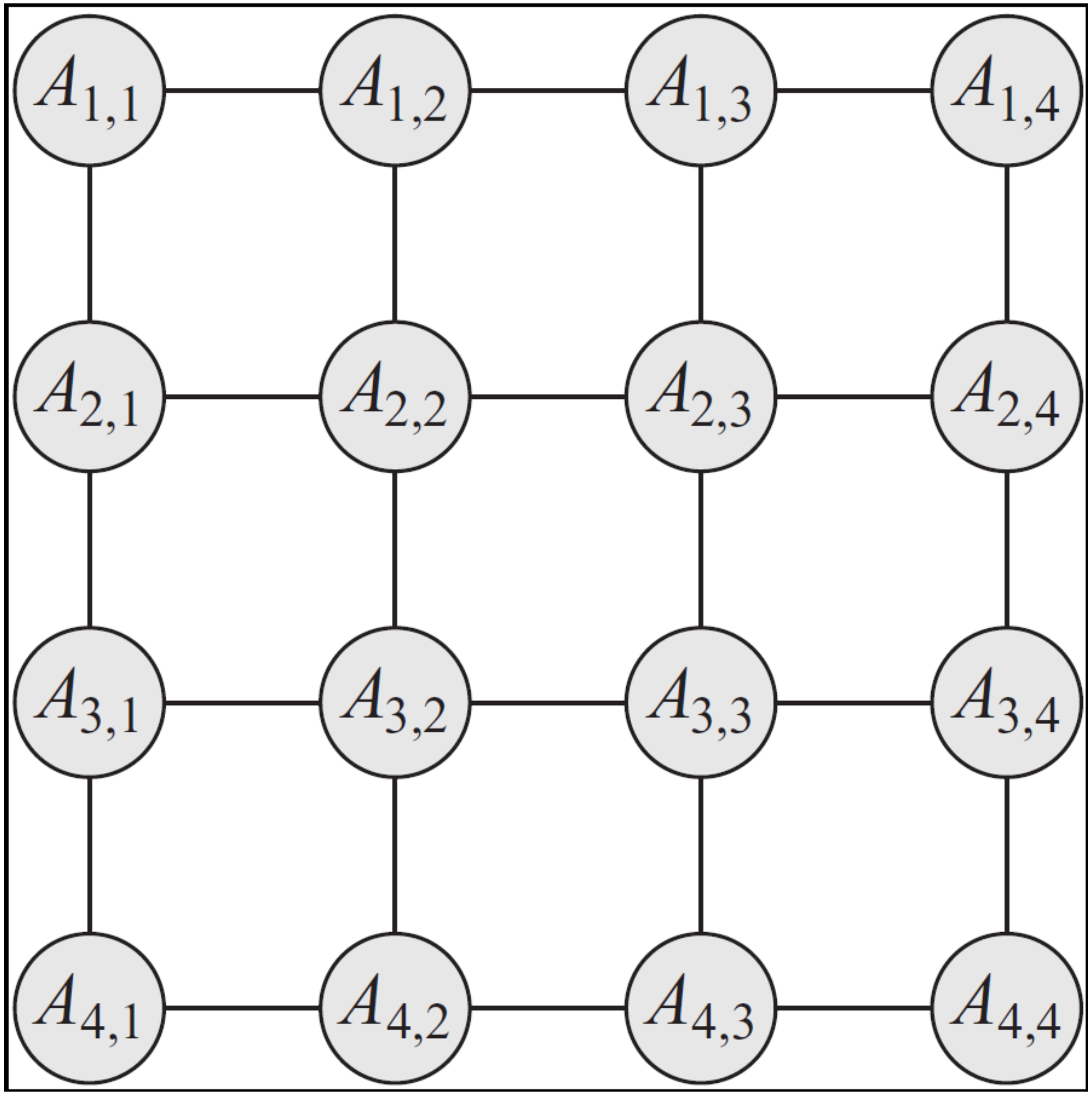
$$\tilde{P}(\boldsymbol{Y}, \boldsymbol{X}) = \prod_{i=1}^{k-1} \phi(Y_i, Y_{i+1}) \prod_{i=1}^{k} \phi(Y_i, X_i)$$

$$Z(\boldsymbol{X}) = \sum_{\boldsymbol{Y}} \tilde{P}(\boldsymbol{Y}, \boldsymbol{X}).$$

Note: no factors over just variables in X; Z is now a function of X, the values of the observed variables. Connections do not have to be between adjacent nodes in a chain only, one target variable node may be connected to multiple evidence nodes.

This model can not predict P(**X**) or P(**Y,X**). In this example, functions are same across the network.

Pairwise Markov Random Field (MRF)



Notes:

a) could represent as a single clique, but would be highly complex

b) Pairwise MRFs can not represent all distributions

c) Commonly used in vision and other applications for simplicity

d) CRF if each node is also connected to an observable

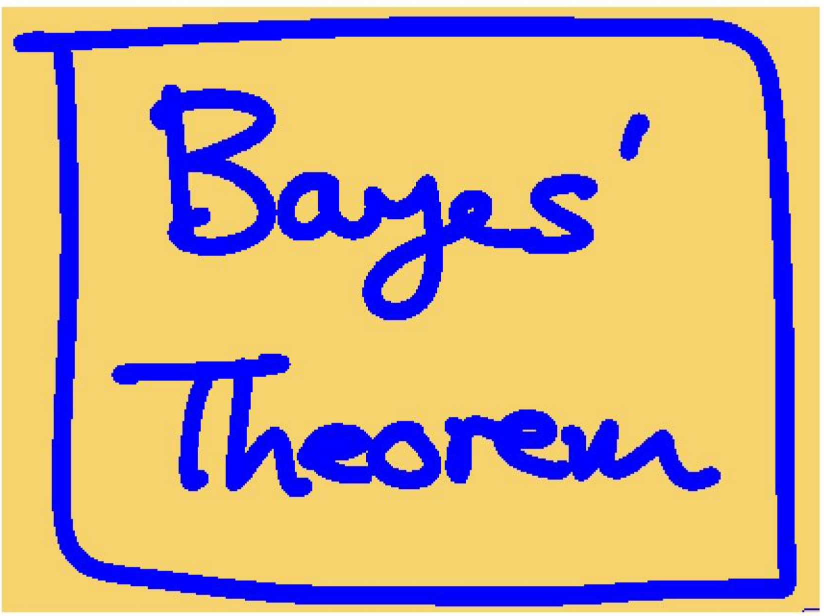
MRFs/CRFs in Computer Vision (Illustrations only)

- Consider each node also connected to another “observation” node (node whose value is given); this makes MRF into a CRF (Conditional Random Field) but term MRF is still commonly used.
- *Denoising*: Observed data is corrupted by noise. Goal is to label nodes with the *correct* numbers. Unary function has high value if label is close to observed value. Binary factor prefers similar labels for neighbors. MAP solution gives the denoised estimate.
- *Segmentation*: Label each pixel as “foreground” or “background”. Unary function provides a label according to some visual features at the point (color, texture, brightness…). Binary function prefers continuity (*i.e.* neighbors with different labels have low values). MAP solution gives best segmentation, given this model. Results are considerably better by considering neighbors.
- Labeling: assign class labels to pixels/regions

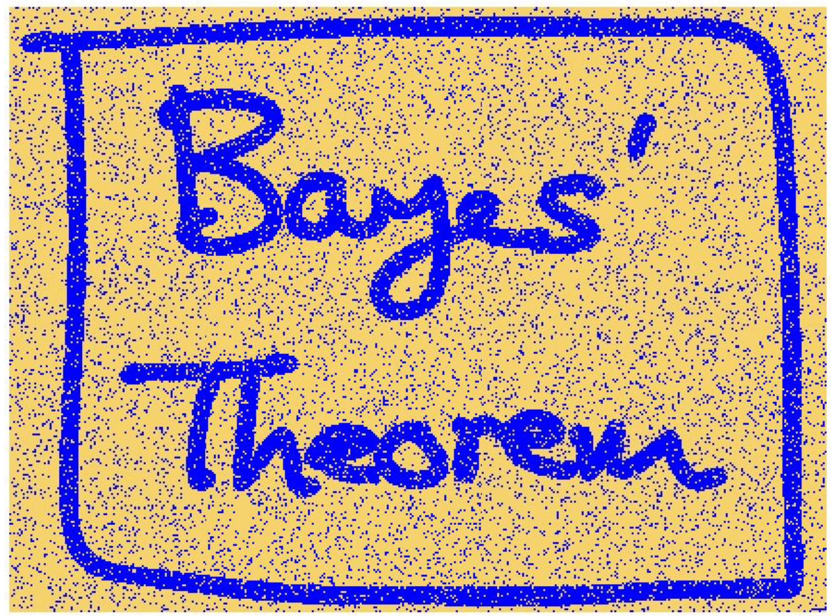
Ising Model

- Originally developed for statistical physics (modeling influence of neighboring atoms)
- Let random variables be binary (have +1 or -1 values)
- $\varepsilon_{i,j}(x_i, x_j) = w_{i,j}\, x_i\, x_j$
- Positive contribution of $w_{i,j}$ when $x_i = x_j$; negative $w_{i,j}$ otherwise.
- Example: same spin for two atoms in ferromagnetic analysis; same intensity value or label in image analysis …
- Combine with unary energy term (based on probability of each random variable value by itself).
- Efficient algorithms exist for computing MAP solution for this model
- Boltzmann machine: variables take values {0,1}
  - Used to model a simple neuron behavior
- *Potts* model is a generalization for multiple valued variables

Illustration: Image De-Noising



Original Image

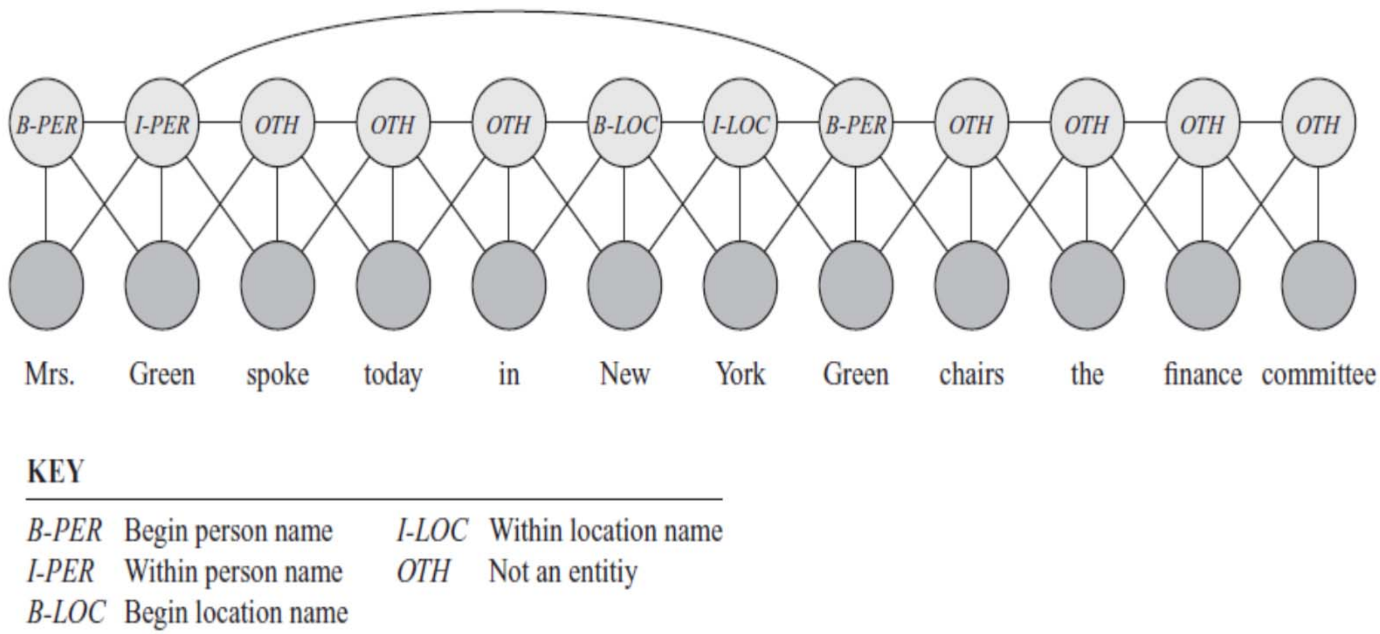


Noisy Image

From: Bishop Book

CRF: Text Analysis Example

- Network for text analysis. CRFs were originally introduced for text analysis (~2001) though now they find common in use in many applications. Note multiple connections to observation nodes. Also, model is not quite a linear chain (contrasts with HMMs which we haven't studied yet).



(a)

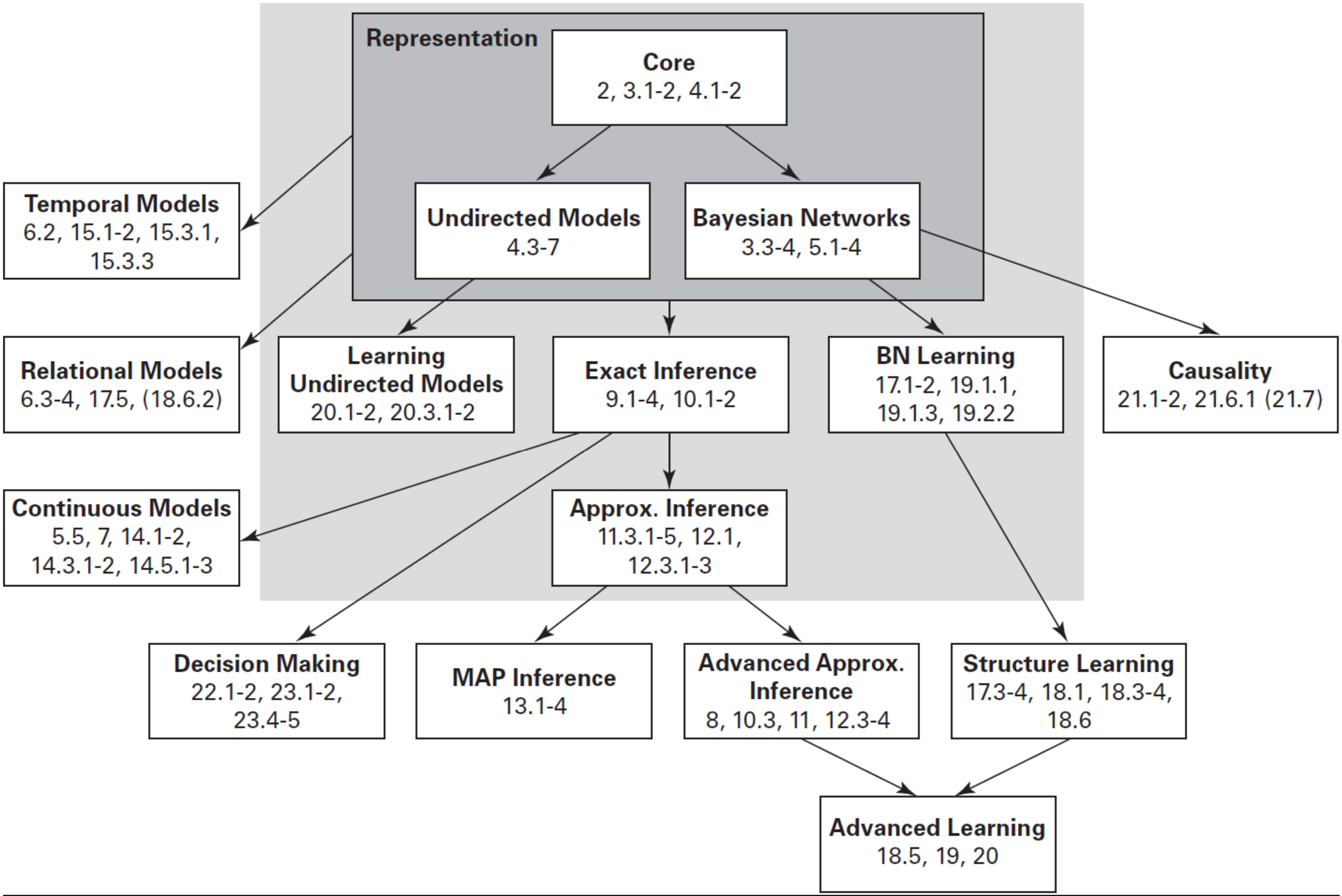
Summary of Topics Covered so Far

- Foundations (chapter 2) basic concepts of probability theory
- Bayesian networks and Markov networks
  - Basic representation
  - Factorization, conditional independences
  - I-maps, P-maps
  - Graphs $\Leftrightarrow$ distributions,
  - BN $\Leftrightarrow$ MN
- Local models (just summarized)
- We have covered chapters 2-5 except for
  - Sections 3.4.3, 4.4.2, 4.6.2, 5.3.2, 5.4.3, 5.4.4, 5.5.1
  - We have (mostly) skipped proofs of theorems

What More is There (in Representation)?

- Temporal models (ch. 6)
  - Future state depends on current state, dependency is not time dependent
  - Can describe relations by a "template" rather than a long sequence of variables
  - We will cover representation together with temporal inference later in the course
- "Plate" Models (ch. 6)
  - Describe multiple variables with same parameters, *i.e.* tossing hundred fair coins; will study along with parameter learning (time permitting)
- Gaussian Networks (ch. 7): Useful in modeling continuous variables: will study a bit later
- Exponential Family: General framework that includes many important distributions classes (Binomial, multinomial, Gaussian…)
  - Will likely not be covered in the class

Book Plan



Representation
Core
2, 3.1-2, 4.1-2
Temporal Models
6.2, 15.1-2, 15.3.1, 15.3.3
Undirected Models
4.3-7
Bayesian Networks
3.3-4, 5.1-4
Relational Models
6.3-4, 17.5, (18.6.2)
Learning Undirected Models
20.1-2, 20.3.1-2
Exact Inference
9.1-4, 10.1-2
BN Learning
17.1-2, 19.1.1, 19.1.3, 19.2.2
Causality
21.1-2, 21.6.1 (21.7)
Continuous Models
5.5, 7, 14.1-2, 14.3.1-2, 14.5.1-3
Approx. Inference
11.3.1-5, 12.1, 12.3.1-3
Decision Making
22.1-2, 23.1-2, 23.4-5
MAP Inference
13.1-4
Advanced Approx. Inference
8, 10.3, 11, 12.3-4
Structure Learning
17.3-4, 18.1, 18.3-4, 18.6
Advanced Learning
18.5, 19, 20

Next Class

- Read sections 9.1 to 9.3 of the KF book