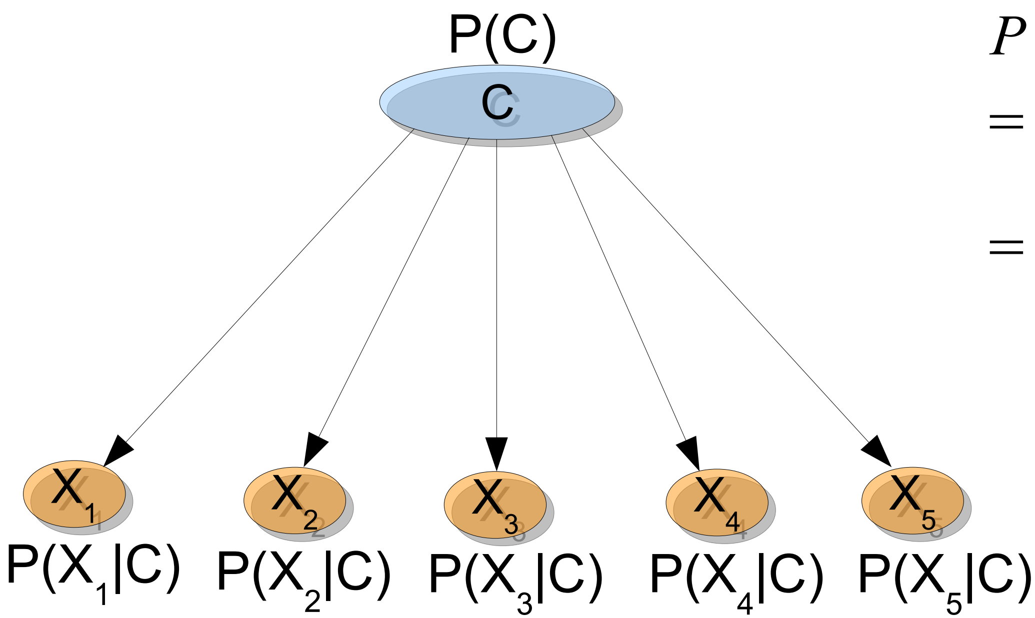
Learning with Missing Data

Handling Missing Data

- Different types of missing data: missing completely a random, missing at random, not missing at random
- Latent (hidden) variable models, like the finite mixture model, always have to deal with hidden data
- We either are interested in the missing data (e.g., we could be interested in the values of the a hidden variable if it corresponds to a clustering of data), or it is treated as ”nuicance” (e.g., if the hidden ”class” variable is only used as a modeling tool to produce a joint probability distribution on the observed variables)
- In the latter case, a Bayesian attempts to marginalize over the hidden data

The Finite Mixture Model



$$P(D) = P(X_1^n, \ldots, X_5^n)$$
$$= \sum_{C^n} P(C^n) P(X_1^n, \ldots, X_5^n | C^n)$$
$$= \sum_{C^n} P(C^n) \prod_i P(X_i^n | C^n)$$

X_1	X_2	X_3	X_4	X_5	C

- With hidden data imposed by C, it is computationally infeasible to compute
  - Maximum likelihood parameters
  - Expected parameters (or max. posterior)
  - Marginal likelihood
- Model ”structure” learning: how many values for C?

K-Means

- Normally, a geometric clustering algorithm
- A probabilistic version:
  1. Start with a random initial clustering $c_1,\ldots,c_n$
  2. Build a model Θ using complete data (X^n,C^n)
  3. Using Θ, assign each data vector X *independently* to it's most probable cluster (i.e., find max $P(C_i \mid X_i, \Theta)$ for all i)
  4. Go to 2.

Expectation Maximization (EM)

- A ”soft” version of K-Means
- Intuitively: data vectors are assigned ”fractionally” to each cluster (with the fractions determined by the classification probabilities)
- The new model Θ is computed from semi-complete data (fractional sufficient statistics)
- For HMMs: the Baum-Welch algorithm

K-Means and EM in practice

- Both provably monotonically improve the likelihood (or posterior), so they converge to a local optimum only
- Convergence can be slow
- To get reasonable results, need to repeat several runs from different starting points
- Can be used together: e.g., first run K-means, then continue with EM
- Can be used to find good starting points for other heuristics

Structure learning with FMM's

- Can find models Θ using different number of values for the hidden variable (different number of parameters)
- Which Θ to choose? (max. likelihood chooses always the model obtained with the highest number of parameters)
- Computing the marginal likelihood not feasible with the missing data imposed by the hidden variable

$$P(K|D) \propto P(D|K)P(K)$$
$$P(D|K) = \int P(D|K,\theta)P(\theta|K)\,d\theta$$
$$P(D|K,\theta) = \prod_i \sum_{k=1}^{K} P(d_i|c_k,\theta)P(c_k|\theta)$$

Approximating the marginal likelihood

- Laplace (Gaussian) approximation
- Bayesian Information Criterion (BIC)
- Akaike Information Criterion (AIC)
- Missing data completion
- Stochastic methods (MCMC etc.)
- Variational methods

Laplace's method / Gaussian approximation

- Based on Taylor approximation at the maximum likelihood parameters:

$$-\log P(D|M) \approx -\log P(D|M,\hat{\theta}) - \log P(\hat{\theta}|M) + \frac{d}{2}\log\frac{n}{2\pi} + \log\sqrt{|I(\hat{\theta})|}$$

- Here ”d” is the number of parameters, ”n” is the size of the data, and $|I(\Theta)|$ is the determinant of the Fisher information matrix at Θ
- A ”penalized log-likelihood” criterion: likelihood grows with more complex models, but it compensated by the penalizing factors
- Jeffreys' prior: $P(\theta|M) = \frac{\sqrt{|I(\theta)|}}{\int \sqrt{|I(\theta)|}\, d\theta}$

BIC and AIC

- BIC: $$-\log P(D|M) \approx -\log P(D|M,\hat{\theta}) + \frac{d}{2}\log n$$
- AIC: $$-\log P(D|M) \approx -\log P(D|M,\hat{\theta}) + d$$
- Both converge asymptotically to the marginal likelihood (minus a constant)
- Hence marginal likelihood is also in a sense a penalized maximum likelihood criterion!
- It is a non-trivial problem to determine the ”correct” value of d

Missing data completion

- Direct marginalization not feasible:

$$P(X^n|M)=\sum_{C^n} P(X^n, C^n|M)=\sum_{C^n} P(X^n|C^n, M)P(C^n|M)$$

- C^n is like an unknown "parameter"
- If you cannot marginalize over a parameter, you can try to maximize it

$$P(X^n|M)\propto max_{C^n} P(X^n|C^n, M)P(C^n|M)$$

- As the "parameter" C^n is actually data, it is easy to think of reasonable "priors" $P(C^n \mid M)$
- With fixed M, C^n can be optimized with K-means, EM, or whatever...

Supervised BN Learning

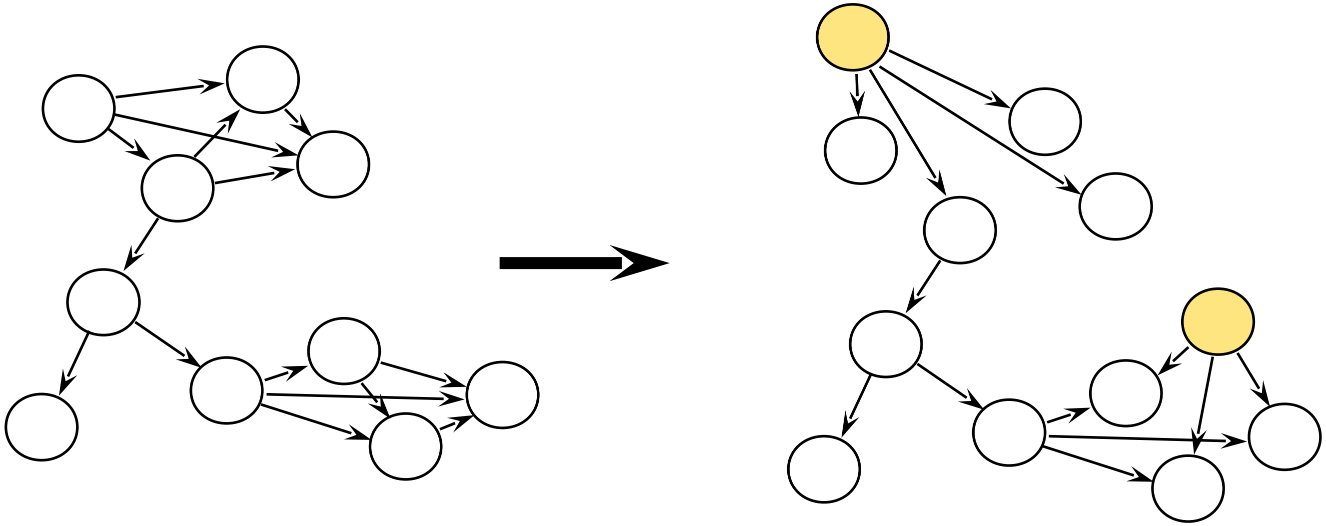
- Parameter learning
  - Generative modeling: Find $arg\ max_{\theta} P(X^n, C^n | M, \theta)$
  - Discriminative modeling: Find $arg\ max_{\theta} P(C^n | X^n, M, \theta)$
  - In general, the result is not the same!
- Structure learning
  - Generative modeling: Find $arg\ max_{M} P(X^n, C^n | M)$
  - Discriminative modeling: Find $arg\ max_{M} P(C^n | X^n, M)$
  - In general, the result is not the same!
  - Marginal conditional likelihood not feasible
    - Kontkanen et al. (UAI 1999): approximations, connection to cross-validation

Optimizing the conditional likelihood

- Bad news: even for the Naive Bayes model, the maximum of the conditional likelihood cannot be presented in closed form
- Good news: For some Bayesian networks (e.g., NB and TAN), the the conditional log-likelihood space is *concave* (Roos et al., MLJ 2005) → it has a single global optimum
- ”Supervised” Naive Bayes = logistic regression
- For model structure learning: marginal conditional likelihood not feasible (Kontkanen et al., UAI 1999)

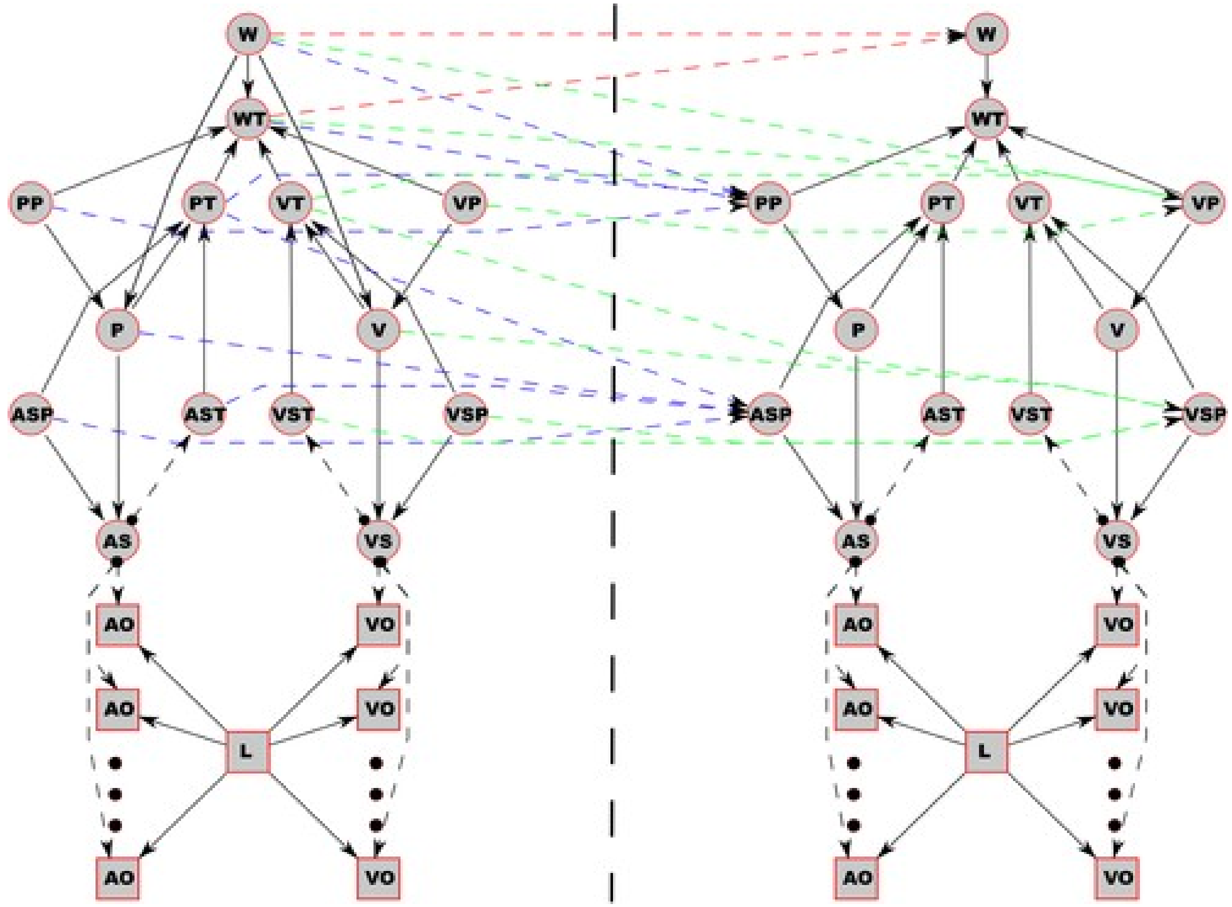
Models with many hidden nodes

- Is it sensible to first learn a Bayesian network (NP-hard) and then try to transform it to a simpler representation for probabilistic inference (NP-hard)?
- How about learning directly structures where inference is easy?



Dynamic Bayesian networks

- Complex Markov models involving temporal dependencies



Undirected Graphical Models

Definitions of independence

- Following definitions equivalent for $X1 \perp X2 \mid Z$:
  - $p(X1,X2 \mid Z) = p(X1 \mid Z)p(X2 \mid Z)$ whenever $p(Z)>0$
  - $p(X1 \mid X2,Z) = p(X1 \mid Z)$ whenever $p(X2,Z)>0$
  - $p(X2 \mid X1,Z) = p(X2 \mid Z)$ whenever $p(X1,Z)>0$
  - $p(X1,X2,Z) = f(X1,Z)g(X2,Z)$ for non-negative functions $f(\cdot),g(\cdot)$
- Definitions symmetric in X1 and X2

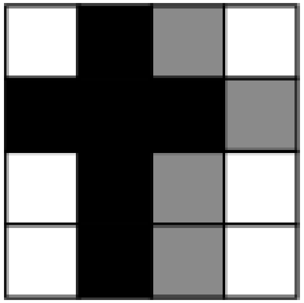
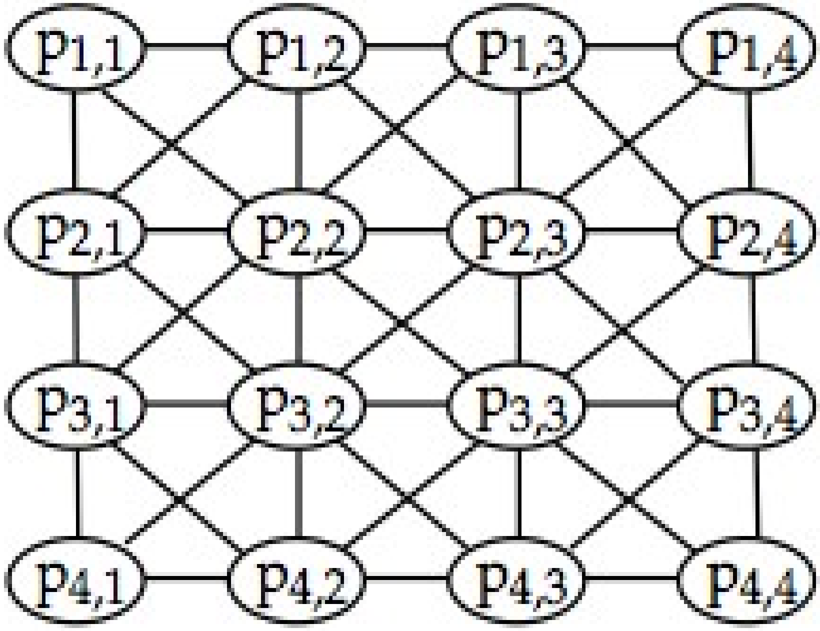


Image models

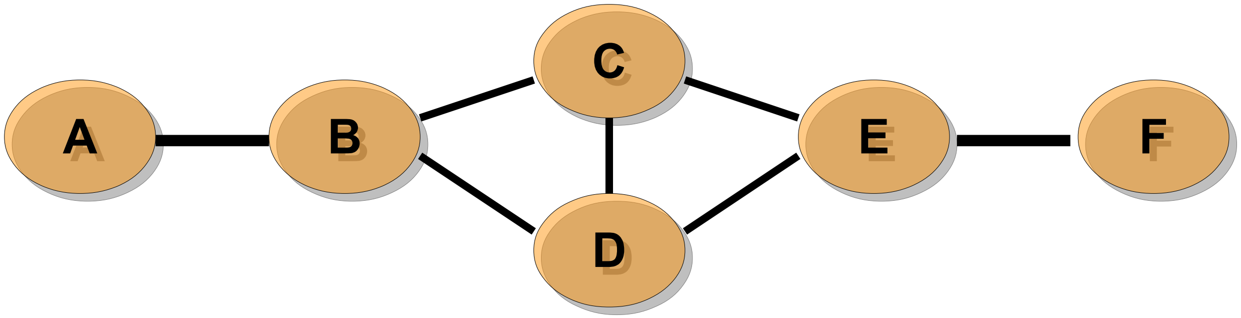
- The graph on the right says that each pixel is influenced only by its neighbors



Undirected graphical models

- Local Markov property:
  - $X \perp (G\text{-nbrs}(X)-\{X\}) \mid \text{nbrs}(X)$
  - Minimal independence properties to uniquely determine a graph
- Global Markov property:
  - For all X_1,X_2,Z: $X_1 \perp X_2 \mid Z$ iff X_1 is separated in the graph from X_2 by Z.
  - How to test for independence
- Functional form: $P(X_1,\ldots,X_n)=\prod_C f_C(X_C)$
  - Product over cliques C (X_C denoting the members of the clique)
  - Definition for purposes of computation

For example...



- Local Markov property:
  - E.g.: B ⊥ E,F | A,C,D; C ⊥ A,F | B,D,E;...
- Global Markov property:
  - E.g.: A,B ⊥ E,F | C,D.
- Functional form:
  - P(A,B,C,D,E)=e(A,B)f(B,C,D)g(C,D,E)h(E,F)

The three properties are equivalent

- Global Markov property implies the local
- Functional form implies the global Markov property
- Hammersley-Clifford theorem: Local Markov property implies the functional form (for discrete variables)

Markov Random Fields

- Undirected graphical models, a.k.a. Markov networks
- Typically use alternative functional form:

$$P(X) = \frac{1}{Z} \exp\left(\sum_C \alpha_C f_C(X_C)\right)$$

- Sometimes also called the Gibbs distribution
- The cliquewise functions f_C are called *clique potentials*
- The normalizer Z is called the *partition function*

Mapping a DAG to a MRF is possible...

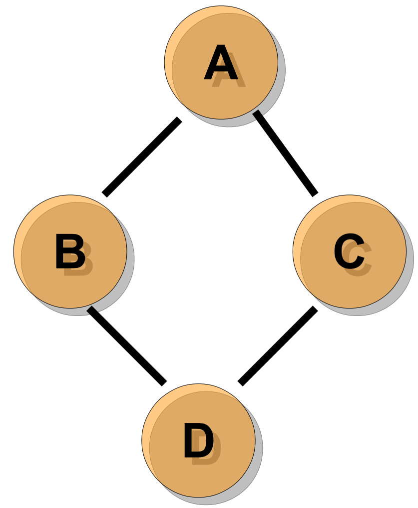
- Mapping is straightforward if a node and its parents in a DAG belong to the same clique in the MRF

$$\prod_i P(X_i | Pa_i) \rightarrow \prod_C f_C(X_C)$$

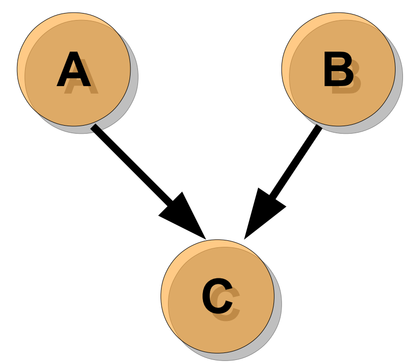
- This means that to get the corresponding MRF, we need to ”marry” nodes with common children (this is called *moralizing* the graph)
- It follows that inference in undirected graphs is NP-hard too...

...but DAGs and MRFs are not equivalent independence models

- $A \perp D \mid B,C$ and
  $B \perp C \mid A,D$



- $A \perp B$ and
  $A \not\perp B \mid C$



Final remarks

- The Bayesian framework offers an elegant, consistent formalism for uncertain reasoning
- The basic principle is simple: compute the probability of what you want to know while marginalizing over the other unknown factors
- We have focused on the discrete Dirichlet-multinomial case and directed acyclic graphs (Bayesian networks), but the same principles apply with other probabilistic model families as well
- Graphical models offer a unifying framework where many popular methods are easily understood
  - E.g. Factor analysis, PCA, ICA, mPCA, HMM, Kalman filter, switching Kalman filter, AR models,...
  - See: http://www.cs.ubc.ca/~murphyk/Bayes/bnintro.html