Completely Observed Graphical Models

Prof. Nicholas Zabaras
Center for Informatics and Computational Science
https://cics.nd.edu/
University of Notre Dame
Notre Dame, IN, USA

Email: nzabaras@gmail.com
URL: https://www.zabaras.com/

April 12, 2018

Contents

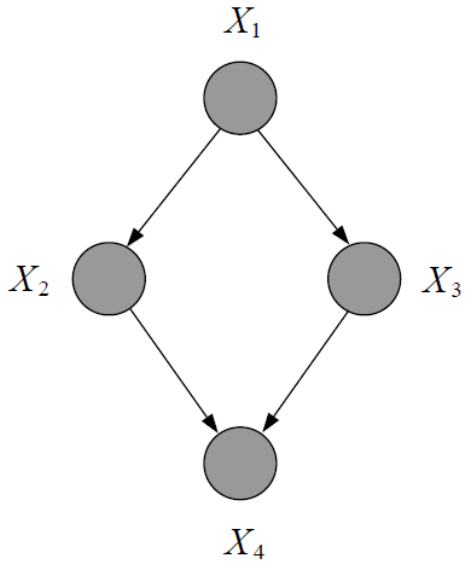
- Kevin Murphy, Machine Learning: A probabilistic Perspective, Chapter 19
- Jordan, M. I. (2007). An introduction to probabilistic graphical models. In preparation (Chapter 9).

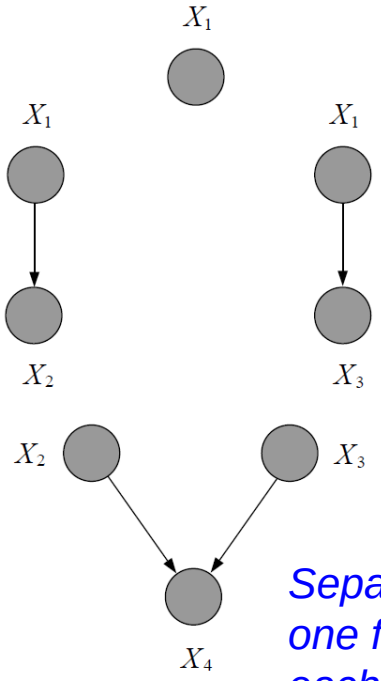
Completely Observed Graphical Models

- We first consider directed graphical models under the assumption of complete observations.
  - Thus *we assign values to all of the random variables in the model (no latent variables)*
- *For directed graphical models, the parameter estimation problem decouples* - we can compute the MLE of parameters by solving the problem separately at each node of the graph.
- *For undirected graphical models, the MLE computation decouples only for decomposable models.*
  - Complication arising from the presence of the global normalization factor Z.
  - There is however a local characterization of MLE for general undirected models.

Completely Observed Graphical Models

- Consider the directed model shown. *Computing the MLE of the parameters of the model breaks into separate MLE problems one for each node conditioned on its parents.*





Separate MLE one for each node

$$p(\boldsymbol{x}|\theta) = p(x_1|\theta_1)p(x_2|x_1,\theta_2)p(x_3|x_1,\theta_3)p(x_4|x_2,x_3,\theta_4)$$

$$\log p(\mathbf{x}|\theta) = \log p(x_1|\theta_1) + \log p(x_2|x_1,\theta_2) + \log p(x_3|x_1,\theta_3) + \log p(x_4|x_2,x_3,\theta_4)$$

- As a result of this decomposition of the joint probability distribution, *maximization of the log probability wrt the parameters, θ_i, for a given i, can be carried out independently of the other maximizations (see plot on the right).*

Directed Models

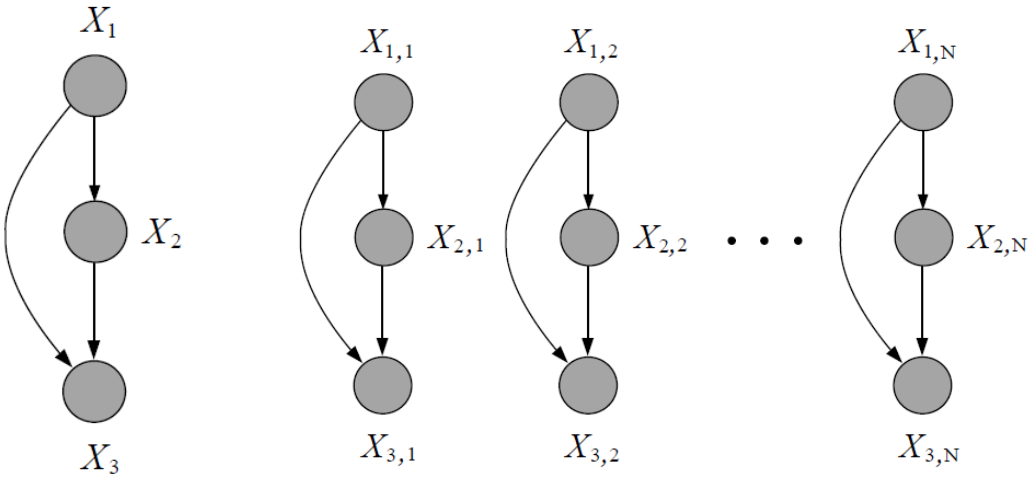
- Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ a directed graph. $\mathcal{V}$ set of nodes, $\mathcal{E}$ set of edges.
- X_u is the rv associated with $u \in \mathcal{V}$, x_u denotes a realization of X_u.
- X_C refers to the set of components indexed by a subset $C \subseteq \mathcal{V}$. *The vector X (all rvs in the graph) is also written as $X_{\mathcal{V}}$.*
- For each node $u \in \mathcal{V}$, we associate $p(x_u|x_{\pi_u}, \boldsymbol{\theta}_u)$. The overall probability associated with the graph $\mathcal{G}$:

$$p(x_{\mathcal{V}}|\theta) = \prod_{u \in \mathcal{V}} p\left(x_u|x_{\pi_u}, \boldsymbol{\theta}_u\right), \; where: \; \boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_m).$$

- Assume complete observation of all of the rvs $X_{\mathcal{V}}$.
- The data consists of N IID observations. *We replicate the graphical model N times - the result is itself a graphical model.*
- We need a way to refer to the overall model that assigns probability to the IID replicates of $X_{\mathcal{V}}$, while we continue referring to the probability model associated with a single observable vector $X_{\mathcal{V}}$ regardless of the sampling process.

N-Disconnected Replicas of $\mathcal{G}$

- We use the notation $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and $p(X_{\mathcal{V}}|\boldsymbol{\theta})$ to refer to the probability model associated with a single observable vector $X_{\mathcal{V}}$.
- We also construct an augmented graphical model, $\mathcal{G}^{(N)} = (\mathcal{V}^{(N)}, \mathcal{E}^{(N)})$ that incorporates the IID sampling assumption (N *disconnected replicates* of $\mathcal{G}$).
- The nodes $\mathcal{V}^{(N)}$ in the augmented graph are indexed as (u, n), where $u \in \mathcal{V}$ designates a node in the underlying graphical model $\mathcal{G}$, and $n \in \{1,2,..,n\}$ is the replication number.
- Similarly, (C, n) denotes the nth replicate of the set C, for $C \subseteq V$. In particular, $(X_{\mathcal{V}}, n)$ denotes the nth replicate of $X_{\mathcal{V}}$



Probability Model for the Graph $\mathcal{G}^{(N)}$

- The observed data are:

$$\mathcal{D} = \left(x_{\mathcal{V},1},\ x_{\mathcal{V},2},\ldots,x_{\mathcal{V},N}\right)$$

- The probability model for the graph $\mathcal{G}^{(N)}$ is:

$$p(\mathcal{D}|\boldsymbol{\theta}) = \prod_n p\left(x_{\mathcal{V},n}|\boldsymbol{\theta}\right) = \prod_n \prod_u p\left(x_{u,n}|x_{\pi_u,n},\boldsymbol{\theta}_u\right)$$

- Taking the logs:

$$\log p(\mathcal{D}|\boldsymbol{\theta}) = \sum_n \sum_u \log p\left(x_{u,n}|x_{\pi_u,n},\boldsymbol{\theta}_u\right)$$

- The log likelihood is a sum of a number of terms, each of which refers to only one of the parameter vectors $\boldsymbol{\theta}_u$. In estimating $\boldsymbol{\theta}_u$ we can ignore all terms that involve $\boldsymbol{\theta}_{u'}$, for $u' \neq u$.
- The terms involving u refer only to x_u and x_{π_u}. For estimating $\boldsymbol{\theta}_u$, we need only focus on the data associated with the node u and its parents. *The observations* $\{x_{u,n}, x_{\pi_u,n}\}, n = 1,..N$ *associated with these nodes are sufficient for* $\boldsymbol{\theta}_u$.

Probability Model for the Graph $\mathcal{G}^{(N)}$

- If the observations $\{x_{u,n}, x_{\pi_u,n}\}\, n = 1,..N$ can be summarized by finite-dimensional sufficient statistics, then the entire estimation problem can be reduced to a collection of finite-dimensional sufficient statistics.

- *This is the case if each $p(x_u|x_{\pi_u},\boldsymbol{\theta}_u)$ is an exponential family distribution.*

- We thus *reduce our problem from observations associated with the $\mathcal{G}^{(N)}$ to statistics associated with $\mathcal{G}$*, and even further to *statistics associated with single nodes and their parents!*

- We next apply these ideas to discrete models.

Discrete Models: Counts & Marginal Counts

- Counts are the sufficient statistics for multinomial rvs. For a given conguration $x_{\mathcal{V}}$, let $m(x_{\mathcal{V}})$ denote the number of times that $x_{\mathcal{V}}$ is observed in the dataset $\mathcal{D}$.
- There are only a finite number of possible configurations $X_{\mathcal{V}}$, and each data point $X_{\mathcal{V},n}$ must be one of these configurations:

$$m(x_{\mathcal{V}}) = \sum_{n} \delta(x_{\mathcal{V}}, x_{\mathcal{V},n})$$

- Also define *marginal counts associated with subsets of nodes*.
- For any given subset C, let $m(x_C)$ denote the number of times that configuration x_C is observed in the data set.

$$m(x_C) = \sum_{x_{\mathcal{V}\setminus C}} m(x_{\mathcal{V}})$$

- Let $\mathcal{V} = \{1, 2, 3\}$. Then, $m(X_{\mathcal{V}})$ can be represented as a 3D table (each entry: how many times that configuration appears in the data). Let the parent of X_2 be X_1. Compute $m(x_1, x_2)$ or $m(x_1)$ as:

$$m(x_1, x_2) = \sum_{x_3} m(x_1, x_2, x_3)\,,\; m(x_1) = \sum_{x_2} m(x_1, x_2) = \sum_{x_2, x_3} m(x_1, x_2, x_3)$$

Discrete Models: Counts & Marginal Counts

- Note also that if we sum over all three variables we obtain the scalar N, the total number of observations.

$$\sum_{x_1,x_2,x_3} m(x_1,x_2,x_3) = N$$

- A particular subset of interest is the subset consisting of a node u and its parents π_u (the family $\phi_u = \{u\} \cup \pi_u$ associated with node u). Then:

$$m\left(x_{\varphi_u}\right) = \sum_{x_{\mathcal{V}\setminus\varphi_u}} m\left(x_{\mathcal{V}}\right)$$

The sum is over all nodes in $\mathcal{V}$ other than u and its parents π_u

- $m\left(x_{\varphi_u}\right)$ *is the count of the number of times a node u and its parents take on a specific configuration.*

The Joint Probability

- Consider the joint probability distribution in the discrete case.
- A separate parameter is associated with each possible joint configuration of a node and its parents. Define the parameter vector $\theta_v\left(x_{\varphi_v}\right)$ to be a nonnegative, multidimensional table indexed by the joint configuration of v and π_v.

$$p\left(x_v|x_{\pi_v},\theta_v\right) \triangleq \theta_v\left(x_{\varphi_v}\right)$$

- The normalization condition requires:

$$\sum_{x_v}\theta_v\left(x_{\varphi_v}\right)=\sum_{x_v}\theta_v\left(x_v,x_{\pi_v}\right)=1,\ \varphi_v=\{v\}\cup\pi_v$$

- Taking the product over v, we obtain the joint distribution:

$$p(x_V|\boldsymbol{\theta})=\prod_v p\left(x_v|x_{\pi_v},\theta_v\right)=\prod_v \theta_v\left(x_{\varphi_v}\right)$$

- The total probability of $\mathcal{D}=\ x_{\mathcal{V},1},\ x_{\mathcal{V},2},\ldots,\ x_{\mathcal{V},N}$ is then:

$$p(\mathcal{V}|\boldsymbol{\theta})=\prod_n p\left(x_{\mathcal{V},n}|\boldsymbol{\theta}\right),\ p\left(x_{\mathcal{V},n}|\boldsymbol{\theta}\right)=\prod_{x_{\mathcal{V}}} p(x_{\mathcal{V}}|\boldsymbol{\theta})^{\delta\left(x_{\mathcal{V}},x_{\mathcal{V},n}\right)}$$

Go through all configurations & pick the one that coincides with $X_{\mathcal{V},n}$

Log-Likelihood

$$p(\mathcal{D}|\boldsymbol{\theta}) = \prod_n p\left(x_{\mathcal{V},n}|\boldsymbol{\theta}\right),\ p\left(x_{\mathcal{V},n}|\boldsymbol{\theta}\right) = \prod_{x_\mathcal{V}} p(x_\mathcal{V}|\boldsymbol{\theta})^{\delta\left(x_\mathcal{V},x_{\mathcal{V},n}\right)},\ p(x_V|\boldsymbol{\theta}) = \prod_v \theta_v\left(x_{\varphi_v}\right)$$

- The dummy variable $x_\mathcal{V}$ ranges across configurations of the nodes. Using this representation we write the joint probability as

$$\log p(\mathcal{D}|\boldsymbol{\theta}) = \sum_n \log p\left(x_{V,n}|\boldsymbol{\theta}\right) = \sum_n \sum_{x_\mathcal{V}} \log p(x_\mathcal{V}|\boldsymbol{\theta})^{\delta\left(x_\mathcal{V},x_{\mathcal{V},n}\right)}$$

$$= \sum_{x_\mathcal{V}} \left(\sum_n \delta\left(x_\mathcal{V}, x_{\mathcal{V},n}\right) \right) \log p(x_\mathcal{V}|\boldsymbol{\theta}) = \sum_{x_\mathcal{V}} m(x_\mathcal{V}) \log p(x_\mathcal{V}|\boldsymbol{\theta})$$

- The sum over n has disappeared and the representation of joint probability from a function on $\mathcal{G}^{(N)}$ has been reduced to a function on $\mathcal{G}$. Finally, we can write:

$$\log p(\mathcal{D}|\boldsymbol{\theta}) = \sum_{x_\mathcal{V}} m(x_\mathcal{V}) \log p(x_\mathcal{V}|\boldsymbol{\theta}) = \sum_{x_\mathcal{V}} m(x_\mathcal{V}) \log \prod_v \theta_v\left(x_{\varphi_v}\right) = \sum_{x_\mathcal{V}} m(x_\mathcal{V}) \sum_v \log \theta_v\left(x_{\varphi_v}\right)$$

$$= \sum_v \sum_{x_{\varphi_v}} \sum_{x_{V \setminus \varphi_v}} m(x_V) \log \theta_v\left(x_{\varphi_v}\right) = \sum_v \sum_{x_{\varphi_v}} m\left(x_{\varphi_v}\right) \log \theta_v\left(x_{\varphi_v}\right)$$

This is sum of terms over families

MLE Estimate

$$\log p(\mathcal{D}|\boldsymbol{\theta}) = \sum_{v}\sum_{x_{\varphi_v}} m(x_{\varphi_v})\log\theta_v(x_{\varphi_v}) \Rightarrow p(\mathcal{D}|\boldsymbol{\theta}) = \exp\left(\sum_{v}\sum_{x_{\varphi_v}} m(x_{\varphi_v})\log\theta_v(x_{\varphi_v})\right)$$

- We have an exponential family distribution for $\mathcal{D}$ with sufficient statistics $m(x_{\varphi_v})$ and $\log\theta_v(x_{\varphi_v})$ as the natural parameters.

- The parameters $\theta_v(x_{\varphi_v})$ appear in separate terms which can be optimized independently of each other. To estimate $\theta_v(x_{\varphi_v})$, we maximize wrt $\theta_v(x_{\varphi_v})$ the following:

$$m(x_{\varphi_v})\log\theta_v(x_{\varphi_v}) + \lambda\left(1 - \sum_{x_v}\theta_v(x_{\varphi_v})\right)$$

- This gives:

$$\hat{\theta}_{v,ML}(x_{\varphi_v}) = \frac{m(x_{\varphi_v})}{m(x_{\pi_v})} = \frac{m(x_v, x_{\pi_v})}{m(x_{\pi_v})}$$

- *(As expected), the MLE is the ratio of the count of times a node and its parents are jointly in a specific configuration to the count of the number of times its parents are in that configuration.*

- These estimates are formed independently at each nodes in $\mathcal{G}$.

MLE Estimate using IRLS

- Tabular representations are useful for graphs with small families (the size of a table is exponential in the number of parents).
- *For large families, we wish to constrain* $p\left(x_\upsilon \mid x_{\pi_\upsilon}, \theta_\upsilon\right)$ *-- use e.g. generalized linear models (GLIMs).*
- For GLIMs can use the IRLS algorithm for obtaining parameter estimates.
- The decoupling of the log likelihood implies that if each local conditional model is a GLIM model, then we solve the MLE problem for the graph as a whole by
  - *running the IRLS algorithm separately at each node.*
- For completely observed data, *any solution scheme to the problem of estimating parameters at a single node (conditional on its parents) applies immediately to general directed graphs.*

Undirected Models

- Undirected models are more flexible than their directed counterparts.
- Potentials in undirected models *are unnormalized functions* defined on arbitrary subsets of nodes.
- Undirected graphical models require an explicit global normalization factor - the $1/Z$ factor in the joint probability.
- *Z couples the parameters and complicates the parameter estimation problem*. However, *for decomposable models the parameter estimation problem decouples.*
- For general undirected models effective parameter estimation algorithms that exploit the structure of the graph are available.
- Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ the set of edges.

Discrete Models

- We parameterize via a set of clique potentials

$$p(x_{\mathcal{V}}|\boldsymbol{\theta}) = \frac{1}{Z}\prod_{C}\psi_C(x_C)\,,\; C \in \mathcal{C},\; \boldsymbol{\theta} = \{\psi_C(x_C),\; C \in \mathcal{C}\}$$

- Note that we do not assume that $\mathcal{C}$ contains all of the cliques in the graph, nor that the cliques in $\mathcal{C}$ are maximal.

- The normalization factor Z is given as $Z = \sum_{x_{\mathcal{V}}}\prod_{C}\psi_C(x_C)$

- We continue by considering discrete models below.

$$p(\mathcal{D}|\boldsymbol{\theta}) = \prod_{n} p\left(x_{\mathcal{V},n}|\boldsymbol{\theta}\right),\; p\left(x_{\mathcal{V},n}|\boldsymbol{\theta}\right) = \prod_{x_{\mathcal{V}}} p(x_{\mathcal{V}}|\boldsymbol{\theta})^{\delta\left(x_{\mathcal{V}},x_{\mathcal{V},n}\right)}$$

$$log p(\mathcal{D}|\boldsymbol{\theta}) = \sum_{n}\sum_{x_{\mathcal{V}}} \log p(x_{\mathcal{V}}|\boldsymbol{\theta})^{\delta\left(x_{\mathcal{V}},x_{\mathcal{V},n}\right)}$$

$$= \sum_{x_{\mathcal{V}}}\left(\sum_{n}\delta\left(x_{\mathcal{V}},x_{\mathcal{V},n}\right)\right)\log p(x_{\mathcal{V}}|\boldsymbol{\theta}) = \sum_{x_{\mathcal{V}}} m(x_{\mathcal{V}})\log p(x_{\mathcal{V}}|\boldsymbol{\theta}) = \sum_{x_{\mathcal{V}}} m(x_{\mathcal{V}})\log\left(\frac{1}{Z}\prod_{C}\psi_C(x_C)\right)$$

$$= \sum_{x_{\mathcal{V}}} m(x_{\mathcal{V}})\sum_{C}\log\psi_C(x_C) - \sum_{x_{\mathcal{V}}} m(x_{\mathcal{V}})\, logZ = \sum_{x_{\mathcal{V}}} m(x_{\mathcal{V}})\sum_{C}\log\psi_C(x_C) - N\, logZ$$

Discrete Models

$$logp(\mathcal{D}|\boldsymbol{\theta}) = \sum_{x_V} m(x_V) \sum_{C} \log\psi_C(x_C) - N\ logZ$$

$$logp(\mathcal{D}|\boldsymbol{\theta}) = \sum_{C} \sum_{x_C} m(x_C) \log\psi_C(x_C) - N\ logZ$$

- *The marginal counts $m(x_C)$, for $C \in \mathcal{C}$, are the sufficient statistics for our model.*
- This is similar to the result for directed models where the cliques C were the families $\{\phi_n\}$.
- *Important difference between the undirected log likelihood and its directed counterpart:* ***the appearance of the term N log Z.***

MLE Estimation

$$logp(\mathcal{D}|\boldsymbol{\theta}) = \sum_{C}\sum_{x_C} m(x_C)\log\psi_C(x_C) - N\ logZ$$

- To find the MLE, we take derivatives of the log likelihood and set to zero.
- Due to the N log Z term, the calculation yields a *coupled, nonlinear set of eqs* in which the parameters appear implicitly.
- The implicit eqs reveal a local property of the MLE that provides the inspiration for *an iterative algorithm for finding the parameter estimates.*
- In the MLE calculation, $\psi_C(x_C)$ is the independent variable, where both *the clique C and the configuration x_C have been fixed* (in the discrete case, this picks out the cell indexed by x_C in the table representing the potential function on clique C).

$$\frac{\partial \log p(\mathcal{D}|\boldsymbol{\theta})}{\partial \psi_C(x_C)} = \frac{m(x_C)}{\psi_C(x_C)} - N\frac{\partial \log Z}{\partial \psi_C(x_C)} = \frac{m(x_C)}{\psi_C(x_C)} - N\frac{1}{Z}\frac{\partial}{\partial \psi_C(x_C)}\left(\sum_{\tilde{x}}\prod_{D}\psi_D(\tilde{x}_D)\right)$$

MLE Estimation

$$\frac{\partial \log p(\mathcal{D}|\boldsymbol{\theta})}{\partial \psi_C(x_C)} = \frac{m(x_C)}{\psi_C(x_C)} - N\frac{1}{Z}\frac{\partial}{\partial \psi_C(x_C)}\left(\sum_{\tilde{x}}\prod_D \psi_D(\tilde{x}_D)\right) =$$

$$= \frac{m(x_C)}{\psi_C(x_C)} - N\frac{1}{Z}\sum_{\tilde{x}}\delta(\tilde{x}_C, x_C)\frac{\partial}{\partial \psi_C(\tilde{x}_C)}\left(\prod_D \psi_D(\tilde{x}_D)\right)$$

$$= \frac{m(x_C)}{\psi_C(x_C)} - N\frac{1}{Z}\sum_{\tilde{x}}\delta(\tilde{x}_C, x_C)\left(\prod_{D\neq C} \psi_D(\tilde{x}_D)\right)$$

$$= \frac{m(x_C)}{\psi_C(x_C)} - N\sum_{\tilde{x}}\delta(\tilde{x}_C, x_C)\frac{1}{\psi_C(\tilde{x}_C)}\frac{1}{Z}\left(\prod_D \psi_D(\tilde{x}_D)\right)$$

$$= \frac{m(x_C)}{\psi_C(x_C)} - N\frac{1}{\psi_C(x_C)}\sum_{\tilde{x}}\delta(\tilde{x}_C, x_C)p(\tilde{x})$$

$$= \frac{m(x_C)}{\psi_C(x_C)} - N\frac{p(x_C)}{\psi_C(x_C)}$$

- Setting this to zero we obtain: $\hat{p}_{ML}(x_C) = \frac{m(x_C)}{N}$

MLE Estimation

$$\hat{p}_{ML}(x_C) = \frac{m(x_C)}{N}$$

- Let us define the empirical distribution

$$\tilde{p}(x) = \frac{m(x)}{N}$$

- Then *the marginal under the empirical distribution* is:

$$\tilde{p}(x_C) = \frac{m(x_C)}{N}$$

- Thus we have the following important characterization of MLE for each clique $C \in \mathcal{C}$: *the model marginals must be equal to the empirical marginals.*

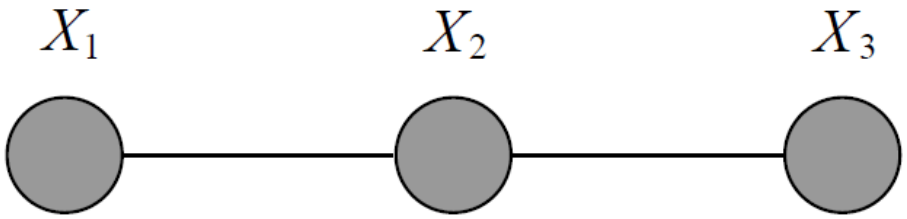
$$\hat{p}_{ML}(x_C) = \tilde{p}(x_C)$$

MLE Estimation

$$\hat{p}_{ML}(x_C) = \tilde{p}(x_C)$$

- This result constrains maximum likelihood models.
- The eq. above provides us with a system of equations (by ranging over all $C \in \mathcal{C}$) that constrains the maximum likelihood estimates, but the parameters $\psi_C(x_C)$ themselves appear implicitly in these equations.
- We will see soon that for "decomposable graphs", the MLE problem decouples, and we can write down MLE estimates by inspection.

Decomposable Models



- Consider the example above with

$$p(x_{\mathcal{V}}) = \frac{1}{Z}\psi_{12}(x_1, x_2)\psi_{23}(x_2, x_3)$$

- Based on our earlier MLE calculations, the sufficient statistics under IID sampling are the empirical marginals $\tilde{p}(x_1, x_2)$, $\tilde{p}(x_2, x_3)$

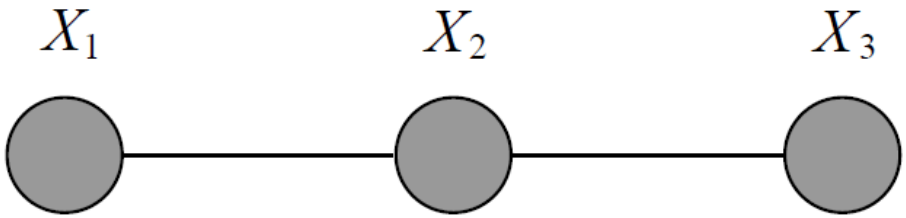
- For a model to be a maximum likelihood model, we require the marginals to equal the empirical marginals. How do we set the parameters so as to achieve this result?

- Let us make the following guess:

$$\hat{p}_{ML}(x_1, x_2, x_3) = \frac{\tilde{p}(x_1, x_2)\tilde{p}(x_2, x_3)}{\tilde{p}(x_2)}, \quad \tilde{p}(x_2) = \sum_{x_1} \tilde{p}(x_1, x_2) = \sum_{x_3} \tilde{p}(x_2, x_3)$$

- That this is a good guess is readily verified.

Decomposable Models



$$\hat{p}_{ML}(x_1, x_2, x_3) = \frac{\tilde{p}(x_1, x_2)\tilde{p}(x_2, x_3)}{\tilde{p}(x_2)}$$

$$\hat{p}_{ML}(x_1, x_2) = \sum_{x_3} \hat{p}_{ML}(x_1, x_2, x_3) = \tilde{p}(x_1, x_2)$$

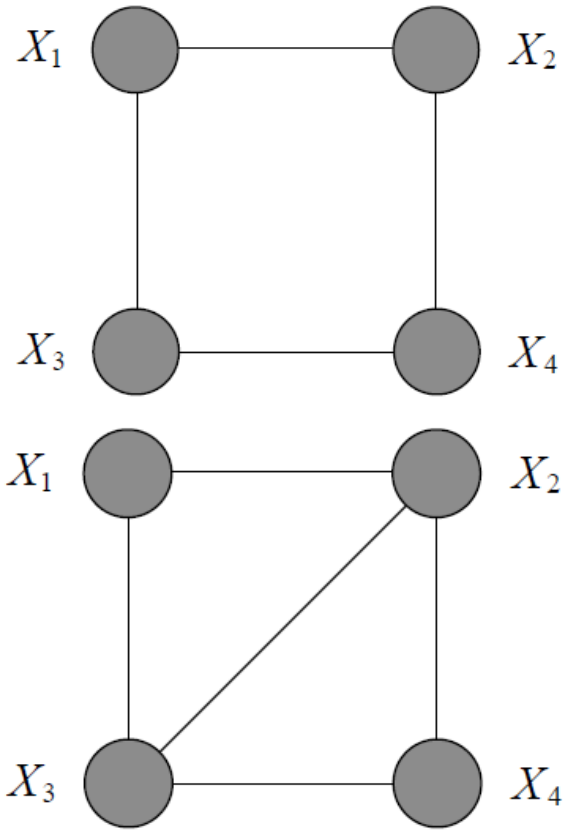
$$\hat{p}_{ML}(x_2, x_3) = \sum_{x_1} \hat{p}_{ML}(x_1, x_2, x_3) = \tilde{p}(x_2, x_3)$$

- The model marginals are indeed equal to the empirical marginals on the cliques C.
- Moreover, we can also easily match the terms in the guessed distribution to the parameters. E.g., we can let:

$$\hat{\psi}_{12,ML}(x_1, x_2) = \tilde{p}(x_1, x_2),\ \hat{\psi}_{23,ML}(x_2, x_3) = \frac{\tilde{p}(x_2, x_3)}{\tilde{p}(x_2)},\ Z = 1$$

- There are many other sets of parameter estimates that yield the same joint distribution; these are all MLE estimates.
- Can we generalize this approach?

Decomposable Models



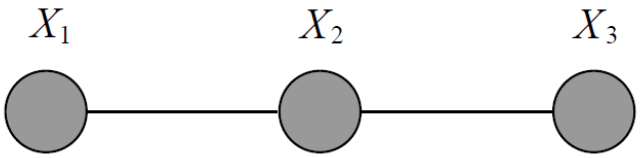
- That the earlier approach cannot work in general, is shown by the graph on the left.
- The analogous ratio of the empirical marginals does not yield a maximum likelihood distribution in this case.
- The problem is not simply due to an inability to handle graphs with loops.
- This can be seen by considering the next graph for which the following guess fits the empirical marginals correctly:

$$\hat{p}_{ML}(x_1, x_2, x_3, x_4) = \frac{\tilde{p}(x_1, x_2, x_3)\tilde{p}(x_2, x_3, x_4)}{\tilde{p}(x_2, x_3)}$$

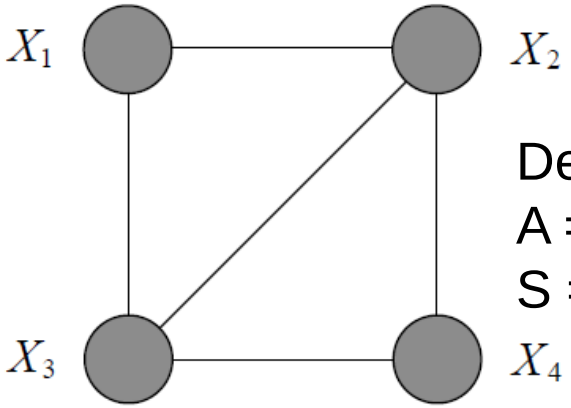
- There is a subtlety. *To recover parameter estimates from the factored form above we need to have potentials that have arguments* $\psi_{123}(x_1, x_2, x_3), \psi_{234}(x_2, x_3, x_4)$. *For other parametrizations, the above estimate is not an MLE.*

Decomposable Models

- We now discuss the underlying concept that makes it possible to write MLE estimates by inspection in some cases but not in others.
- *A graph is said to be decomposable if it can be recursively subdivided into disjoint sets A, B and S, where S separates A and B, and **where S is complete**.*

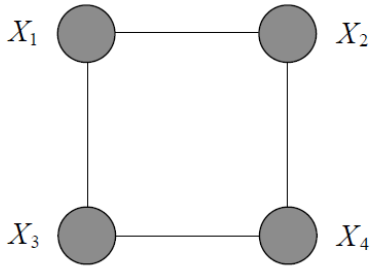


Decomposable with A = X_1, B = X_3 and S = X_2.



Decomposable with A = X_1, B = X_4 and S = {X_2 , X_3}

- You can verify that the graph on the right is not decomposable.

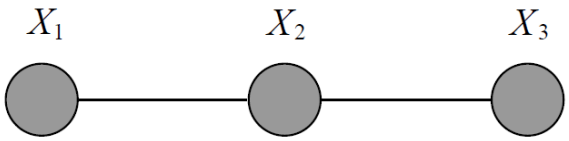


Decomposable Models

- We can find MLE estimates for decomposable graphs by inspection, but only if the potentials are defined on maximal cliques.
- That is, *our parameterization must be such that the set $\mathcal{C}$ ranges over the maximal cliques in the graph*.
- Given this constraint, we write the MLE by inspection as follows:

  - *for every clique C, set the clique potential to the empirical marginal for that clique,*

  - *for every non-empty intersection between cliques, associate an empirical marginal with that intersection, and divide that empirical marginal into the potential of one of the two cliques that form the intersection.*

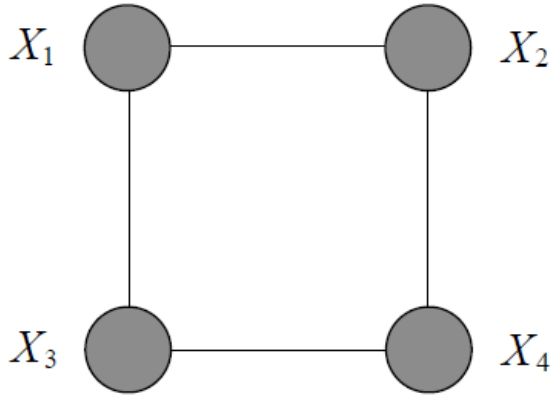


$$\hat{\psi}_{12,ML}(x_1, x_2) = \tilde{p}(x_1, x_2),$$

$$\hat{\psi}_{23,ML}(x_2, x_3) = \frac{\tilde{p}(x_2, x_3)}{\tilde{p}(x_2)}$$

Decomposable Graphs & Graph Elimination

- Note that in the graph shown is it impossible to eliminate the nodes without adding extra edges to the graph.



- This suggests *a relationship between decomposability and graph elimination.*

- Decomposability lies with elimination at the core of the relationship between graphs and probabilities.

Iterative Proportional Fitting

- We now turn to an alternative algorithm for *finding MLE in all undirected graphs, whether decomposable and nondecomposable*.
- *In the decomposable case, the algorithm turns out to converge in a finite number of iterations, updating each potential once.*
- *The algorithm* can be viewed as a general solution to the MLE problem that *takes advantage of the decomposable structure in the problem if it is present.*

Iterative Proportional Fitting

- A common strategy for solving systems of implicit equations is to iterate (hopefully) to a fixed point.
- Iterative proportional fitting (IPF), an algorithm for MLE in undirected models is an example of such a strategy.
- In general, the iteration of fixed-point equations does not converge and the algorithm may not behave well in the sense of ascending an objective function at each step.
- *IPF however does converge and behave well - the log likelihood is guaranteed to increase or remain the same after each update*.
- *IPF thus it not only a fixed-point algorithm, but also a coordinate ascent algorithm.*

Iterative Proportional Fitting

- We begin with a heuristic justification of IPF in terms of fixed-point iteration.

- To develop an iterative approach to MLE for undirected models, let us return to the gradient of the log likelihood, but retain the factors $\psi_C(x_C)$

$$\frac{\tilde{p}(x_C)}{\psi_C(x_C)} = \frac{p(x_C)}{\psi_C(x_C)}, \tilde{p}(x_C) = \frac{m(x_C)}{N}$$

- Note that the parameter $\psi_C(x_C)$ appears explicitly in this equation in two places, but also appears implicitly in the marginal $p(x_C)$.

- We can obtain an iterative algorithm by holding the values of $\psi_C(x_C)$ fixed on the right-hand side of the Eq above, both in the numerator and the denominator, and solving for the free parameter $\psi_C(x_C)$ on the left-hand side.

$$\psi_C^{(t+1)}(x_C) = \psi_C^{(t)}(x_C) \frac{\tilde{p}(x_C)}{p^{(t)}(x_C)}$$

Iterative Proportional Fitting

$$\psi_C^{(t+1)}(x_C) = \psi_C^{(t)}(x_C)\frac{\tilde{p}(x_C)}{p^{(t)}(x_C)}$$

- The IPF algorithm cycles through all of the cliques $C \in \mathcal{C}$, applying the above Eq. to each clique in turn; such a cycle constitutes a single iteration of the algorithm.

- *The IPF update equation has two interesting properties: (1) the marginal $p^{(t+1)}(x_C)$ is equal to the empirical marginal $\tilde{p}(x_C)$ and (2) the normalization factor Z remains constant across IPF updates.*

$$p_C^{(t+1)}(x_C) = \sum_{x_{V\setminus C}} p^{(t+1)}(x) = \sum_{x_{V\setminus C}} \frac{1}{Z^{(t+1)}} \prod_D \psi_D^{(t+1)}(x_D) = \frac{1}{Z^{(t+1)}} \sum_{x_{V\setminus C}} \psi_C^{(t+1)}(x_C) \prod_{D\neq C} \psi_D^{(t)}(x_D)$$

$$= \frac{1}{Z^{(t+1)}} \sum_{x_{V\setminus C}} \psi_C^{(t)}(x_C)\frac{\tilde{p}(x_C)}{p^{(t)}(x_C)} \prod_{D\neq C} \psi_D^{(t)}(x_D) = \frac{Z^{(t)}}{Z^{(t+1)}}\frac{\tilde{p}(x_C)}{p^{(t)}(x_C)} \sum_{x_{V\setminus C}} \frac{1}{Z^{(t)}} \prod_D \psi_D^{(t)}(x_D)$$

$$= \frac{Z^{(t)}}{Z^{(t+1)}}\frac{\tilde{p}(x_C)}{p^{(t)}(x_C)} \sum_{x_{V\setminus C}} p^{(t)}(x) = \frac{Z^{(t)}}{Z^{(t+1)}}\frac{\tilde{p}(x_C)}{p^{(t)}(x_C)} p^{(t)}(x_C) = \frac{Z^{(t)}}{Z^{(t+1)}}\tilde{p}(x_C)$$

Iterative Proportional Fitting

$$p_C^{(t+1)}(x_C) = \frac{Z^{(t)}}{Z^{(t+1)}} \tilde{p}(x_C)$$

- The two distributions above are normalized, thus summing both sides wrt x_C gives that the normalization factor remains constant across IPD iterations:

$$Z^{(t+1)} = Z^{(t)}$$

- Returning to the Eq. on the top of the slide, in addition we also derive that the IPD finds a distribution such that the marginal with respect to x_C is equal to the empirical marginal:

$$p_C^{(t+1)}(x_C) = \tilde{p}(x_C)$$

Properties of the IPF Update

- IPF converges to the required MLE estimates.

$$\hat{p}_{ML}(x_C) = \tilde{p}(x_C)$$

- We saw that for the MLE, the model marginals are equal to the empirical marginals, for all $C \in \mathcal{C}$. IPF works toward the goal of equal model and empirical marginals, by equating a single model marginal and empirical marginal at a time.

- It can be shown that the constancy of Z across IPF iterations implies that we can write IPF in terms of joint probabilities:

$$p^{(t+1)}(x_V) = p^{(t)}(x_V)\frac{\tilde{p}(x_C)}{p^{(t)}(x_C)}$$

- This is used often as a definition of IPF – but note that does not provide any insights on the actual implementation as

$$\psi_C^{(t+1)}(x_C) = \psi_C^{(t)}(x_C)\frac{\tilde{p}(x_C)}{p^{(t)}(x_C)}$$

Properties of the IPF Update

$$p^{(t+1)}(x_V) = p^{(t)}(x_V)\frac{\tilde{p}(x_C)}{p^{(t)}(x_C)}$$

- From this equation, we immediately obtain:

$$p^{(t+1)}(x_V) = p^{(t)}\left(x_{V\setminus C}|x_C\right)\tilde{p}(x_C)$$

- This provides an *interpretation of an IPF iteration as retaining*
  - *Retaining the old conditional probability* $p^{(t)}\left(x_{V\setminus C}|x_C\right)$
  - *Replacing the old marginal probability* $p^{(t)}(x_C)$ *with the new marginal* $\tilde{p}(x_C)$

IPF As Coordinate Ascent

- Let us derive IPF as a coordinate ascent. For fixed C and varying x_C, we take derivative of the log likelihood wrt $\psi_C(x_C)$

$$\frac{\partial \log p(\mathcal{D}|\boldsymbol{\theta})}{\partial \psi_C(x_C)} = \frac{m(x_C)}{\psi_C(x_C)} - N\frac{1}{Z}\sum_{\tilde{x}} \delta(\tilde{x}_C, x_C)\left(\prod_{D\neq C}\psi_D(\tilde{x}_D)\right)$$

- We maximize wrt $\psi_C(x_C)$ keeping all other $\psi_D(x_D), D \neq C$ constant.

$$\frac{\partial \log p(\mathcal{D}|\boldsymbol{\theta})}{\partial \psi_C(x_C)} = \frac{m(x_C)}{\psi_C(x_C)} - \frac{N}{Z}\sum_{\tilde{x}} \delta(\tilde{x}_C, x_C)\left(\prod_{D\neq C}\psi_D^{(t)}(\tilde{x}_D)\right)$$

- Recalling that the normalization constant remains constant during iterations, we can show that the IPF update Eq. sets the gradient of the log likelihood to zero:

$$\frac{\partial \log p(\mathcal{D}|\boldsymbol{\theta})}{\partial \psi_C(x_C)} = \frac{m(x_C)}{\psi_C^{(t+1)}(x_C)} - \frac{N}{Z^{(t+1)}}\sum_{\tilde{x}} \delta(\tilde{x}_C, x_C)\left(\prod_{D\neq C}\psi_D^{(t)}(\tilde{x}_D)\right)$$

$$= \frac{m(x_C)}{\psi_C^{(t+1)}(x_C)} - \frac{N}{Z^{(t)}}\sum_{\tilde{x}} \delta(\tilde{x}_C, x_C)\left(\prod_{D\neq C}\psi_D^{(t)}(\tilde{x}_D)\right) =$$

$$= \frac{m(x_C)}{\psi_C^{(t+1)}(x_C)} - \frac{N}{\psi_C^{(t)}(x_C)}\sum_{\tilde{x}} \delta(\tilde{x}_C, x_C)\frac{1}{Z^{(t)}}\left(\prod_{D}\psi_D^{(t)}(\tilde{x}_D)\right) = \frac{m(x_C)}{\psi_C^{(t+1)}(x_C)} - \frac{N}{\psi_C^{(t)}(x_C)}p^{(t)}(x_C)$$

$$\frac{\partial \log p(\mathcal{D}|\boldsymbol{\theta})}{\partial \psi_C(x_C)} = 0 \; for$$

$$\psi_C^{(t+1)}(x_C) = \psi_C^{(t)}(x_C)\frac{\tilde{p}(x_C)}{p^{(t)}(x_C)}$$

View from the KL Divergence

- We have shown that the IPF algorithm is coordinate ascent in the log likelihood. This result can also be obtained via the KL divergence.
- The KL divergence has a useful decomposition that reflects the decomposition of a joint distribution into the product of a marginal and a conditional.

$$p(x_A, x_B) = p(x_B \mid x_A)p(x_A), \text{ and } q(x_A, x_B) = q(x_B \mid x_A)q(x_A), \text{ we have:}$$

$$D\big(p(x_A, x_B) \parallel q(x_A, x_B)\big) = D\big(p(x_A) \parallel q(x_A)\big) + \sum_{x_A} p(x_A) D\big(p(x_B \mid x_A) \| q(x_B \mid x_A)\big)$$

- Recall that the problem of maximizing the likelihood is equivalent to that of minimizing the KL divergence to the empirical distribution

$$\tilde{p}(x) = \frac{1}{N}\sum_{n=1}^{N} \delta(x, x_n)$$

$$D\big(\tilde{p}(x) \,||\, p(x|\theta)\big) = \sum_x \tilde{p}(x) \log \frac{\tilde{p}(x)}{p(x|\theta)} = \sum_x \tilde{p}(x) \log \tilde{p}(x) - \sum_x \tilde{p}(x) \log p(x|\theta)$$

$$= \sum_x \tilde{p}(x) \log \tilde{p}(x) - \frac{1}{N}\sum_{n=1}^{N} \log p(x_n|\theta) = \sum_x \tilde{p}(x) \log \tilde{p}(x) - \frac{1}{N}\ell(\theta|D)$$

View from the KL Divergence

- Thus we wish to perform coordinate descent in $D\left(\tilde{p}(x) \,||\, p(x|\theta)\right)$
- That is, we pick a clique C and adjust the clique potential $\psi_C(x_C)$ so as to minimize the KL divergence. Using the earlier identity

$$D\left(\tilde{p}(x) \,||\, p(x|\theta)\right) = D\left(\tilde{p}(x_C) \,||\, p(x_C|\theta)\right) + \sum_{x_C} \tilde{p}(x_C) D\left(\tilde{p}(x_{V\setminus C}|x_C)||p(x_{V\setminus C}|x_c,\theta)\right)$$

- Changes to the clique potential $\psi_C(x_C)$ have no effect on the conditional distribution $p(x_{V\setminus C} \mid x_C, \theta)$
- Thus, the 2nd term is unaltered by changes to $\psi_C(x_C)$ and minimizing the KL divergence reduces to minimizing the 1st term. This term is minimized by setting the marginal $p(x_C|\theta)$ equal to the empirical marginal $\tilde{p}(x_C)$. This is exactly what an IPF update achieves.
- *Thus IPF is coordinate descent in $D\left(\tilde{p}(x) \,||\, p(x|\theta)\right)$ is equivalent to coordinate ascent in the log likelihood.*

Gradient Ascent

- An alternative to IPF is to perform gradient ascent on the log likelihood. In this case, given that the gradient is evaluated only at the current value of the parameters, no subtleties arise.
- Evaluating the gradient at

$$\frac{\partial \log p(\mathcal{D}|\boldsymbol{\theta})}{\partial \psi_C(x_C)} = \frac{m(x_C)}{\psi_C^{(t)}(x_C)} - \frac{N}{\psi_C^{(t)}(x_C)} p^{(t)}(x_C)$$

leads to the following gradient ascent algorithm – ρ the step size:

$$\psi_C^{(t+1)}(x_C) = \psi_C^{(t)}(x_C) + \frac{\rho}{\psi_C^{(t)}(x_C)} \left(\tilde{p}(x_C) - p^{(t)}(x_C) \right)$$

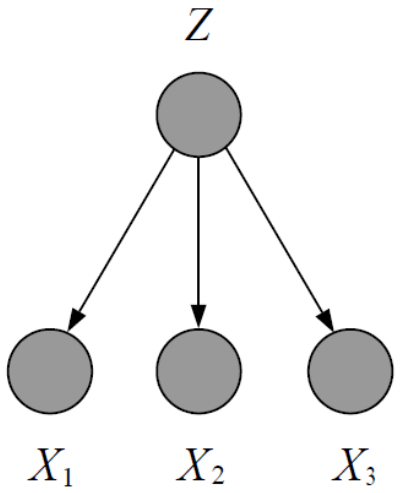
- We see that *the difference between the empirical marginals and the model marginals drives the algorithm.*
- Compared to IPF, *all of the parameters can be adjusted simultaneously.*
- Disadvantages: *need to choose a step size*, and *Z does not remain constant but must be recalculated after each iteration*.

Latent Variables

- In the case of complete data, the log likelihood is a sum where each of the parameters appear in different terms.
- This leads to a decoupling of the parameter estimation problem.
- When some of the variables are unobserved (when we have latent variables) this situation breaks down.
- In this case, the likelihood is a marginal probability, obtained by summing or integrating over the latent variables. The log likelihood is the log of this sum, and the log is prevented from moving past the sum to act on the product of potentials.
- The parameter estimation problem does not decouple.

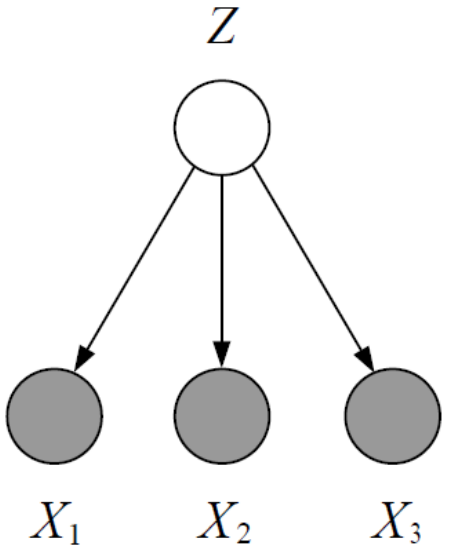
Latent Variables

- Consider the graphical model shown. Here the nodes X_i are conditionally independent given Z; thus, when Z is observed that the log likelihood decouples:



$$l(\theta; x, z) = \log p(x, z \mid \theta)$$
$$= \log p(z \mid \theta_z) + \log p(x_1 \mid z, \theta_1)$$
$$+ \log p(x_2 \mid z, \theta_2) + \log p(x_3 \mid z, \theta_3)$$

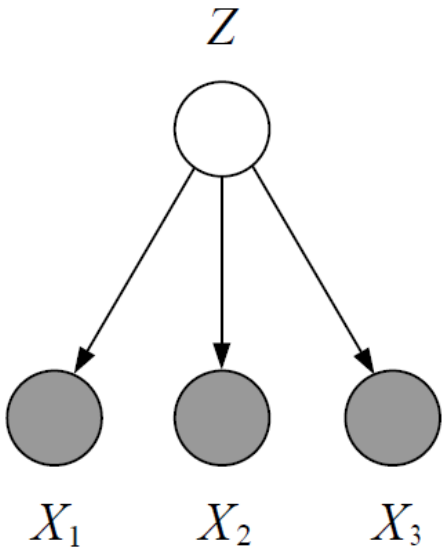
- If, on the other hand, Z is latent, the log likelihood does not decouple:



$$l(\theta; x, z) = \log \sum_z p(x, z \mid \theta) =$$
$$= \log \sum_z \left[p(z \mid \theta_z)\, p(x_1 \mid z, \theta_1)\, p(x_2 \mid z, \theta_2)\, p(x_3 \mid z, \theta_3) \right]$$

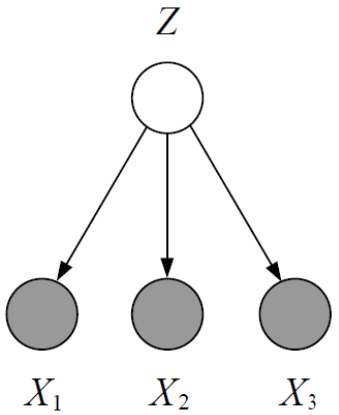
Latent Variables

- Adjusting one parameter has an effect on all of the other parameters.
  - Our uncertainty about Z is reflected in a probabilistic dependence among the variables X_i and hence among the estimates of the parameters θ_i.
- This coupling complicates the task of parameter estimation.
- The Expectation-Maximization (EM) algorithm deals with this complexity.
- *The EM algorithm in essence allows us to treat latent variable problems with complete data tools.*

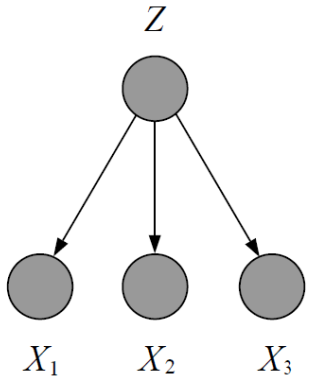


Latent Variables

- The EM algorithm allows us to solve the maximum problem for the graph



by solving a sequence of MLE problems based on the graph



- EM makes *use of convexity to move the logarithm past the sum in coupled log likelihoods*.

- This decouples the problem and allows the complete data tools discussed here to be brought into play.

Summary

- We discussed parameter estimation in completely observed graphical models (directed & undirected graphs).

- For directed graphs, the log likelihood decouples into separate terms, one for each parameter, and thus the problem of parameter estimation also decouples.

  - One collects the sufficient statistics associated with each node and its parents and
  - estimates the parameter vector at that node using those sufficient statistics.
  - This is done independently at each node in the graph.

- If the probability model at each node is a generalized linear model, then we can run the IRLS algorithm independently at each node. This is a Newton algorithm on the graph as a whole.

Summary

- For undirected graphs, we distinguish between decomposable & non-decomposable models.
- For decomposable models, we can solve for MLE analytically (heuristically write down the solution).
- There is an important relationship between decomposability and the junction tree algorithm.
- For non-decomposable models, IPF takes the form of a simple scaling algorithm in which the potentials are multiplied by a ratio of marginal probabilities.
- IPF is a fixed point algorithm and also a coordinate ascent algorithm.