CS281B/Stat241B: Advanced Topics in Learning & Decision Making

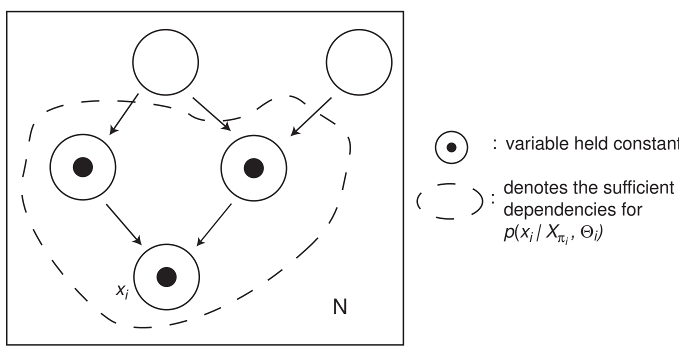
10/14/04: Observed Variables, Latent Variables, and Mixture Models

Lecturer: Michael I. Jordan *Scribes: Holly Atkinson*

1 Maximum Likelihood parameter estimation for completely observed digraph models

Given a graph $\mathcal{G}$ and choices of parameterizations θ_i of $p(x_i|x_{\pi_i}, \theta_i)$ at each node, and given i.i.d. data $\mathcal{D} = \{x_{\nu,1}, x_{\nu,2}, \ldots, x_{\nu,N}\}$, form the sufficient statistics for each θ_i (i.e., the part of $\mathcal{D}$ that determines θ_i). These will be functions on the families $\Phi_i = \{i\} \cup \pi_i$. Here (ν, n) denotes the n^{th} replicate of the entire set of observable nodes ν, and the random vector X is indexed by the nodes of the graph. π_i is the set of parent nodes of node i.

A model is *completely observed* when there is a value assigned to each random variable in the model: there are no latent variables. This allows us to decouple the ML estimate —it can then be solved separately at each node (conditioned on its parents π_i).



The local conditional probability distribution for node x_i is

$$p(x_i|x_{\pi_i}, \theta_i) = h(x_i) \exp\{(\theta_i^T x_{\pi_i})T(x_i) - A(\theta_i^T x_{\pi_i})\} \tag{1}$$

This is an application of the general conditional probability for a member of the exponential family

$$p(x|\eta) = \frac{1}{Z(\eta)} h(x) \exp\{\eta^T T(x)\} \tag{2}$$

where $A(\eta) = \log Z(\eta)$, and η is the *natural parameter* for an exponential family member. Note: in eq. (1), the term $(\theta_i^T X_{\pi_i})$ is η, and X_{π_i} refers to the parents of node X_i.

So the sufficent statistic for θ_i is

$$\left(\sum_{n=1}^{N} x_{\pi_{i,n}} T(x_{i,n}), \sum_{i=1}^{N} x_{\pi_i,n} \right) \tag{3}$$

i.e., the sufficient statistic is a pair of vectors.

Last step: compute ML estimates using a numerical optimization routine via IRLS (Iterative Reweighted Least-Squares (Newton-Raphson), a method for finding the roots of an arbitrary equation).

2 ML parameter estimation for completely observed undirected graphs

For an observed undirected graph, the overall joint probability distribution of the model is

$$p(x_\nu|\theta) = \frac{1}{Z(\psi)} \prod_{C \in \mathcal{C}} \psi_C(x_C) \tag{4}$$

where $\theta = \{\psi_C(x_C), C \in \mathcal{C}\}$ is the collection of parameters, and Z is the normalization factor.

$\tilde{p}(x)$ = the corresponding empirical distribution —e.g., a distribution which places a point mass at each data point x_n in the dataset $\mathcal{D}$.

The log likelihood (in terms of marginal counts) is then

$$\begin{aligned} l(\theta; \mathcal{D}) &= \log p(\mathcal{D}|\theta) && (5)\\ &= \sum_C \sum_{x_C} m(x_C) \log \psi_C(x_C) - N \log Z && (6) \end{aligned}$$

so

$$\begin{aligned} \frac{\partial l}{\partial \psi_C(x_C)} &= \frac{m(x_C)}{\psi_C(x_C)} - N \frac{p(x_C)}{\psi_C(x_C)} = 0 && (7)\\ \hat{p}_{ML}(x_C) &= \frac{1}{N} m(x_C) && (8)\\ \hat{p}_{ML}(x_C) &= \tilde{p}(x_C) && (9) \end{aligned}$$

Unfortunately, the normalization factor Z couples the parameters (although there is a subclass of decomposable undirected models) and this makes the parameter estimation problem more difficult. A technique called IPF is used to estimate parameters for non-decomposable undirected models.

2.1 Iterative Proportional Fitting in ML estimation on undirected models

Eq. (7), the gradient of the log likelihood, can be restated as

$$\frac{\tilde{p}(x_C)}{\psi_C(x_C)} = \frac{p(x_C)}{\psi_C(x_C)} \tag{10}$$

where the empirical marginal $\tilde{p}(x_C)$ is defined as $\frac{m(x_C)}{N}$. In IPF, cliques are processed one at a time: at iteration t,

$$\psi_C^{(t+1)}(x_C) = \psi^{(t)}(x_C)\frac{\tilde{p}(x_C)}{p^{(t)}(x_C)} \tag{11}$$

where

$$p^{(t)}(x_C) = p(x_C|\psi^{(t)}) \tag{12}$$

Note that X_C is a subset of nodes (clique indexed by C), $\tilde{p}(x_C)$ is the empirical marginal probability, and $p^{(t)}(x_C)$ is the current marginal for this clique.

- Use the elimination algorithm to get the marginal probability for this clique
- Need a publication? A dynamic programming algorithm needs to be derived for this sequence of applications of the elimination algorithm to optimize its runtime

2.1.1 Facts about IPF

- $p^{(t+1)}(x_C) = \tilde{p}(x_C)$ specifically for clique C, although subsequent iterations may alter this relationship as it is achieved for other cliques
- $Z^{(t+1)} = Z^{(t)} \Longrightarrow$ the partition function is invariant under IPF, so there's no need to recalculate it at each step
- IPF is a "coordinate ascent" method, with ascent in likelihood. At each update step, the gradient of the log likelihood is set to 0.
  - Furthermore, IPF is a "block" coordinate ascent method, as with each step you adjust a table of values rather than a single value
  - "coordinate ascent" means that it's a function of multiple variables (coordinates); movement is always parallel to the axes/coordinates in each step (like "city block" movement, and unlike gradient ascent).
- IPF is proven to converge *locally*

2.2 The KL Divergence perspective

The Kullback-Leibler (KL) divergence can be used to decompose a joint probability into the product of a marginal and a conditional. Let

$$p(x_A, x_B) = p(x_B|x_A)p(x_A) \tag{13}$$
$$q(x_A, x_B) = q(x_B|x_A)q(x_A) \tag{14}$$

then

$$D(p(x_A, x_B)\|q(x_A, x_B)) = D(p(x_A)|q(x_A)) + \sum_{x_A} D(p(x_B|x_A)\|q(x_B|x_A)) \tag{15}$$

where D is the Kullback-Leibler (KL) divergence.

In fact, ML is equivalent to

$$\min D(\tilde{p}(x)\|p(x|\psi)) = D(\tilde{p}(x_C)\|p(x_C|\psi)) + \sum_{x_C} \tilde{p}(x_C) D(\tilde{p}(x_{\nu\setminus C})\|p(x_{\nu\setminus C}|x_C, \psi)) \tag{16}$$

where $\nu = A \cup B$. (KL divergence is a calculation of cross-entropy between two distributions—in this case, between the empirical distribution and the model. It yields the log likelihood scaled by $\frac{1}{N}$.)

- Note that the second term in (16) is independent of $\psi_C(x_C)$

The adjustment achieved by IPF is that the marginal probability $p(x_C|\psi)$ converges on $\tilde{p}(x_C)$ —minimizing the KL divergence to the empirical distribution is equivalent to maximizing the likelihood of the model relative to the data.

$$\tilde{\psi}_{ML} = \operatorname*{argmax}_{\psi} \log p(x|\psi) = \operatorname*{argmin}_{\psi} D(\tilde{p}(x)\|p(x|\psi)) \tag{17}$$

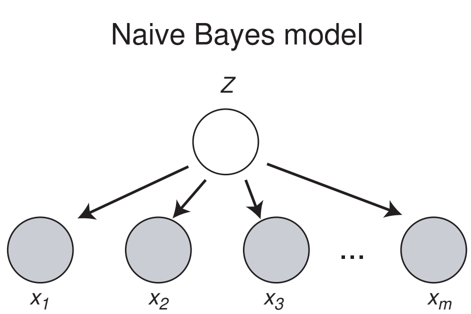
Coordinate ascent under the KL divergence formalism:

- pick C
- hold $\psi_D(x_D)$ fixed for $D \neq C$
- minimize $D(\cdot\|\cdot)$ with respect to $\psi_C(x_C)$

3 Latent variable models

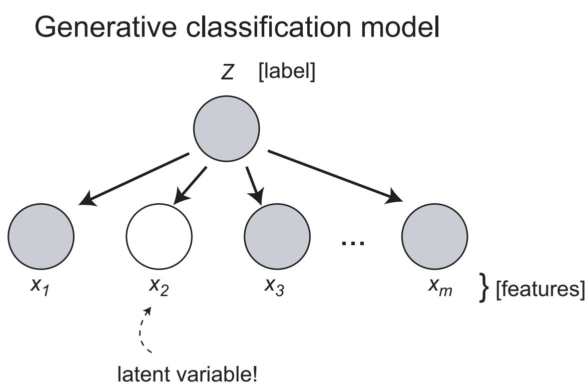
"Latent variables" ≡ "hidden variables": refers to missing data. Including this information simplifies the description of the problem, even though the data will never be seen.

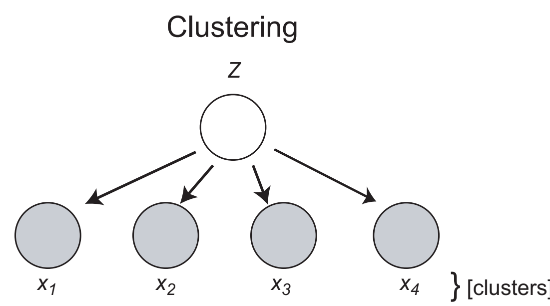
Naive Bayes model:

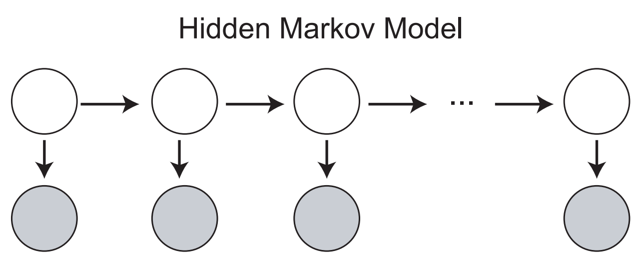


- assumes independence of Xs (leaves)
- if Z is observed, becomes a generative classification model
- if edges exist among Xs, then you get a "tree-augmented naive Bayes" model; the issue of interest then becomes deciding where to put the edges

Note: Bayesians don't distinguish between latent variables and parameters.



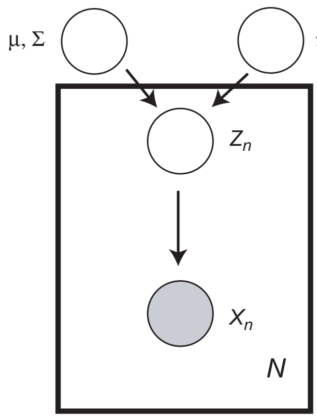


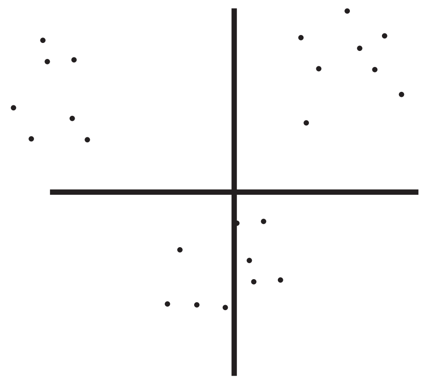


For the HMM diagrammed above, one example application is 'Part of speech tagging.' Given: text. Not given: labelling of text (e.g., *noun*, *verb*, *adj*).

4 Finite mixture models (A.K.A. probabalistic clustering)

In the case of a model involving latent variables that can take one of a finite set of values, the applicable trick is to express this using mixtures.





Given: incomplete data $\mathcal{D} = \{x_n\}_{n=1}^N$ (as opposed to complete data, $\mathcal{D}_C = \{(z_n, x_n)\}_{n=1}^N$)

$$p(x_n, z_n) = p(z_n)p(x_n|z_n) \tag{18}$$

$$p(x_n) = \sum_{z_n} p(z_n)p(x_n|z_n) \tag{19}$$

In eq (18) above, z_n is some vector ($p(z_n) = \pi$), and $p(x_n|z_n)$ is some probability distribution (e.g., Gaussian). Eq. (19) is the general expression for a finite mixture, applicable when the observable data takes the form of a mixture.

4.1 Gaussian mixtures

Here, the mixture components are Gaussian distributions with parameters θ_i is defined as (μ_i, Σ_i), Σ signifying a covariance matrix. K, the number of mixture components, is chosen in advance:

$$p(x|\theta) = \sum_{i=1}^K \pi_i \frac{1}{(2\pi)^{\frac{m}{2}} |\Sigma_i|^{\frac{1}{2}}} \exp\{-\frac{1}{2}(x - \mu_i)^T \Sigma_i^{-1} (x - \mu)\} \tag{20}$$

To calculate the probability of the latent variable Z conditioned on the observed varible X, of interest in the clustering application (the conditional probability of Z can be used to assign X to one of the clusters), we begin by rewriting the equation using r^i to denote the conditional probability that the ith component of Z is equal to one. From Bayes rule,

$$r^i \triangleq p(Z^i = 1|x, \theta) \tag{21}$$

so

$$p(z_n|\theta) = \prod_j \pi_j^{z_n^j} \tag{22}$$

$$p(x_n|z_n, \theta) = \prod_j N(\mu_j, \Sigma_j)^{z_n^i} \tag{23}$$

To begin to estimate θ, we find the log likelihood of the data:

$$l(\theta) = \sum_n \log \sum_{z_n} (\prod_j \pi_j^{z_n^j} \prod_j N(\mu_j, \Sigma_j)^{z_n^i}) \tag{24}$$

Unfortunately, taking the log does not "decouple" the problem, so we must use Jensen's inequality (related to the definition of convexity) to move the log within the inner sum:

$$l(\theta) = \sum_n \sum_{z_n} \log(\prod_j \pi_j^{z_n^j} \prod_j N(\mu_j, \Sigma_j)^{z_n^i}) \tag{25}$$

Note: the EM (Expectation Maximization) algorithm is a better way to approach this problem.