6.7810 Final Exam Cheat Sheet

Gaussian Random Variables

$\mathbf{x} \sim N(\mu, \mathbf{\Lambda})$
$E[\mathbf{x}] = \mu$
$E[(\mathbf{x}-\mu)(\mathbf{x}-\mu)^T] = \mathbf{\Lambda}$
$p_x(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}|\mathbf{\Lambda}|^{1/2}} \exp\left[-\frac{1}{2}[(\mathbf{x}-\mu)^T\mathbf{\Lambda}^{-1}[(\mathbf{x}-\mu)\right]$

Transformations

$\mathbf{z} \sim N(\mu_z, \mathbf{\Lambda}_z) \quad \mathbf{x} = \mathbf{A}\mathbf{z} + \mathbf{b} \rightarrow \mathbf{x} \sim N(\mu_x, \mathbf{\Lambda}_x)$
$\underline{\mu}_x = \mathbf{A}\mu_z + \mathbf{b} \quad \Lambda_x = \mathbf{A}\mathbf{\Lambda}_z\mathbf{A}^T$

Gaussian Information Form

$p_{\mathbf{x}}(\mathbf{x}) = \frac{1}{Z}\exp\left[-\frac{1}{2}\mathbf{x}^T\mathbf{J}\mathbf{x} + \mathbf{h}^T\mathbf{x}\right]$
$\mathbf{J} = \mathbf{\Lambda}^{-1} \quad \mathbf{h} = \mathbf{J}\mu$

Marginalization

$p(\mathbf{x}_1) \sim N(\mu_1, \mathbf{\Lambda}_{11}) = N(\mathbf{h}_1\prime, \mathbf{J}_{11}\prime)$
$\mathbf{h}_1' = \mathbf{h}_1 - \mathbf{J}_{12}\mathbf{J}_{22}'\mathbf{h}_2 \quad \mathbf{J}_{11}' = \mathbf{\Lambda}_{11}^{-1} = \mathbf{J}_{11} - \mathbf{J}_{12}\mathbf{J}_{22}^{-1}\mathbf{J}_{21}$
This is **Schur's Complement**.

Conditioning

$p_{\mathbf{x}_1|\mathbf{x}_2}(\mathbf{x}_1|\mathbf{x}_2) \sim N^{-1}(\mathbf{h}_1'', \mathbf{J}_{11}'')$
$\mathbf{J}_{11}'' = \mathbf{J}_{11} \quad \mathbf{h}_1'' = \mathbf{h}_1 - \mathbf{J}_{12}\mathbf{x}_2$
$\mu_1'' = \mu_1 + \mathbf{\Lambda}_{12}\mathbf{\Lambda}_{22}^{-1}(\mathbf{x}_2 - \mu_2) \quad \mathbf{\Lambda}_{11}'' = \mathbf{\Lambda}_{11} - \mathbf{\Lambda}_{12}\mathbf{\Lambda}_{22}^{-1}\mathbf{\Lambda}_{21}$

Characterizations

$\exists \mathbf{A} \in \mathbf{M}_{n\times n}, \mathbf{b} \in \mathbf{M}_{n\times 1}$ such that for $\mu \in \mathcal{N}(0, \mathbf{I})$, $\mathbf{x} = \mathbf{A}\mu + \mathbf{b}$
$\mathbf{x} \in \mathcal{N}(\mu, \mathbf{\Lambda})$ iff $\forall \mathbf{a} \in \mathcal{M}_{N\times 1}$, $\mathbf{a}^T\mathbf{x}$ isa Gaussian Random Variable.

Information Measures

Entropy: $H(x) = -\sum_{x\in\mathcal{X}} p(x)\log p(x)$ (also denoted $H(p)$)
Conditional Entropy:
$H(u|v) = -\sum_{u,v} p_{u,v}(u,v)p_{u|v}(u|v) = H(u) - D_{\mathrm{KL}}(p_{u,v}||p_u p_v)$.
Mutual Information: $I(u;v) = D_{\mathrm{KL}}(p_{u,v}||p_u p_v)$
Conditional Mutual Information:
$I(u;v|w) = \mathbb{E}_w[D_{\mathrm{KL}}(p_{u,v|w}||p_{u|w}p_{v|w})]$

Concentration Inequalities

Hoeffding's Inequality: for ind. $X_1, \cdots X_n$ taking values in $[a_i, b_i]$, $t > 0$, we have

$$\mathbb{P}\left[\left|\sum_{i=1}^n X_i - \mathbb{E}\left[\sum_{i=1}^n X_i\right]\right| > t\right] \leq 2e^{-2t^2/\sum_{i=1}^n (b_i-a_i)^2}$$

Pinsker's Inequality: We have

$$\mathrm{TV}(p,q) = \sup_{x\in\mathcal{X}} \|p(x) - q(x)\| \leq \sqrt{\frac{1}{2}D_{KL}(p||q)}$$

Markov's Inequality: Suppose $X \geq 0$. Then, $\forall a > 0$,

$$\mathbb{P}[X \geq a] \leq \mathbb{E}[X]/a$$

Chebyshev's Inequality: We have

$$\mathbb{P}[|X - \mathbb{E}[X]| \geq a] \leq \mathrm{var}[X]/a^2$$

for all $a > 0$.
Chernoff Bounds: For $t > 0$, we have

$$\mathbb{P}[X \geq a] \leq \mathbb{E}[e^{tx}]/e^{ta}$$

Undirected Graphical Models (L2)

Graph Seperation Property: For sets A, B, C we have $x_A \perp x_B | x_C$ if C seperates A and B.
Factorization Property (Hammersley-Clifford): For a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, if $p_{x_\mathcal{V}}$ is *strictly positive*, P satisfies the conditional independence properties by graph seperation iff $\exists$ a factorization

$$p_{x_\mathcal{V}} = \frac{1}{Z}\prod_{\mathcal{C}\in \mathrm{cl}^*\mathcal{G}} \psi_\mathcal{C}(x_\mathcal{C})$$

where $\mathrm{cl}^*(\mathcal{G})$ are the maximal cliques. Moreover, the "if" holds even w/o the strictly positive constraint. Generally such a factorization requires $O(|\mathcal{X}|^{\max|\mathcal{C}|})$ parameters.

Directed Graphical Models (L3)

D-seperation Property: A path is a **blocking path** between $a \in \mathcal{A}$ and $b \in \mathcal{B}$ wrt nodes in $\mathcal{C}$ if $\exists$ a node c on the path that satisfies either
(i). the node $c \in \mathcal{C}$ and the arrows on the path do not meet head to head
(ii). The node $c \notin \mathcal{C}$, and none of its descendants are in $\mathcal{C}$, and the arrows on the path meet head to head.
The set of nodes $\mathcal{A}$ is **d-separated** from the set of nodes $\mathcal{B}$ with respect to $\mathcal{C}$ if every path between nodes $a \in \mathcal{A}$ and $b \in B$ is blocked. We have $x_\mathcal{A} \perp x_\mathcal{B} | x_\mathcal{C}$ if and only if $\mathcal{A}$ and $\mathcal{B}$ are d-separated by $\mathcal{C}$.

Factor Graphs (R3)

Bipartite graphs between variable nodes $\mathcal{V}$ and factor nodes $\mathcal{F}$. We say G represents p if

$$p_{x_\mathcal{V}} \propto \prod_{j\in\mathcal{F}} \phi_j(x_{\mathcal{V}_j})$$

The space complexity is given by $\sum_{j\in\mathcal{F}} |\mathcal{X}|^{|\mathcal{V}_j|} \leq |\mathcal{F}||\mathcal{X}|^D$ where $D = \max_{j\in\mathcal{F}} |\mathcal{V}_j|$.

Properties of Graphical Models (L4)

	Directed	Undirected
Factorization	$p_x(x) = \prod_{i=1}^N p(x_i\|x_{\pi_i})$	$p_x(x) = \frac{1}{Z}\prod_{\mathcal{C}\in\mathrm{cl}^*(G)} \phi_\mathcal{C}(x_\mathcal{C})$
Global Markov	$x_\mathcal{A} \perp x_\mathcal{B} \| x_\mathcal{C}$ $\forall \mathcal{A}, \mathcal{B}$ d-separated wrt $\mathcal{C}$	$x_\mathcal{A} \perp x_\mathcal{B} \| x_\mathcal{C}$ $\forall \mathcal{A}, \mathcal{B}$ seperated by $\mathcal{C}$
Local Markov	$x_i \perp x_{\mathrm{nd}(i)\setminus\pi(i)} \| x_{\pi(i)}$ where nd are the non-descendants	$x_i \perp x_{\mathcal{V}\setminus(N(i)\cup\{i\})} \| x_{N(i)}$
Pairwise Markov	$x_i \perp x_j \| x_{\mathrm{nd}(i)\setminus\{j\}} \forall j \in \mathrm{nd}(i)\setminus\pi_i$	$x_i \perp x_j \| x_{\mathcal{V}\setminus\{i,j\}} \forall (i,j) \notin \mathcal{E}$

Directed

Global MP $\Leftrightarrow$ Local MP $\Leftrightarrow$ Pairwise MP

Undirected

Global MP $\implies$ Local MP $\implies$ Pairwise MP. If $p(x) > 0$ (strictly positive) we get $\Leftrightarrow$.

Dependencies (L4)

G is an **I-map** of P if $\mathcal{I}(G) \subseteq \mathcal{I}(P)$ (P has at least as many conditional independencies)
G is a **D-map** of P if $\mathcal{I}(G) \supseteq \mathcal{I}(P)$ (G has at least as many conditional independencies)
G is a **P-map** of P if $\mathcal{I}(G) = \mathcal{I}(P)$
G is a **minimal** I-map if removing any edge would make it no longer an I-map

Constructing Minimal I-Maps (L4)

Directed minimal I-map: Choose a topological ordering and factorize using Bayes' Rule:

$$p_{x_1,\cdots x_n}(x_1,\cdots,x_n) = p_{x_1}(x_1)p_{x_2|x_1}(x_2|x_1)\cdots p(x_n|x_1,\cdots,x_{n-1})$$

and remove as many conditional independencies as possible.

Undirected minimal I-map: Start w/a fully connected MRF over $\mathcal{V}$ and in arbitrary order for each edge $(i,j) \in \mathcal{E}$ remove it if it satisfies the Local Markov Property:

$$x_i \perp x_j | x_{\mathcal{V}\setminus\{i,j\}}$$

Here, we assume p is *strictly positive*. This minimal I-map is unique for p.

Conversion Between Directed/Undirected

Directed → Undirected: Moralization (add edges fully connecting parents of each node i) turns a directed to undirected. A moralization is a minimal I-map of the directed graph. If no edges are added, then the undirected graph is a P-map.

Undirected → Directed Add an orientation to all edges and then chrodalize with the elimination algorithm. Undirected graph G has a directed P-map iff G is chordal.

Fact: Chordal graphs have perfect undirected and directed graphs.

Gaussian Graphical Models (L5)

For $\mathbf{x} = \begin{bmatrix}\mathbf{x}_1\\ \mathbf{x}_2\\ \mathbf{x}_3\end{bmatrix}$, $\mu = \begin{bmatrix}\mu_1\\ \mu_2\\ \mu_3\end{bmatrix}$, $\mathbf{\Lambda} = \begin{bmatrix}\Lambda_{11} & \Lambda_{12} & \Lambda_{13}\\ \Lambda_{21} & \Lambda_{22} & \Lambda_{23}\\ \Lambda_{31} & \Lambda_{32} & \Lambda_{33}\end{bmatrix}$,

i. $\mathbf{x}_1 \perp \mathbf{x}_2$ iff $\mathbf{\Lambda}_{12} = 0$

ii. $\mathbf{x}_1 \perp \mathbf{x}_2 | \mathbf{x}_3$ iff $\mathbf{J}_{12} = 0$

Exponential Families (L6)

Given sufficient statistics $f_i : \mathcal{X} \to \mathbb{R}$ the associated exponential family is given by

$$p_x(x;\theta) = \frac{1}{Z(\theta)} \exp\left(\sum_{i=1}^{K} \theta_i f_i(x)\right)$$

if $\Theta = \{\theta \in \mathbb{R}^K\}$ is not $\emptyset$ and open, it is *regular* and if the f_i are linearly independent it is minimal.

We have the maximum entropy distribution given certain sufficient statistics is exponential:

$$\max_p H(p) = -\mathbb{E}_p[\log p] \quad \text{subject to } \mathbb{E}_p[f_i(x)] = \mathbb{E}_{\mathcal{D}}[f_i(x)]$$

implies

$$p(x) \propto \exp\left(\sum_{i=1}^{K} \lambda_i f_i(x)\right).$$

If we let $\theta = [\lambda_i]$ and $f = [f_i]$, we get $p(x) \propto \exp(\theta^T f(x))$.

Fact: We have the gradient of the log partition function of the exponential family is given by

$$\nabla_\theta \log Z(\theta) = \mathbb{E}_\theta[f(x)].$$

We also have that $\log Z(\theta)$ is convex.

Variable Elimination (L7)

We can marginalize out variables one at a time. The messages are then given by

$$m_i(x_{s_i}) = \sum_{x_i} \prod_{\varphi_i \in \Psi} \varphi_i(x_i, x_{s_i})$$

where Φ is the set of potentials that contains x_i and s_i is the corresponding set of nodes.

The **Reduced Graph** is the graph made by reducing/eliminating nodes and adding edges (as in elimination).

The **Reconstructed Graph** adds all edges during reduction back to the original graph and chordalizes the graph.

To **condition** on a variable, note $p(x_\mathcal{V}|y_\mathcal{V}) \propto p_{x_\mathcal{V},y_\mathcal{V}}(x_\mathcal{V}, y_\mathcal{V})$ so just remove the node in the graph and augment potentials.

For **directed** graphical models (Bayes nets) just apply moralization and then do belief propogation.

Sum-Product/ Belief Propogation (L8)

Sum Product (Sequential): For an undirected graphical model:

$$p(x) = \prod_i \phi_i(x_i) \prod_{(i,j)\in E} \psi_{ij}(x_i, x_j)$$

Choose arbitrary node i as root and start from all other nodes that are leaves and generate messages going towards it using elimination algorithm. The message updates/marginals are:

$$m_{i\to j}(x_j) = \sum_{x_i} \phi_i(x_i)\psi_{ij}(x_i,x_j) \prod_{k\in N(i)\setminus\{j\}} m_{k\to i}(x_i)$$

$$p_{x_i}(x_i) \propto \phi_i(x_i) \prod_{k\in N(i)} m_{k\to i}(x_i)$$

Efficient implementation has time complexity $O(N|\mathcal{X}|^2)$.

Sum Product (Parallel):

Initialize $\forall (i,j) \in \mathcal{E}$: $m^0_{i\to j}(x_j) = 1$. Iteratively apply

$$m^{t+1}_{i\to j}(x_j) = \sum_{x_i} \phi_i(x_i)\psi_{i,j}(x_i,x_j) \prod_{k\in\mathcal{N}(i)\setminus\{j\}} m^t_{k\to i}(x_i)$$

For a tree this will converge in d = diameter of $\mathcal{G}$ iterations. This has time complexity $O(dN|\mathcal{X}|^2)$.

Loopy Belief Propogation

We can still apply the parallel update rules even if the graph is not a true, but the marginals will no longer be exact. This corresponds to optimizing the Gibbs Variational Principle using the Bethe approximation of the entropy.

Max Product (R4):

Goal is now to find MAP estimate: $\mathbf{x}^* \in \arg\max_x p_\mathbf{x}(\mathbf{x})$. If there is a *unique* argmax we can simply find the max of each of the marginals and adjoin but in general this doesn't work.

What we need to do is to add companion messsages that store the argmax for each configuration: namely, repeat as before but use the max operator rather than the sum operator and store the companion messages δ that has the argmax:

$$m_{i\to j}(x_j) = \max_{x_i\in\mathcal{X}} \left[\phi_i(x_i)\psi_{ij}(x_i,x_j) \prod_{k\in N(i)\setminus\{j\}} m_{k\to i}(x_i)\right]$$

Then for an arbitrary node i compute the max-marginal:

$$p_{x_i}(x_i) \propto \phi_i(x_i) \prod_{k \in N(i)} m_{k \to i}(x_i)$$

and then backtrack:

$$\delta_{i \to j}(x_j) = \arg\max_{x_i \in \mathcal{X}} \phi_i(x_i)\psi_{i,j}(x_i, x_j) \prod_{\mathcal{N}(i) \setminus \{j\}} m_{k \to i}(x_i)$$

Forward-Backward Algorithm (L22):

Given HMM with nodes $(x_1, \ldots, x_N, y_1, \ldots, y_N)$, compute posterior marginal using BP in two passes (forward and backwards). Messages are:

$$\begin{cases} m_{i-1 \to i}(x_i) = \mathbb{P}(y_1, \cdots, y_{i-1}, x_i) & \text{(forward)} \\ m_{i+1 \to i}(x_i) = \mathbb{P}(y_{i+1}, \cdots, y_N | x_i) & \text{(backward)} \end{cases}.$$

The most common variant is the (α, β) **form**:

$$\underbrace{p(y_{i+1}|x_{i+1}) m_{i \to i+1}(x_{i+1})}_{\alpha_{i+1}(x_{i+1})} = \sum_{x_i} p(x_{i+1}|x_i) \underbrace{m_{y_i \to x_i}(x_i) m_{i-1 \to i}(x_i)}_{\alpha_i(x_i)}$$

where $\alpha_1(x_1) = p(x_1, y_1)$ and

$$\underbrace{m_{i+1 \to i}(x_i)}_{\beta_i(x_i)} = \sum_{x_{i+1}} p(x_{i+1}|x_i) p(\hat{y}_{i+1}|x_{i+1}) \underbrace{m_{i+2 \to i+1}(x_{i+1})}_{\beta_{i+1}(x_{i+1})}$$

where $\beta_N(x_N) = 1$. Messages are related by:

$$m_{i \to i+1}(x_{i+1}) = \sum_{x_i} \mathbb{P}(x_{i+1}|x_i) \alpha_i(x_i)$$

and

$$m_{i+1 \to i}(x_i) = \beta_i(x_i).$$

We then have the posterior:

$$p(x_i | y_1, \ldots, y_N) = \frac{\alpha_i(x_i)\beta_i(x_i)}{\sum_{x_i'} \alpha_i(x_i')\beta_i(x_i')}$$

Junction Trees (L9)

For graphs with loops we transform the graph into a "tree" of maximal cliques and add indicator functions that constrain the cliques to have agreeing nodes. More specifically, we constrain the clique tree to have the following:

Junction Tree Property: for any vertex $v \in \mathcal{V}$, let $\mathscr{C}_v$ denote the set of all maximal cliques in the original graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ that contain node v. If the set of nodes $\mathscr{C}_v$ is still connected upon removal of all cliques in the clique tree that don't contain v are removed, the clique tree remains connected.

Once we have a junction tree we create "supervariables" corresponding to the clique nodes and then add edge potentials between the cliques to be indicator functions that make the nodes agree. To construct a Junction tree we use the following:

Junction Tree Algorithm:
(i) Chordalize the input graph with the elimination algorithm.
(ii) Find all maximal cliques in the chordal graph.
(iii) Determine seperator sizes and build weighted clique graph
(iv) Find MST for clique graph with Kruskal (add largest weights not creating cycle first)
(v) Assign singleton and edge potentials to junction tree.
(vi) Do sum product on junction tree.
(vii). Marginalize on the junfction tree to get node marginals.
The runtime is exponential in the treewidth $O(|\text{cl}^*(\mathcal{G})||\mathcal{X}|^{tw+1})$ where $tw = \max_i |\mathcal{C}_i| - 1$ is the treewidth and $\text{cl}^*(\mathcal{G})$ is the number of maximal cliques (this is to get the clique/node marginals).

Linear Dynamical Systems (L10)

A **Linear Dynamical System (LDS)** with input $\{\mathbf{v}_i\}_{i=1}^{N-1}$ and $\mathbf{x}_1 \sim \mathcal{N}(\mu, \mathbf{\Sigma})$ is defined

$$\mathbf{x}_{i+1} = \mathbf{A}_i \mathbf{x}_i + \mathbf{D}_i \mathbf{v}_i$$

for some $\mathbf{A}_1, \cdots, \mathbf{A}_{N-1}$ and $\mathbf{D}_1, \cdots, \mathbf{D}_{N-1}$.
Such a sequence $\mathbf{x}_1, \cdots, \mathbf{x}_N$ with inputs $\boldsymbol{\epsilon}_1, \cdots, \boldsymbol{\epsilon}_{N-1}$ iid. unit Gaussians is called a **Gauss-Markov** process and can be represented by a HMM.

Gaussian Belief Propogation (L10)

Gaussian Sum Product

We can apply Sum-Product on Gaussians due to conjugate duality as:

$$m_{i \to j}(\mathbf{x}_j) \propto \mathcal{N}^{-1}(\mathbf{x}_j; \mathbf{h}_{i \to j}, \mathbf{J}_{i \to j})$$

where

$$\mathbf{h}_{i \to j} = -\mathbf{J}_{ji} \left(\mathbf{J}_{ii} + \sum_{k \in N(i) \setminus \{j\}} \mathbf{J}_{k \to i} \right)^{-1} \left(\mathbf{h}_i + \sum_{k \in N(i) \setminus \{j\}} \mathbf{h}_{k \to i} \right)$$

$$\mathbf{J}_{i \to j} = -\mathbf{J}_{ji} \left(\mathbf{J}_{ii} + \sum_{k \in N(i) \setminus \{j\}} \mathbf{J}_{k \to i} \right)^{-1} \mathbf{J}_{ij}$$

The marginals can then be computed as

$$p_{\mathbf{x}_i}(\mathbf{x}_i) = \mathcal{N}^{-1}(\mathbf{x}_i; \hat{\mathbf{h}}_i, \hat{\mathbf{J}}_i)$$

where

$$\hat{\mathbf{h}}_i = \mathbf{h}_i + \sum_{k \in N(i)} \mathbf{h}_{k \to i}, \quad \hat{\mathbf{J}}_i = \mathbf{J}_{ii} + \sum_{k \in N(i)} \mathbf{J}_{k \to i}.$$

Kalman Filtering (R5)

Given a LDS

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{v}_t, \mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}), \mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{\Lambda}_0)$$

and noisy observations:

$$\mathbf{y}_t = \mathbf{C}\mathbf{x}_t + \mathbf{w}_t, \quad \mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$$

which corresponds to a Hidden Markov Model, our goal is now to find the latent posteriors $p_{\mathbf{x}_t|\mathbf{y}_t}(\mathbf{x}_t|\mathbf{y}_t)$ for $t = 0, \cdots, T$.
We alternate between two main steps (filtering and smoothing) which correspond to the forward and backwards recursion steps in Gaussian Belief propogation. Filtering is further split into Prediction and Update Steps using the $(\alpha - \beta)$ form of the forward-backward algorithm.

Filtering

$\alpha(x_{i+1}) = \int \alpha(x_i) p(x_{i+1}|x_i) p(y_{i+1}|x_{i+1}) dx_i$

Prediction

$\mu_{i+1|i} = A\mu_{i|i}$
$\Sigma_{i+1|i} = A\Sigma_{i|i}A^T + Q$
$\mu_{0|-1} = 0$
$\Sigma_{0|-1} = \Lambda_0$

Update

$\mu_{i+1|i+1} = \mu_{i+1|i} + G_{i+1}(y_{i+1} - C\mu_{i+1|i})$
$\Sigma_{i+1|i+1} = \Sigma_{i+1|i} - G_{i+1}C\Sigma_{i+1|i}$
$G_{i+1} = \Sigma_{i+1|i}C^T(C\Sigma_{i+1|i}C^T + R)^{-1}$

Smoothing

$\gamma(x_i) = \int \gamma(x_{i+1}) \left[\frac{\alpha(x_i)p(x_{i+1}|x_i)}{\int \alpha(x_i')p(x_{i+1}|x_i')dx_i'}\right] dx_{i+1}$
$\gamma(x_i) = \frac{\alpha(x_i)\beta(x_i)}{p(y_0^t)}$
$\mu_{i|t} = \mu_{i|i} + F_i(\mu_{i+1|t} - \mu_{i+1|i})$
$\Sigma_{i|t} = F_i(\Sigma_{i+1|t} - \Sigma_{i+1|i})F_i^T + \Sigma_{i|i}$
$F_i = \Sigma_{i|i}A^T\Sigma_{i+1|i}^{-1}$

Approximate Inference (L11-14)

KL-Variational Principle (L12)

We have for a distribution

$$p(x;\theta) = \frac{\exp\left(F(x;\theta)\right)}{Z(\theta)}$$

the KL-variational principle:

$$\log Z(\theta) = \sup_q \left\{ \underbrace{\sum_x q(x)F(x;\theta)}_{\mathbb{E}_q[F(x;\theta)]} + H(q) \right\}$$

which follows from the fact that $\max_q\{D_{KL}(q||p(x;\theta))\} = 0$.
For pairwise graphical models, we have further:

Naive Mean Field Approximation (L12)

This gives a **lower bound** on the log partition function.
Now optimize over Q_{prod} where each $q_{prod} \in Q_{prod}$ factorizes as:

$$q_{\text{prod}}(x_1, \cdots, x_N) = \prod_{j=1}^N q_j(x_j)$$

where each q_j is a distribution over $\mathcal{X}_j$. Then, we get $H(q_{\text{prod}}) = \sum_{j=1}^N H(q_j)$.
For a pairwise graphical model:

$$\log Z(\theta) \geq \sup_{q\in Q_{\text{prod}}} \left\{ \sum_{j=1}^N \mathbb{E}_{q_j}[F_j(X_j;\theta)] + \sum_{(j,k)\in\mathcal{E}} \mathbb{E}_{q_j,q_k}[F_{jk}(X_j,X_k;\theta)] + \sum_{j=1}^N H(q_j) \right\}$$

where we have the constraints $\sum_{x\in\mathcal{X}_j} q_j(x_j) = 1$. This corresponds to $\rho_{jk} \to \infty$ in ρ-convexified.

Coordinate Ascent Updates

The Lagrangian coordinate ascent update is given by:

$$q_j'(x_j) = \frac{\exp(F_j(x) + \sum_{k\in\mathcal{N}(j)} \mathbb{E}_{q_k}[F_{jk}(x, X_k)]}{\sum_{y\in\mathcal{X}_j} \exp(F_j(y) + \sum_{k\in\mathcal{N}(j)} \mathbb{E}_{q_k}[F_{jk}(y, X_k)])}$$

Bethe Variational Problem (BVP) (L13)

We have the **Bethe Variational Approximation** for the entropy which is exact on trees:

$$H_{\text{bethe}}(q) = \sum_{j=1}^N H(q_j) - \sum_{(j,k)\in\mathcal{E}} I(q_{jk}, q_j, q_k)$$

where we have the constraints $\sum_{x_j\in\mathcal{X}_j} q_j(x_j) = 1$ with Lagrange multipliers λ_j and $q_j(x_j) = \sum_{x_k\in\mathcal{X}_k} q_{jk}(x_j, x_k)$ with Lagrange multipliers $\lambda_{k\to j}(x_j)$.
Provides **neither** a lower or upper bound and can be nonconcave. The bethe approximation of entropy in general also does not have to be concave leading to possibly poor local maximums. There also does not have to be a joint marginal satisfying the self-consistency constraints leading to only pseudo-marginals. This corresponds to $\rho_{jk} = 1$ in the ρ-convexified BVP.

Sum-Product as a Lagrangian Update

Doing a Lagrangian update

$$q_j(x_j) \propto \exp(F_j(x_j)) \prod_{k\in\mathcal{N}(j)} \underbrace{\lambda_{k\to j}(x_j)}_{m_{k\to j}(x_j)}$$

and the edge marginals:

$$q_{jk}(x_j, x_k) \propto e^{F_j(x_j)+F_k(x_k)+F_{jk}(x_j,x_k)} \left[\prod_{l\in\mathcal{N}(j)\setminus k} \underbrace{e^{\lambda_{l\to j}(x_j)}}_{m_{l\to j}(x_j)}\right] \left[\prod_{l\in\mathcal{N}(k)\setminus j} \underbrace{e^{\lambda_{l\to k}(x_k)}}_{m_{l\to k}(x_k)}\right]$$

by the self-consistency conditions we get the sum-product updates:

$$m_{k\to j}(x_j) \propto \sum_{x_k} e^{F_k(x_k)+F_{jk}(x_j,x_k)} \prod_{l\in\mathcal{N}(k)\setminus\{j\}} m_{l\to k}(x_k)$$

Thus, loopy BP is just the BVP on graph with cycles (which will always have fixed points).

ρ-convexified Bethe Variational Principle (L14)

Fact: We have for any spanning tree $\mathcal{T}$ of a distribution q Markov wrt a graphical model G:

$$H(q) \leq H(q_{\mathcal{T}})$$

Taking convex-combinations of the spanning trees we thus get an **upper bound** of the entropy:

$$H(q) \leq \sum_{\mathcal{T}} \rho_{\mathcal{T}} H(q_{\mathcal{T}}) = \sum_{j=1}^N H(q_j) - \sum_{\mathcal{T}} \rho_{\mathcal{T}} \left\{ \sum_{(j,k)\in\mathcal{E}(\mathcal{T})} I(q_{jk}, q_j, q_k) \right\}$$

or writing the *edge-appearance probability*

$$\rho_{jk} = \sum_{\mathcal{T}} \rho_{\mathcal{T}} \mathbb{1}[(j,k) \in \mathcal{E}(\mathcal{T})]$$

we get

$$H(q) \leq H_\rho(q) = \sum_{j=1}^N H(q_j) - \sum_{(j,k)\in\mathcal{E}} \rho_{jk} I(q_{jk}, q_j, q_k)$$

We thus get the ρ-convexified BVP:

$$\begin{aligned}\log Z(\theta) &\leq \sup_q\{G_\rho(q)\} \\ &= \sup_q \left\{ \sum_{j=1}^N \mathbb{E}_{q_j}[F_j(X_j)] + \sum_{(j,k)\in\mathcal{E}} \mathbb{E}_{q_{jk}}[F_{jk}(X_j, X_k)] \right. \\ &\quad \left. + \sum_{j=1}^N H(q_j) - \sum_{(j,k)\in\mathcal{E}} \rho_{jk} I(q_{jk}, q_j, q_k) \right\}.\end{aligned}$$

where as before we have the constraints $\sum_{x_j\in\mathcal{X}_j} q_j(x_j) = 1$ and $\sum_{x_k} q_{jk}(x_j, x_k) = q_j(x_j)$.

The message passing updates are then given by:

$$m_{k\to j}(x_j) \propto \sum_{x_k} e^{F_k(x_k)+\rho_{jk}F_{jk}(x_j,x_k)} \frac{\prod_{l\in\mathcal{N}(k)\setminus\{j\}}[m_{l\to k}(x_k)]^{\rho_{lk}}}{[m_{j\to k}(x_k)]^{1-\rho_{jk}}}$$

with single node marginals:

$$q_j(x_j) \propto \exp(F_j(x_j)) \prod_{k\in\mathcal{N}(j)} [m_{k\to j}(x_j)]^{\rho_{lk}}$$

Sampling (L15-16)

Monte Carlo Estimation

$\mu^* = \sum_{x\in\mathcal{X}} p(x)f(x) = \mathbb{E}_p[f] \approx \frac{1}{n}\sum_{i=1}^n f(x_i) = \hat{\mu}_n$ where $x_i \sim f(x)$ with $\mathbb{E}[\hat{\mu}_n] = \mu^*$ and $\text{var}[\hat{\mu}_n] = \frac{\text{var}[f]}{n}$

Importance Sampling (Accept-Reject)

Given an unnormalized distribution $\tilde{p}(x)$ and a proposal distribution q such that $q(x) > 0$ for all $p(x) > 0$ and a constant C such that $\frac{\tilde{p}(x)}{q(x)} \le C$, do the following:

i. Generate $X \sim q$ and $U \sim \text{Uniform}([0,1])$

ii. Accept $Y = X$ if $U \le \frac{\tilde{p}(x)}{q(x)}$ else reject.

This returns $Y \sim p(x)$.

Markov-Chain Monte Carlo (MCMC)

Definitions

Given a Markov Chain with transition matrix $\mathbf{P}$:

i. A distribution over the states π is **stationary** for $\mathbf{P}$ if $\sum_j \pi_j \mathbf{P}_{ji} = \pi_i$.

ii. P is **irreducible** if all states can be reached from any other state.

iii. P is **aperiodic** if there is no guaranteeed periods in the chain (if there are self-transitions with $p < 1$ then it is aperiodic)

iv. If $\mathbf{P}$ is irreducible and aperiodic then it is **ergodic** and has a unique stationary distribution π that $\mathbf{P}$ converges to.

v. $\mathbf{P}$ is **reversible** with respect to π if it satisfies the **principle of detailed balance**:

$$\pi_i \mathbf{P}_{ij} = \pi_j \mathbf{P}_{ji}$$

for all $i \neq j$. If P is ergodic and reversible wrt π, then π is the stationary distribution of $\mathbf{P}$.

Gibbs Sampling

This is precisely Metropolis-Hastings with Glauber Dynamics. Define the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = N$ with the Markov-Chain $Q = [a_\mathcal{V}, b_\mathcal{V}]$ by:

$$Q_{a_\mathcal{V}, b_\mathcal{V}} = \begin{cases} p(b_k | b_{\mathcal{V}\setminus\{k\}})/N & \text{if } a_\mathcal{V}, b_\mathcal{V} \text{ differ only in the kth element} \\ 1 - \sum_{a_\mathcal{V} \neq b_\mathcal{V}} \mathbf{P}_{a_\mathcal{V}, b_\mathcal{V}} & \text{if } a_\mathcal{V} = b_\mathcal{V} \\ 0 & \text{otherwise} \end{cases}$$

To implement this:

i. Choose $k \in \mathcal{V}$ randomly uniformly.

ii. Set $x'_{\mathcal{V}\setminus\{k\}} = x_{\mathcal{V}\setminus\{k\}}$ and generate x'_k from $p(\cdot | x_{\mathcal{V}\setminus\{k\}})$.

After a burn-in period, we reach a stationary distribution where we get $x \sim p$. For diversity of samples (to escape low-temperature areas) we typically take one of every K future samples.

Metropolis-Hastings

Start with a proposal distribution $\mathbf{Q}$ with $\mathbf{Q}_{ii} > 0$ and $G(\mathbf{Q})$ is connected (all states can be reached) and define $\mathbf{A}$ by

$$\mathbf{A}_{ij} = \min\left\{1, \frac{\tilde{p}_x(j)Q_{ji}}{\tilde{p}_x(i)Q_{ij}}\right\}$$

and define $\mathbf{P}$ by

$$\mathbf{P}_{ij} = \begin{cases} Q_{ij}A_{ij} & \text{if } j \neq i \\ 1 - \sum_{j\neq i} P_{ij} & \text{if } j = i \end{cases}$$

Then p_x will be reversible wrt $\mathbf{P}$. To implement this, more specifically do the following:

i. Generate proposed state j' according to $\mathbf{Q}$ w/state i.

ii. Flip a biased coin with acceptance probability $\mathbf{A}_{ij}$ and accept if heads.

Parameter Learning (L19)

The **Maximum Likelihood Estimate (MLE)** of data x^k is given by

$$\hat{\theta}_{ML} = \arg\max_\theta \frac{1}{K}\sum_{k=1}^K \ln p(x^k;\theta) := \arg\max_\theta \ell(\theta;\mathcal{D})$$

The log-likelihood is also given by

$$\ell(\theta;\mathcal{D}) = \sum_{a\in\mathcal{X}} \hat{p}(a)\ln p(a;\theta) = -H(\hat{p}) - D_{\text{KL}}(\hat{p}||p(\cdot;\theta))$$

Maximum Likelihood Esimation (L19)

Empirical Moment Matching: We have in general by the above, the MLE estimate is given by

$$\hat{\theta}_{ML} = \arg\max_\theta \ell(\theta;\mathcal{D}) = \arg\min_\theta D_{KL}(\hat{p}||p(;\theta))$$

Directed Graphical Model: for a directed graph $\mathcal{G}$ if the graph is "fully parameterized", ie. the parameters θ for each node conditioned on parents has it's *own* independent parameter, the MLE is just the empirical conditional eg.

$$\theta_2^{\text{MLE}} = \hat{p}_{x_2|x_1}(\cdot|\cdot)$$

Exponential Family: We have for exponential families:

$$\ell(\theta;\mathcal{D}) = \sum_{i=1}^p \theta_i \mathbb{E}_{\hat{p}}[f_i(\cdot)] - \log Z(\theta)$$

since we have the optimization is a concave optimization problem (recall $\frac{\partial}{\partial\theta_l}\log Z(\theta) = \mathbb{E}_\theta[f_l(\cdot)]$). To get the MLE estimate we can then use **projected gradient descent**.

Bayesian Estimation (L19)

Treat θ as a random variable rather than as a point estimate $\hat{\theta}_{MLE}$, treat θ as a random variable over some prior $p_\theta(\theta)$:

$$p_x(x) = \int_{-\infty} p_\theta(\theta)p_{x|\theta}(x|\theta)d\theta$$

When we observe some data, our prior now becomes a posterior:

$$p_{x|\mathcal{D}}(x|\mathcal{D}) = \int_{-\infty}^{\infty} p_{\theta|\mathcal{D}}(\theta|\mathcal{D})p_{x|\theta}(x|\theta)d\theta$$

where we use the posterior

$$p_{\theta|\mathcal{D}}(\theta|\mathcal{D}) = \frac{p_\theta(\theta)p_{\mathcal{D}|\theta}(\mathcal{D}|\theta)}{\int_{-\infty}^{\infty} p_\theta(\theta')p_{\mathcal{D}|\theta}(\mathcal{D}|\theta')d\theta'}$$

In general, due to the normalization factor this computation is intractable but for certain **conjugate priors** it becomes simple.

Dirichlet Priors:

For categorical distributions (where $p_{\mathcal{D}|\theta}(\mathcal{D}|\theta)$ follows some $\theta = [\theta_1, \cdots, \theta_L]$), we have the **Dirichlet Prior**, denoted by $p_\theta(\theta) = \mathcal{D}(\alpha_1, \cdots, \alpha_L)$ satisfies: $p_\theta(\theta) \propto \prod_{l=1}^{L} \theta_l^{\alpha_l - 1}$ we have

$$\theta|\mathcal{D} \sim \mathcal{D}(\alpha_1 + K\hat{p}(a_1), \cdots, \alpha_L + K\hat{p}(a_L))$$

and likelihood:

$$p_{x|\mathcal{D}}(a_l|\mathcal{D}) = \frac{K\hat{p}(a_l) + \alpha_l}{K + \sum_{l=1}^{L} \alpha_l}$$

MAP Estimates

Sometimes we want a point estimate given this conjugate prior, where $\hat{\theta}_{\text{MAP}} = \arg\max_\theta p_{\theta|\mathcal{D}}(\theta|\mathcal{D})$. For a Dirichlet prior:

$$\hat{\theta}_l^{\text{MAP}} = \frac{K\hat{p}(\alpha_l) + \alpha_l - 1}{K - L + \sum_{l=1}^{L} \alpha_l}$$

Learning Ising Models (L20)

Ising Model

For a pairwise Markov Ising Model,

$$\begin{cases} \theta_{ii} = \ln p_{x_\mathcal{V}}(\mathbf{e}_i) - \ln p_{x_\mathcal{V}}(\mathbf{0}) \\ \theta_{ij} = \ln p_{x_\mathcal{V}}(\mathbf{e}_{ij}) - \ln p_{x_\mathcal{V}}(\mathbf{e}_i) - \ln p_{x_\mathcal{V}}(\mathbf{e}_j) + \ln p_{x_\mathcal{V}}(\mathbf{0}) \end{cases}$$

and use $\hat{p} \xrightarrow{\text{a.s.}} p$. Also, log odds is logistic: ie. for each $i \in \mathcal{V}$,

$$\ln \frac{p_{x_i|x_{\mathcal{V}\setminus\{i\}}}(1|x_{\mathcal{V}\setminus\{i\}})}{p_{x_i|x_{\mathcal{V}\setminus\{i\}}}(0|x_{\mathcal{V}\setminus\{i\}})} = \theta_{ii} + \sum_{j \neq i} \theta_{ij} x_j$$

Logistic Regression

For logits:

$$\ln \frac{p_{y|\mathbf{z}}(1|\mathbf{z};\mathbf{w})}{p_{y|\mathbf{z}}(0|\mathbf{z};\mathbf{w})} = \sum_{l=1}^{L} w_l z_l$$

we have that the (normalized) log-likelihood:

$$\ell(\mathbf{w};\mathcal{D}) = \frac{1}{K} \sum_{k=1}^{K} y_k \sum_{l=1}^{L} w_l z_l^k + \exp\left(\sum_{l=1}^{L} w_l z_l^k\right)$$

The gradient is then

$$\frac{\partial l}{\partial w_l} = \frac{1}{K} \sum_{k=1}^{K} (z_l^k y^k - p_{y|z}(1|\mathbf{z}^k;\mathbf{w}))$$

and then use projected gradient descent.

Structure Learning (L20)

Given candidate graph structures $\mathcal{G}'$ (eg. trees on nodes $\mathcal{V}$) the maximum likelihood model structure is given by:

$$\hat{G} = \arg\max_{G'} \max_{\theta^{G'}} \ell((G', \theta^{G'}; \mathcal{D})$$

Directed Graphical Models: For directed graphs with fully-parameterized parameters (ie. each $\forall p(x_i|x_{\pi_i})$ there is an independent parameter) the score is given by

$$\hat{l}(G;\theta) = \sum_{i=1}^{N} \hat{I}(x_i; x_{\pi_i})$$

Tree Learning (L20)

Chow-Liu: For a tree-structured graphical model w/ fully parameterized parameters, then we can take the fully connected graph over the parameters with empirical mutual information on the edges:

$$\hat{I}(x_i, x_j) = \sum_{x_i, x_j} \hat{p}(x_i, x_j) \log\left(\frac{\hat{p}_{x_i,x_j}(x_i, x_j)}{\hat{p}_{x_i}(x_i)\hat{p}_{x_j}(x_j)}\right)$$

Then, we can find a MST using Kruskal which will then correspond to your maximum likelihood model structure. Chow-Liu has the property that it is *consistent* with respect to trees, namely, that if the underlying probability distribution is modeled by a tree then Chow-Liu will return the correct graphical model.

Computational Challenges of Learning (L21)

Learning a general MRF w/degree of each node $\leq d$ in time $N^{o(d)}$ is impossible if learning parity w/noise in time $N^{o(d)}$ is impossible. The parity problem is given by:

Learning Parity w/Noise: Let $X \sim \text{Uniform}\{-1, 1\}^N$ and $f(x) = \prod_{i \in S} x_i$ where $S \subset [N]$ such that $|S| \leq d$. Let the noise $Z = 1$ with probability $1 - \epsilon$, and $Z = -1$ with probability ϵ. We observe i.i.d. noisy observations $(X^k, f(X^k) \cdot Z^k)$, $k = 1, \cdots, K$. The goal is to find S.

Learning Ising Models (L21)

We can utilize logistic regression with projected gradient descent on the M**-estimator**:

$$\mathcal{L}_K(\theta) = \mathcal{L}(\theta;\mathcal{D}) = \frac{1}{K} \sum_{k=1}^{K} \exp\left(-\sum_{i<j} \theta_{ij}(x_i x_j - \frac{1}{4}) - \sum_i \theta_{ii}(x_i - \frac{1}{2})\right)$$

For an ising model with each node having degree $\leq d$ we get the **sparsity constraint** $\|\theta\|_1 \leq \sqrt{d}\,\|\theta\|_2, \forall \theta \in \Theta$ where we assume $\theta \in [-B, B]$ is bounded. Then, as $K \to \infty$, $\mathcal{L}(\theta;\mathcal{D}) \to \mathcal{L}(\theta)$. This is computationally efficient since it doesn't require us to compute the partition function $Z(\theta)$ and can be computed efficiently through the Lasso optimization:

$$\hat{\theta}_K^{sc} \in \arg\min_{\theta \in [-B,B]^{N \times N}} \mathcal{L}_K(\theta) + \lambda_L \sum_i \|\theta_i\|_1$$

Learning Models with Latent Variables (L22)

In contrast to the supervised learning setting, now we do unsupervised learning where we see only some observed variables $\mathbf{y} = (y^{(1)}, \cdots, y^{(K)})$ so the log-likelihood is now given by:

$$\ell(\boldsymbol{\theta}) = \sum_{k=1}^{K} \ln p_y(y^{(k)}) = \sum_{k=1}^{K} \ln \sum_{x \in \mathcal{X}} p_{x,y}(x, y^{(k)}; \boldsymbol{\theta})$$

Expectation Maximization

Now assume we have a set of observed variables $\mathbf{y}$ and latents $\mathbf{x}$. Now introduce an arbitrary posterior over the alphabet $\mathcal{X}$ that does not depend on the parameters to get the **Evidence Lower Bound (ELBO)**:

$$\ell(\boldsymbol{\theta}) = \ln p_\mathbf{y}(\mathbf{y};\boldsymbol{\theta}) \geq \mathbb{E}_{q(\cdot|y)}\left[\ln \frac{p_{\mathbf{x},\mathbf{y}}(\mathbf{x},\mathbf{y};\boldsymbol{\theta})}{q(\mathbf{x}|\mathbf{y})}\right] = \ell_{q(\cdot|y)}(\boldsymbol{\theta})$$

EM Algorithm: Iteratively optimize the ELBO by optimizing over q and $\boldsymbol{\theta}$:

(E-step): $q^{(i+1)} = \arg\max_q \ell(q, \theta^{(i)})$

(M-step): $\boldsymbol{\theta}^{(i+1)} \in \arg\max_{\boldsymbol{\theta}} \ell(q^{(i+1)}, \boldsymbol{\theta})$

Solving those steps gives:

(E-step): $q^{(i+1)} = p_{x|y}(\cdot|y; \boldsymbol{\theta}^(i))$

(M-step):

$\boldsymbol{\theta}^{(i+1)} \in \arg\max_{\theta} \mathbb{E}_{q(\cdot|\mathbf{y})} [\ln p_{\mathbf{x},\mathbf{y}}(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta})] = \arg\max_{\boldsymbol{\theta}} \hat{\ell}_c^q(\theta).$

Thus, combining we get

$$\hat{\boldsymbol{\theta}}^{(l+1)} \in \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{p_{\mathbf{x}|\mathbf{y}}(\cdot|\mathbf{y};\hat{\boldsymbol{\theta}}^{(l)})} \ln p(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta})$$

Learning Hidden Markov Models

For a homogenous hidden Markov model with parameters $\boldsymbol{\theta} = (\boldsymbol{\pi}, \mathbf{A}, \boldsymbol{\eta})$ given by

$$[\boldsymbol{\pi}]_i = \pi_i = p_{x_0}(i), \quad [\mathbf{A}]_{ij} = a_{ij} = p_{x_{t+1}|x_t}(j|i), \quad [\boldsymbol{\eta}]_{ij} = \eta_{ij} = p_{y_t|x_t}(j|i)$$

We then get that the the complete log likelihood is given by:

$$\begin{aligned}
&\ln p_{\mathbf{x},\mathbf{y}}(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}) \\
&= \ln \pi_{x_0} \prod_{t=0}^{T-1} a_{x_t,x_{t+1}} \prod_{t=0}^{T} \eta_{x_t,y_t} \\
&= \ln \prod_{i=1}^{M} \pi_{x_0}^{\mathbb{1}_{x_0=i}} \prod_{t=0}^{T-1} \prod_{i=1}^{M} \prod_{j=1}^{M} a_{ij}^{\mathbb{1}_{x_t=i,\mathbb{1}_{x_{t+1}=j}}} \prod_{t=0}^{T} \prod_{i=1}^{M} \prod_{j=1}^{M} \eta_{ij}^{\mathbb{1}_{x_t=i,\mathbb{1}_{y_t=j}}} \\
&= \sum_{i=1}^{M} x_0^i \ln \pi_i + \sum_{t=0}^{T-1} \sum_{i=1}^{M} \sum_{j=1}^{M} x_t^i x_{t+1}^j \ln a_{ij} + \sum_{t=0}^{T} \sum_{i=1}^{M} \sum_{j=1}^{M} x_t^i y_t^j \eta_{ij} \\
&= \sum_{i=1}^{M} \underbrace{x_0^i}_{\zeta_i} \ln \pi_i + \sum_{i=1}^{M} \sum_{j=1}^{M} \underbrace{\sum_{t=0}^{T-1} x_t^i x_{t+1}^j}_{m_{ij}} \ln a_{ij} + \sum_{i=1}^{M} \sum_{j=1}^{M} \underbrace{\sum_{t=0}^{T} x_t^i y_t^j}_{n_{ij}} \eta_{ij}
\end{aligned}$$

With Lagrange multipliers, we get the stationary points:

$$\hat{\pi}_i = \zeta_i, \quad \hat{a}_{ij} = \frac{m_{ij}}{\sum_{k=1}^{K} m_{ik}}, \quad \hat{n}_{ij} = \frac{n_{ij}}{\sum_{k=1}^{K} n_{ik}}$$

Baum-Welch Algorithm

Applying EM to the homogenous HMM case, we get the ELBO is given by:

$$\begin{aligned}
&\mathbb{E}_{p_{\mathbf{x}|\mathbf{y}}(\cdot|\mathbf{y};\hat{\boldsymbol{\theta}}^{(l)})} \ln p(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}) \\
&= \sum_{i=1}^{M} \underbrace{\mathbb{E}_{p_{x_0|\mathbf{y}}(\cdot|\mathbf{y};\hat{\boldsymbol{\theta}}^{l})} x_0^i \ln \pi_i}_{\zeta_i} \\
&+ \sum_{i=1}^{M} \sum_{j=1}^{M} \underbrace{\sum_{t=0}^{T-1} \mathbb{E}_{p_{x_t,x_{t+1}|\mathbf{y}}(\cdot|\mathbf{y};\hat{\boldsymbol{\theta}}^{(l)})} x_t^i x_{t+1}^j}_{m_{ij}} \ln a_{ij} \\
&+ \sum_{i=1}^{M} \sum_{j=1}^{M} \underbrace{\sum_{t=0}^{T} \mathbb{E}_{p_{x_t|\mathbf{y}}(\cdot|\mathbf{y};\hat{\theta}^{(l)})} x_t^i y_t^j}_{n_{ij}} \eta_{ij}
\end{aligned}$$

To compute the marginals:

$$\mathbb{E}_{p_{x_t|y}(\cdot|\mathbf{y};\hat{\boldsymbol{\theta}}^{(l)})} x_t^i = p_{x_t|\mathbf{y}}(i|\mathbf{y}; \hat{\boldsymbol{\theta}}^{(l)})$$

$$\mathbb{E}_{p_{x_t,x_{t+1}|y}(\cdot,\cdot|\mathbf{y};\hat{\boldsymbol{\theta}}^{(l)})} x_t^i x_{t+1}^j = p_{x_t,x_{t+1}|\mathbf{y}}(i, j|\mathbf{y}; \hat{\boldsymbol{\theta}}^{(l)})$$

we use the $(\alpha - \beta)$ form of forward-backward:

$$\begin{aligned}
p_{\boldsymbol{x}_t,\boldsymbol{x}_{t+1}|\mathbf{y}}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}|\mathbf{y}) &\propto m_{t-1\to t}(\boldsymbol{x}_t) \underbrace{\phi_t(\boldsymbol{x}_t)}_{p(y_t|\boldsymbol{x}_t)} \underbrace{\psi_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1})}_{p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t)} \underbrace{\phi_{t+1}(\boldsymbol{x}_{t+1})}_{p(y_{t+1}|\boldsymbol{x}_{t+1})} m_{t+2\to t+1}(\boldsymbol{x}_{t+1}) \\
&= m_{t-1\to t}(\boldsymbol{x}_t) p(y_t|\boldsymbol{x}_t) p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t) p(y_{t+1}|\boldsymbol{x}_{t+1}) m_{t+2\to t+1}(\boldsymbol{x}_{t+1}) \\
&= \sum_{\boldsymbol{x}_{t-1}} \alpha_{t-1}(\boldsymbol{x}_{t-1}) p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) p(y_t|\boldsymbol{x}_t) p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t) p(y_{t+1}|\boldsymbol{x}_{t+1}) \beta_{t+1}(\boldsymbol{x}_{t+1}) \\
&= \alpha_t(\boldsymbol{x}_t) p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t) p(y_{t+1}|\boldsymbol{x}_{t+1}) \beta_{t+1}(\boldsymbol{x}_{t+1}).
\end{aligned}$$

Thus, edge marginals at iteration l take the form:

$$p_{\boldsymbol{x}_t,\boldsymbol{x}_{t+1}|\mathbf{y}}(i, j|\mathbf{y}; \hat{\boldsymbol{\theta}}^{(l)}) \propto \alpha_t(i) \hat{a}_{ij}^{(l)} \hat{\eta}_{jy_{t+1}}^{(l)} \beta_{t+1}(j)$$

We thus get the EM-updates:

$$\hat{\pi}_i^{(l+1)} = \zeta_i = \mathbb{E}_{p_{\boldsymbol{x}_0|\mathbf{y}}(\cdot|\mathbf{y};\hat{\boldsymbol{\theta}}^{(l)})} \boldsymbol{x}_0^i = p_{\boldsymbol{x}_0|\mathbf{y}}(i|\mathbf{y}; \hat{\boldsymbol{\theta}}^{(l)})$$

$$\hat{a}_{ij}^{(l+1)} = \frac{m_{ij}}{\sum_{k=1}^{M} m_{ik}} = \frac{\sum_{t=0}^{T-1} p_{\boldsymbol{x}_t,\boldsymbol{x}_{t+1}|\mathbf{y}}(i, j|\mathbf{y}; \hat{\boldsymbol{\theta}}^{(l)})}{\sum_{t=0}^{T-1} p_{\boldsymbol{x}_t|\mathbf{y}}(i|\mathbf{y}; \hat{\boldsymbol{\theta}}^{(l)})}$$

$$\hat{\eta}_{ij}^{(l+1)} = \frac{n_{ij}}{\sum_{k=1}^{M} n_{ik}} = \frac{\sum_{t=0}^{T} p_{\boldsymbol{x}_t,y_t|\mathbf{y}}(i, j|\mathbf{y}; \hat{\boldsymbol{\theta}}^{(l)})}{\sum_{t=0}^{T} p_{\boldsymbol{x}_t|\mathbf{y}}(i|\mathbf{y}; \hat{\boldsymbol{\theta}}^{(l)})}$$

Repeating these steps (inference using forward-backwards and optimization of the parameters using the EM updates) we get the **Baum-Welsch algorithm** for finding the MLE of an HMM.

Miscallaneous

Jensen's Inequality

We have for a convex function $f : I \to \mathbb{R}$ (ie. $\forall t \in [0, 1]$, $\forall x_1, x_2 \in I$ $f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2)$ or if f is twice differentiable, $f''(x) \geq 0$ for all $x \in I$) we have that

$$f\left(\sum_i a_i x_i\right) \leq \sum_i a_i f(x_i)$$

In terms of expectation, this gives

$$f(\mathbb{E}[x]) \leq \mathbb{E}[f(x)]$$

For concave functions we get th opposite direction.

Sampling from Tree Graphical Models

We can use sum product to get the conditionals $p_{x_j|x_1,\cdots,x_{j-1}}$ by noting

$$p(x_1, \cdots, x_j) \propto \prod_{i=1}^{j} \phi_j(x_j) \prod_{\substack{(i,k)\in\mathcal{E} \\ i,k\leq j}} \psi_{i,k}(x_i, x_k) \prod_{(v,k)\in\mathcal{E}^+} m_{v\to k}(x_k)$$

where we have $\mathcal{E}^+$ is the set of edges that go from a node not in $\{x_1, \cdots, x_j\}$ to one of the $\{x_1, \cdots, x_j\}$. Note by Bayes' Rule we have

$$p_{\mathbf{x}}(\mathbf{x}) = p_{x_0}(x_0) p_{x_1|x_0}(x_1|x_0) \cdots p(x_n|x_1, \cdots, x_{n-1})$$

Thus, we can sample first from $x_0 \sim p_{x_0}(x_0)$ then sample from all of the conditionals $x_1 \sim p_{x_1|x_0}(x_1|x_0)$ and so on.

Higher Order Markov Chains

We have a Markov chain of order n (ie. each node is connected to the previous n nodes only) has junction tree that is just given by Markov chain with clique size $n + 1$ that slides across the chain.