Stat 521A
Lecture 18

Outline

- Cts and discrete variables (14.1)
- Gaussian networks (14.2)
- Conditional Gaussian networks (14.3)
- Non-linear Gaussian networks (14.4)
- Sampling (14.5)

Hybrid networks

- A “hybrid” GM contains discrete and cts variables
- Except in the case that everything is all discrete or all Gaussian, exact inference is rarely possible
- The reason is that the basic operations of multiplication, marginalization and conditioning are not closed except for tables and MVNs

Gaussian networks

- We can always convert a Gaussian DGM or UGM to an MVN and do exact inference in $O(d^2)$ space and $O(d^3)$ time
- However, d can be large (eg 1000x1000 image)
- We seek methods that exploit the graph structure, that will take $O(d\ w^2)$ space and $O(d\ w^3)$ time, where w is the tree width
- In cases where w is too large, we can use loopy belief propagation, which takes O(1) space and O(d) time

Canonical potentials

- When performing VarElim or ClqTree propagation, we have to represent factors \phi(x). These may not be Gaussians, but can always be represented as exponentials of quadratics

$X1 \rightarrow X2 \rightarrow X3$

$p(x1, x2) \quad p(x3 \mid x2) = N(x3 \mid W_3 x2, \Sigma_3)$

$$\mathcal{C}(X; K, h, g) = \exp\left(-\frac{1}{2}X^T K X + h^T X + g\right).$$

Thus, $\mathcal{N}(\mu; \Sigma) = \mathcal{C}(K, h, g)$ where:

$$\begin{aligned} K &= \Sigma^{-1} \\ h &= \Sigma^{-1}\mu \\ g &= -\frac{1}{2}\mu^T \Sigma^{-1}\mu - \log\left((2\pi)^{n/2}|\Sigma|^{1/2}\right). \end{aligned}$$

Operations on canonical potentials

- Multiplication

$$\mathcal{C}(K_1, h_1, g_1) \cdot \mathcal{C}(K_2, h_2, g_2) = \mathcal{C}(K_1 + K_2, h_1 + h_2, g_1 + g_2)$$

$$\phi_1(X,Y) = \mathcal{C}\left(X, Y; \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}, -3\right) \quad * \quad \phi_2(Y,Z) = \mathcal{C}\left(Y, Z; \begin{bmatrix} 3 & -2 \\ -2 & 4 \end{bmatrix}, \begin{pmatrix} 5 \\ -1 \end{pmatrix}, 1\right).$$

$$= \quad \mathcal{C}\left(X, Y, Z; \begin{bmatrix} 1 & -1 & 0 \\ -1 & 4 & -2 \\ 0 & -2 & 4 \end{bmatrix}, \begin{pmatrix} 1 \\ 4 \\ -1 \end{pmatrix}, -2\right)$$

- Division

$$\frac{\mathcal{C}(K_1, h_1, g_1)}{\mathcal{C}(K_2, h_2, g_2)} = \mathcal{C}(K_1 - K_2, h_1 - h_2, g_1 - g_2)$$

Operations on canonical potentials

Marginalization (requires KYY be pd)

$$\int \mathcal{C}(X, Y; K, h, g)\, dY.$$

$$\begin{aligned} K' &= K_{\mathbf{XX}} - K_{\mathbf{XY}} K_{\mathbf{YY}}^{-1} K_{\mathbf{YX}} \\ h' &= h_{\mathbf{X}} - K_{\mathbf{XY}} K_{\mathbf{YY}}^{-1} h_{\mathbf{Y}} \\ g' &= g + \tfrac{1}{2}\left(|Y| \log(2\pi) - \log |K_{\mathbf{YY}}| + h_{\mathbf{Y}}^T K_{\mathbf{YY}} h_{\mathbf{Y}} \right). \end{aligned}$$

Conditioning (Y=y)

$$\begin{aligned} K' &= K_{\mathbf{XX}} \\ h' &= h_{\mathbf{X}} - K_{\mathbf{XY}}\, y \\ g' &= g + h_{\mathbf{Y}}^T y - \frac{1}{2} y^T K_{\mathbf{YY}}\, y. \end{aligned}$$

Kalman filter- smoother

- If you apply the FB algorithm with these new operators, you get the same results as the RTS smoother

$$
\begin{tikzcd}
x_1 \arrow[r] \arrow[d] & x_2 \arrow[r] \arrow[d] & x_3 \arrow[d] \\
y_1 & y_2 & y_3
\end{tikzcd}
$$

Gaussian LBP

- If the treewidth is too large, we can pass messages on the original (pairwise) graph
- We just apply the regular BP rules with the new operators. Once can show this is equivalent to the following:

$$p(X_1, \ldots, X_n) \propto \left(-\frac{1}{2}X^T J X + h^T X\right).$$

$$\delta_{i \to j}(x_j) = \exp\left(-\frac{1}{2}J_{i\to j}x_j^2 + h_{i\to j}x_j\right).$$

$$\begin{aligned} \hat{J}_{i\setminus j} &= J_{ii} + \textstyle\sum_{k \in \mathrm{Nb}_i - \{j\}} J_{k \to i} \\ \hat{h}_{i\setminus j} &= h_i + \textstyle\sum_{k \in \mathrm{Nb}_i - \{j\}} h_{k \to i}. \end{aligned}$$

$$\begin{aligned} J_{i\to j} &= -J_{ji}\hat{J}_{i\setminus j}^{-1} J_{ji} \\ h_{i\to j} &= -J_{ji}\hat{J}_{i\setminus j}^{-1}\hat{h}_{i\setminus j}. \end{aligned}$$

$$\begin{aligned} \hat{J}_i &= J_{ii} + \sum_{k \in \mathrm{Nb}_i} J_{k \to i} \\ \hat{h}_i &= h_i + \sum_{k \in \mathrm{Nb}_i} h_{k \to i}. \end{aligned}$$

$$\begin{aligned} \hat{\mu}_i &= (\hat{J}_i)^{-1}\hat{h}_i \\ \hat{\sigma}_i^2 &= (\hat{J}_i)^{-1} \end{aligned}$$

Gaussian LBP

- Thm 14.2.4. If LBP converges, then the means are exact, but the variances are too small (overconfident)
- Thm. A sufficient condition for convergence is that the potentials are pairwise normalizable
- Any attractive model (all +ve correlations) is pairwise normalizable
- The method for computing the means is similar to solving a set of linear equations

Pairwise normalizable

- Def 7.3.3. A pairwise MRF with energies of the form

$$\begin{aligned}\epsilon_i(x_i) &= d_0^i + d_1^i x_1 + d_2^i x_i^2 \\ \epsilon_{ij}(x_i, x_j) &= a_{00}^{i,j} + a_{01}^{i,j} x_i + a_{10}^{ij} x_j + a_{11}^{ij} x_i x_j + a_{02}^{ij} x_i^2 + a_{20}^{ij} x_j^2\end{aligned}$$

  is called pairwise normalizable if

$d_2^i > 0, \forall i$ and $\begin{pmatrix} a_{02}^{ij} & a_{11}^{ij}/2 \\ a_{11}^{ij}/2 & a_{20}^{ij} \end{pmatrix}$ is psd for all i,j

- Thm 7.3.4. If the MRF is pairwise normalizable, then it defines a valid Gaussian.
- Sufficient but not necessary eg.

$$\begin{pmatrix} 1 & 0.6 & 0.6 \\ 0.6 & 1 & 0.6 \\ 0.6 & 0.6 & 1 \end{pmatrix}$$

May be able to reparameterize the node/edge potentials to ensure pairwise normalized.

Conditional linear Gaussian networks

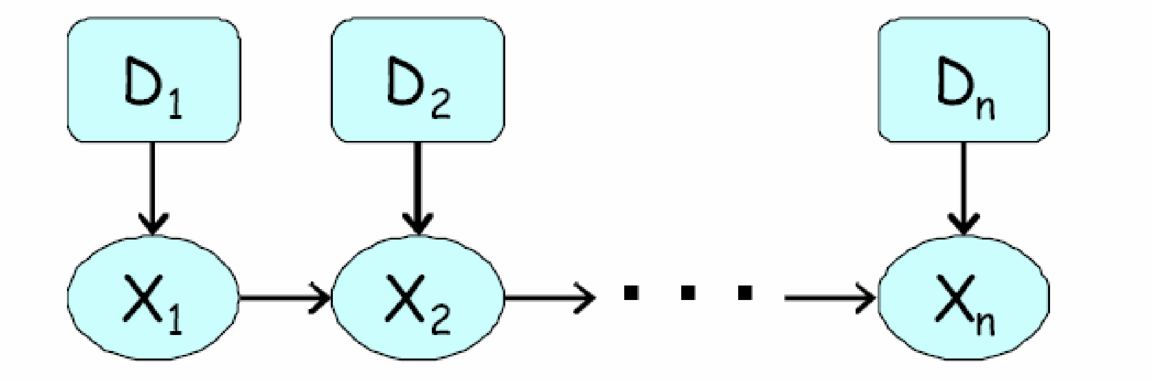
- Suppose all discrete nodes only have discrete parents, and all cts nodes either have discrete parents, cts parents, or no parents.
- Further, assume all cts CPDs have the form

$$p(X = x | C = \mathbf{c}, D = k) = \mathcal{N}(x | \mathbf{w}_k^T \mathbf{c}, \sigma_k^2)$$

- This is called a CLG network. It is equivalent to a mixture of MVNs, where the distribution over discrete indicators has structure, as does each covariance matrix.
- We create a canonical factor for each discrete setting of the variables in a clique.

Inference in CLG networks

- Thm 14.3.1. Inference in CLG networks is NP-hard, even if they are polytrees.
- Pf (sketch). Consider the network below. When we sum out D_1, $p(X_1)$ is a mixture of 2 Gaussians. In general, $p(X_i)$ is a mixture of 2^i Gaussians.



$$p(X_2) = \sum_{D_2} P(D_2) \int_{X_1} p(X_2 \mid X_1, D_2) \sum_{D_1} p(X_1 \mid D_1).$$

Weak marginalization

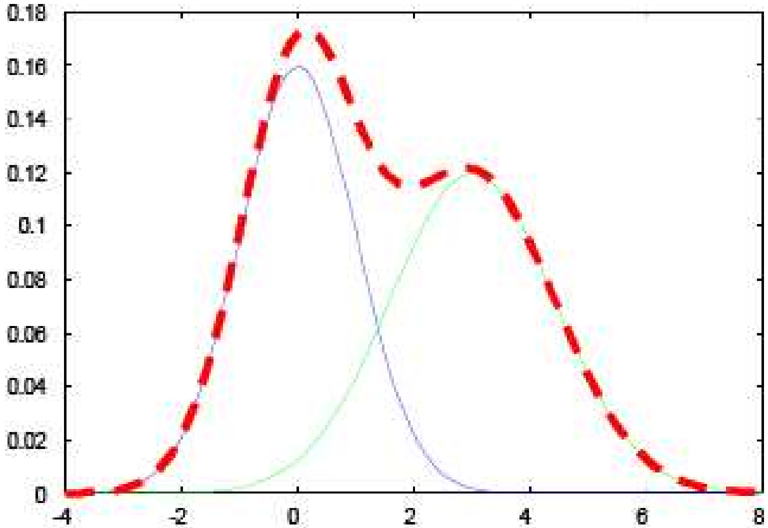
- To prevent the blowup in the number of mixture components, we can project back to the class of single mixtures at each step, as in EP
- Prop 14.3.6. argmin_q KL(p|q) where q is a Gaussian has parameters (

$$\begin{aligned} \mu_i &= E_p[X_i] \\ \Sigma_{i,j} &= Cov_p[X_i; X_j] \end{aligned}$$

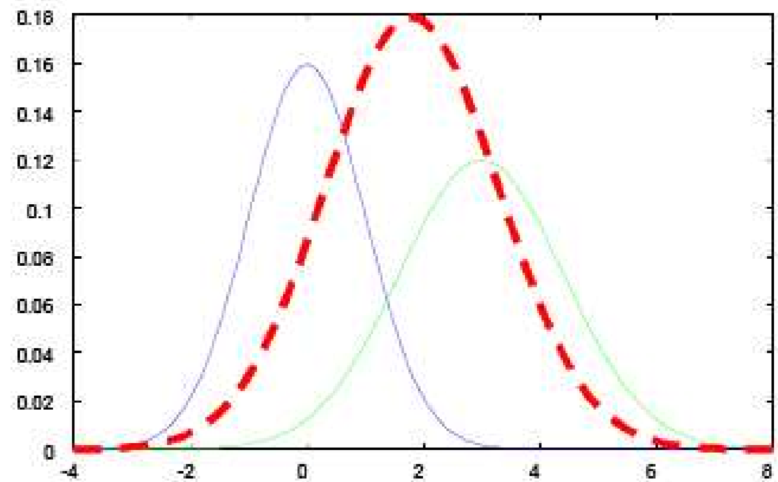
- Prop 14.3.7. argmin_q KL(p,q) where p is a mixture of Gaussians is a single Gaussian with params

$$\begin{aligned} \mu &= \sum_{i=1}^{k} w_i \mu_i \\ \Sigma &= \sum_{i=1}^{k} w_i \Sigma_i + \sum_{i=1}^{k} w_i (\mu_i - \mu)(\mu_i - \mu)^T. \end{aligned}$$

Weak marginalization



(a)



(b)

Canonical vs moment form

- Weak marginalization is defined in terms of moment form
- To convert a canonical factor to moment form, we require that it represent a valid joint density
- This typically requires we pass messages from parents to children.
- Once we have initialized all factors, they can be converted to moment form.
- However, division in the backwards pass may cause some variances to become negative! (see Ex 14.3.13)
- EP is hairy!

Strong marginalization

- By using a constrained elimination order, in which we integrate out before summing out, we can ensure that the upwards pass never needs to perform weak marginalization.
- Furthermore, one can show that the downwards pass results in exact results for the discrete variables and exact 1st and 2nd moments for the cts variables (Lauritzen's "strong jtree" algorithm)
- However, the constrained elim order usually results in large discrete cliques, making this often impractical.

Non linear dependencies

- In a linear Gaussian network, the mean is a linear function of its parents.
- Now assume $X_i = f(U_i, Z_i)$, where $Z_i \sim N(0,I)$

auxiliary variables into the variables of interest. For a vector of functions $\vec{f} = (f_1, \ldots, f_d)$ and a Gaussian distribution p_0, we use the notation $p(X_1, \ldots, X_d) = (p_0 \bigoplus \vec{f})$ to refer to the distribution that has $p(f_1(Z), \ldots, f_d(Z)) = p_0(Z)$ and 0 elsewhere.

- Examples

Example 14.4.1: . *For example, consider a multi-variate Gaussian* $p(X_1, \ldots, X_d) = \mathcal{N}(\boldsymbol{X}; \boldsymbol{\mu}, \Sigma)$. *We define a matrix A to be a $d \times d$ matrix such that $AA^T = \Sigma$; A is often called the* square root *of Σ, and is guaranteed to exist whenever Σ is positive definite. In this case we can show (see Exercise 14.6) that we can redefine p as:*

$$p(\boldsymbol{X}) = p_0(\boldsymbol{W}) \bigoplus (A\boldsymbol{W} + \mu), \tag{14.14}$$

where $p_0(\boldsymbol{W}) = \mathcal{N}(\boldsymbol{W}; \boldsymbol{0}, I)$, for I the identity matrix. ■

Example 14.4.2: *As another example, consider the non-linear CPD $X \sim \mathcal{N}\left(\sqrt{Y_1^2 + Y_2^2}; \sigma^2\right)$. We can reformulate this CPD in terms of a deterministic, non-linear function, as follows: We introduce a new exogenous variable W that captures the stochasticity in the CPD. We then define $X = f(Y_1, Y_2, W)$ where $f(Y_1, Y_2, W) = \sqrt{Y_1^2 + Y_2^2} + \sigma W$.* ■

Taylor series approx

- We can linearize f and then fit a Gaussian (basis of the EKF algorithm)

As we know, if $p_0(\mathbf{Z})$ is a Gaussian distribution and $X = f(\mathbf{Z})$ is a linear function, then $p(X) = p(f(\mathbf{Z}))$ is also a Gaussian distribution. Thus, one very simple and commonly used approach is to approximate f as a linear function $\hat{f}$, and then define $\hat{p}$ in terms of $\hat{f}$.

The most standard linear approximation for $f(\mathbf{Z})$ is the Taylor series expansion around the mean of $p_0(\mathbf{Z})$:

$$\hat{f}(\mathbf{Z}) = f(\boldsymbol{\mu}) + \nabla f|_{\boldsymbol{\mu}} \mathbf{Z}. \tag{14.15}$$

Can be bad if f not linear near mu

Although the Taylor series expansion provides us with the optimal linear approximation to f, the Gaussian $\hat{p}(X) = p_0(\mathbf{Z}) \bigoplus \hat{f}(\mathbf{Z})$ may not be the optimal Gaussian approximation to $p(X) = p_0(\mathbf{Z}) \bigoplus f(\mathbf{Z})$.

Example 14.4.4: *Consider the function $X = Z^2$, and assume that $p(Z) = \mathcal{N}(Z; 0, 1)$. The mean of X is simply $\mathbb{E}_p[X] = \mathbf{E}_p[Z^2] = 1$. The variance of X is*

$$\mathrm{Var}_p[X] = \mathbf{E}_p\left[Z^2\right] - \mathbf{E}_p[Z]^2 = \mathbf{E}_p\left[Z^4\right] - \mathbf{E}_p\left[Z^2\right]^2 = 3 - 1^2 = 2.$$

On the other hand, the first order Taylor series approximation of f at the mean value $Z = 0$ is:

$$\hat{f}(Z) = 0^2 + (2Z)_{U=0}Z \equiv 0.$$

Thus, $\hat{p}(X)$ will simply be a delta function where all the mass is located at $X = 0$, a very poor approximation to p. ∎

M projection using quadrature

- Best Gaussian approx has these moments

$$
\begin{aligned}
E_p[X_i] &= \int_{-\infty}^{\infty} f_i(z) p_0(z) dz \\
E_p[X_i X_j] &= \int_{-\infty}^{\infty} f_i(z) f_j(z) p_0(z) dz.
\end{aligned}
$$

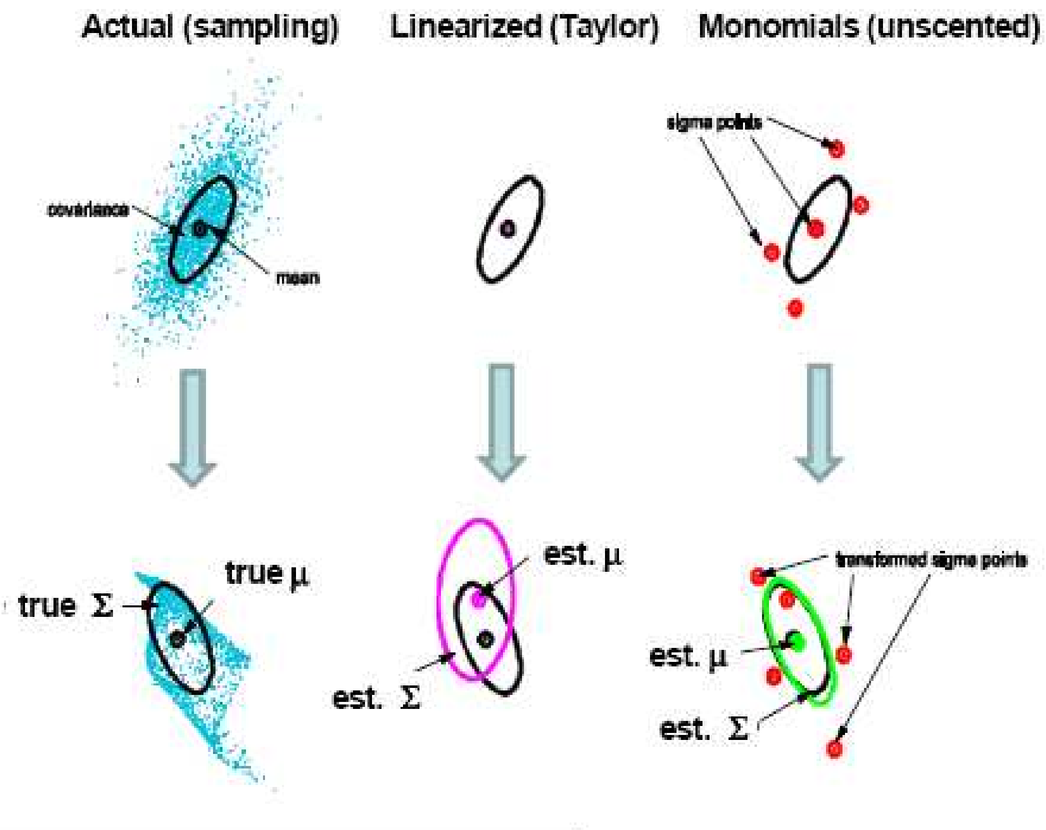
- Gaussian quadrature computes this integral for any W(z)>0 (here, Gaussian)

$$\int_a^b W(z) f(z) dz \approx \sum_{j=1}^{m} w_j f(z_j).$$

Unscented transform

- Pass mean and +- std in each dim through transform, and then fit Gaussian to transformed points

$$\int_{-\infty}^{\infty} W(z) f(z) dz \approx \left(1 - \frac{d}{\lambda^2}\right) f(0) + \sum_{i=1}^{d} \frac{1}{2\lambda^2} f(\lambda z_i^+) + \sum_{i=1}^{d} \frac{1}{2\lambda^2} f(\lambda z_i^-).$$



Nonlinear GMs

- We approximate nonlinear factors by approximating them by Gaussians
- The above methods require a joint Gaussian factor, not a canonical factor – we have to pass messages in topological order, and introduce variables one at a time to use the above tricks
- Linearization is done relative to current \mu. In EP, we iterate, and re-approximate each factor in the context of its incoming messages, which provides a better approx. to the posterior.
- Pretty hairy.

Discrete children, cts parents

- C -> D arcs are useful eg thermostat turns on/off depending on temperature
- We can approximate Gaussian * logistic by a Gaussian (variational approx)
- We can combine these Gaussian factors with the other factors as usual.

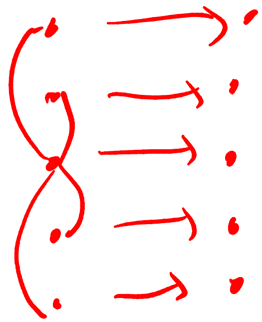
Sampling

- Sampling is the easiest way to handle cts and mixed variables
- “Collapsed particles” (Rao-Blackwellisation): sample the discretes, integrate out cts analytically. Each particle has a value for D and a Gaussian over C. Good for PF or MCMC.

$$\begin{tikzcd}
D_1 \arrow[r] \arrow[d] & D_2 \arrow[r] \arrow[d] & D_3 \arrow[d] \\
C_1 \arrow[r] \arrow[d] & C_2 \arrow[r] \arrow[d] & C_3 \arrow[d] \\
y_1 & y_2 & y_3
\end{tikzcd}$$

Non-parametric BP

- We can combine sampling and msg passing.
- We approximate factors/ msgs by samples.
- Factors are lower dimensional than full joints.
- Eg hand-pose tracking



Adaptive discretization

- We can discretize all the cts variables, then use a method for discrete vars.
- To increase accuracy, we expand the grid resolution for variables whose posterior entropy is high.
- Can use such approximations as proposal distributions for MH.