L05 – Generalized PGMs

(AIMA 13.3.2 or Barber 5.3) and AIMA 13.2.3 (or Koller and Friedman 7.1–7.2 (really best for Gaussian models))

What you should know after this lecture

- Conditioning on evidence in factor graph
- Max-product to find maximum-likelihood assignment
- Variable elimination in loopy graphs
- Intro to continuous graphical models

Inference in factor graphs

Some inference problems:

- Joint distribution: In a factor graph, use table multiplication to compute a big table

$$\frac{1}{Z}\prod_k \phi_k$$

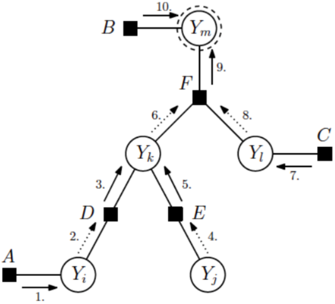
  where Z is the sum of all table entries
- Marginal distribution: $P(Y)$ where $Y \subset \mathcal{V}$
- Conditional probability: $P(Y \mid E = e)$, where $Y \subset \mathcal{V}$, $E \subset \mathcal{V}$, and $Y \cap E = \emptyset$; and e is the observed values of the variables in E. Note that it is not necessary that $Y \cup E = \mathcal{V}$.
- Most probable assignment (MAP):

$$\text{argmax}_y P(Y = y \mid E = e) \ .$$

  Note that the MAP of a set of variables is not necessarily the set of MAPs of the individual variables.

Sum-Product reminder

1. Select V_i as root
2. Recursively compute $P(V_i) \propto \prod_{\phi \in N(V_i)} \mu_{\phi \to V_i}$
3. Pass messages back down the tree, at each node computing marginal $P(V_j) \propto \prod_{\phi \in N(V_j)} \mu_{\phi \to V_j}$



Recall that $\propto$ means "proportional to," and we generally need to normalize to get a distribution.

Handling evidence

To compute $P(V \mid E = e)$, add a new potential for every variable $V_i \in E$ that assigns 1 to $V_i = e_i$ and 0 to all other values for V_i. Then run sum-product.

More than marginal!

Easy to compute $P(V_i, V_j)$ if they are connected in the graph via one factor ϕ:

$$P(V_i, V_j) \propto \phi \prod_{\phi_i \in N(V_i) \setminus \phi} \mu_{\phi_i \to V_i} \prod_{\phi_j \in N(V_j) \setminus \phi} \mu_{\phi_j \to V_j} \prod_{V_k \in N(\phi) \setminus \{V_i, V_j\}} \mu_{V_k \to \phi}$$

Multiply everything coming into V_i, V_j, and ϕ from elsewhere, and normalize

If they aren't neighbors, then for each value $V_i = v_i$, compute

$$P(V_i = v_i, V_j = v_j) = P(V_i = v_i \mid V_j = v_j) P(V_j = v_j)$$

using tools we have already established.

Finding most probable assignment in a factor graph

We can an algorithm very similar to sum product, called max product. Just as $ab + ac = a \cdot (b + c)$, $\max(ab, ac) = a \cdot \max(b, c)$ for non-negative a.
Do forward pass with messages as for sum-product, but

$$\mu_{\phi \to V}(v) = \max_{\bar{w} \in N(\phi) \setminus V} \phi(v, \bar{w}) \prod_{W \in N(\phi) \setminus V} \mu_{W \to \phi}(w)$$

Keep track of the values of W that yielded the max for each v:

$$M_V(v) = \operatorname*{argmax}_{\bar{w} \in N(\phi) \setminus V} \phi(v, \bar{w}) \prod_{W \in N(\phi) \setminus V} \mu_{W \to \phi}(w)$$

Decoding to find most probable assignment

Work backward from root V:

$$v^* = \operatorname*{argmax}_v \mathsf{P}(v)$$

Best value for each child W_i of V:

$$w_1^*, \ldots, w_k^* = M_V(v)$$

Handling loopy factor graphs

Exact inference is exponential in the number of variables in the "tree width" (largest group of variables that has to be considered jointly)

1. Cutset conditioning: pick a subset of nodes C such that, if they were removed, the remaining graph would be a tree. Iterate over assignments to C, do inference, and then reassemble the answers.
2. Variable elimination: iteratively,
    - Pick a variable V (efficiency depends on how you do this)
    - Define new $\phi' = \sum_v \prod_{\phi \in N(V)} \phi$
    - Remove V and all $\phi \in N(V)$ from graph
    - Add ϕ' (defined on all neighboring variables)
    - Until you have a tree (or one big table!)
3. Junction tree alg : complicated!

Variable elimination

Assume a factor graph such that

$$p(a, b, c, d, e) \propto \phi_{AB}(a, b)\phi_{AC}(a, c)\phi_{BCD}(b, c, d) \\ \phi_{DE}(d, e)\phi_{DF}(d, f)$$

Imagine we want to know $p(A)$.

$$p(a) \propto \sum_{b \in \Omega_B, c \in \Omega_C, d \in \Omega_D, e \in \Omega_E, f \in \Omega_f} \phi_{AB}(a, b)\phi_{AC}(a, c)\phi_{BCD}(b, c, d) \\ \phi_{DE}(d, e)\phi_{DF}(d, f)$$

Eliminate F

Consider "eliminating" variable F: push the sum over F followed by all factors involving F to the end

$$p(a) \propto \sum_{b\in\Omega_B, c\in\Omega_C, d\in\Omega_D, e\in\Omega_E} \phi_{AB}(a,b)\phi_{AC}(a,c)\phi_{BCD}(b,c,d)$$
$$\phi_{DE}(d,e) \sum_{f\in\Omega_f} \phi_{DF}(d,f)$$

Find all the other variables $U_1, \ldots, U_k$ involved in any factors mentioning F (in this case it's just D). Call those factors $\phi'_1, \ldots \phi'_m$. Make a new factor ϕ_1 on $U_1, \ldots U_k$ defined (using table multiplication) by: $\phi_1 = \sum_{f\in\Omega f} \phi'_1 \cdot \ldots \cdot \phi'_m$
In our case $\phi_1(d) = \sum_{f\in\Omega f} \phi_{DF}(d,f)$. Now, we have a new, equivalent (in terms of its distribution on all the other variables), factor graph

$$p(a,b,c,d,e) \propto \phi_{AB}(a,b)\phi_{AC}(a,c)\phi_{BCD}(b,c,d)\phi_{DE}(d,e)\phi_1(d)$$

Eliminate E

Now let's eliminate variable E: push the sum over E followed by all factors involving E to the end

$$p(a) \propto \sum_{b \in \Omega_B, c \in \Omega_C, d \in \Omega_D} \phi_{AB}(a,b)\phi_{AC}(a,c)\phi_{BCD}(b,c,d)\phi_1(d) \sum_{e \in \Omega_E} \phi_{DF}(d,e)$$

Find all the other variables $U_1, \dots, U_k$ involved in any factors mentioning E (in this case it's just D). Call those factors $\phi'_1, \dots \phi'_m$. Make a new factor ϕ_2 on $U_1, \dots U_k$ defined (using table multiplication) by: $\phi_2 = \sum_{e \in \Omega E} \phi'_1 \cdot \dots \cdot \phi'_m$.
In our case $\phi_2(d) = \sum_{e \in \Omega E} \phi_{DE}(d,e)$. Now, we have a new, equivalent (in terms of its distribution on all the other variables), factor graph

$$p(a,b,c,d) \propto \phi_{AB}(a,b)\phi_{AC}(a,c)\phi_{BCD}(b,c,d)\phi_1(d)\phi_2(d)$$

Eliminate D

Now let's eliminate variable D: push the sum over D followed by all factors involving D to the end

$$p(a) \propto \sum_{b \in \Omega_B, c \in \Omega_C} \phi_{AB}(a,b)\phi_{AC}(a,c) \sum_{d \in \Omega_D} \phi_{BCD}(b,c,d)\phi_1(d)\phi_2(d)$$

Find all the other variables $U_1, \ldots, U_k$ involved in any factors mentioning D (in this case it's B, C). Call those factors $\phi'_1, \ldots \phi'_m$. Make a new factor $\phi_3 = \sum_{d \in \Omega D} \phi'_1 \cdot \ldots \phi'_m$
In our case $\phi_3(b,c) = \sum_{d \in \Omega D} \phi_{BCD}(b,c,d)\phi_1(d)\phi_2(d)$. Now, we have a new, equivalent (in terms of its distribution on all the other variables), factor graph

$$p(a,b,c) \propto \phi_{AB}(a,b)\phi_{AC}(a,c)\phi_3(b,c)$$

Eliminate C

Now let's eliminate variable C: push the sum over C followed by all factors involving C to the end

$$p(a) \propto \sum_{b \in \Omega_B, c \in \Omega_C} \phi_{AB}(a, b) \sum_{c \in \Omega_C} \phi_{AC}(a, c) \phi_3(b, c)$$

Find all the other variables $U_1, \ldots, U_k$ involved in any factors mentioning C (in this case it's A, B). Call those factors $\phi'_1, \ldots \phi'_m$. Make a new factor ϕ_4 on $U_1, \ldots U_k$ defined (using table multiplication) by:

$$\phi_4 = \sum_{c \in \Omega C} \phi'_1 \cdot \ldots \cdot \phi'_m$$

In our case $\phi_4(a, b) = \sum_{c \in \Omega C} \phi_{AC}(a, c) \phi_3(b, c)$. Now, we have a new, equivalent (in terms of its distribution on all the other variables), factor graph

$$p(a, b) \propto \phi_{AB}(a, b) \phi_4(a, b)$$

Eliminate B

Now let's eliminate variable B: push the sum over B followed by all factors involving B to the end

$$p(a) \propto \sum_{b \in \Omega_B} \phi_{AB}(a, b)\phi_4(a, b)$$

Compute $\phi_5(a) = \sum_{b \in \Omega_B} \phi_{AB}(a, b)\phi_4(a, b)$. Now, we have a new, equivalent (in terms of its distribution on all the other variables), factor graph

$$p(a) \propto \phi_5(a)$$

Yay!

Facts about variable elimination

- Computational complexity is exponential in the number of variables in the biggest factor you have to compute along the way
- This depends on variable order! What if we choose to eliminate D first in this problem?
- It's NP-hard to find the optimal variable order.
- Still, there are heuristics that can make this a good strategy.

Conjugate families of probability distributions

In order for exact probabilistic inference to be tractable, we generally need for the joint and conditional distributions of factors to be conjugate:[1]

- Let $f(\theta_A)(a)$ be the pdf of a random variable A and $f(\theta_B)(b)$ be the pdf of a random variable B, where f has some fixed parametric form and θ specifies a particular pdf in that family.
- Then the product of the pdfs on A and B has the form $f(\theta_{AB})(a, b)$ where θ_{AB} is a function of θ_A and θ_B.

$$f(\theta_A)(a) \cdot f(\theta_B)(b) = f(\theta_{AB})(a, b) = f(g(\theta_a, \theta_b))(a, b)$$

[1] The actual definition is more general and specifically relates a prior distribution and an observation distribution, but this basic idea is what we need for now.

Categorical distribution is conjugate family

We have been using the categorical distribution[2]

- $\Omega = \{x_1, \ldots, x_M\}$
- $\theta^A = (\theta_1^A, \ldots, \theta_M^A)$ $\qquad\qquad$ $\theta^B = (\theta_1^B, \ldots, \theta_M^B)$
- $f_A(\theta^A)(x_i) = \theta_i^A$ $\qquad\qquad$ $f_B(\theta^B)(x_i) = \theta_i^B$

If we multiply these functions on the same variable (e.g. during message passing), then we get

- $f_{AB}(\theta_{AB})(x_i) = \theta_i^{AB} = \frac{1}{Z}\theta_i^A \cdot \theta_i^B$

where $Z = \sum_{i=1}^{M} \theta_i^A \theta_i^B$

[2] We like the name “multinoulli” better, though!

Categorical distribution is conjugate for joint

Combining two categorical distributions on <u>different</u> variables:

- $\Omega_A = \{a_1, \ldots, a_M\}$ $\qquad$ $\Omega_B = \{b_1, \ldots, b_N\}$
- $\theta^A = (\theta^A_1, \ldots, \theta^A_M)$ $\qquad$ $\theta^B = (\theta^B_1, \ldots, \theta^B_N)$
- $f_A(\theta^A)(a_i) = \theta^A_i$ $\qquad$ $f_B(\theta^B)(b_i) = \theta^B_i$

If we multiply these functions on different variables (e.g. computing the joint when A and B are independent), then we get

- $\Omega_{AB} = \Omega_A \times \Omega_B$
- $f_{AB}(\theta^{AB})(a_i, b_j) = \theta^{AB}(a_i, b_j) = \theta^A_i \cdot \theta^B_j$

Univariate Gaussian is conjugate family

- $\Omega = \mathbb{R}$
- $\theta_A = (\mu_A, \sigma_A^2)$ $\qquad$ $\theta_B = (\mu_B, \sigma_B^2)$
- $f_A(\theta_A)(x) = \frac{1}{\sqrt{2\pi}\sigma_A} \exp\{-\frac{1}{2\sigma_A^2}(x - \mu_A)^2\}$
- $f_B(\theta_B)(x) = \frac{1}{\sqrt{2\pi}\sigma_B} \exp\{-\frac{1}{2\sigma_B^2}(x - \mu_B)^2\}$

If we multiply these functions on the same variable (e.g. during Bayes rule), then

- Observe that multiplying f's yields

$$f_{AB}(\theta_{AB})(x) = \frac{1}{\sqrt{2\pi}\sigma_A}\frac{1}{\sqrt{2\pi}\sigma_B} \exp\{-\frac{1}{2\sigma_A^2}(x - \mu_A)^2 - \frac{1}{2\sigma_B^2}(x - \mu_B)^2\}$$

- After completing the square and some algebra, we find that $f_{AB}(\theta_{AB})(x) = \frac{1}{\sqrt{2\pi}\sigma_{AB}} \exp\{-\frac{1}{2\sigma_{AB}^2}(x - \mu_{AB})^2\}$ where

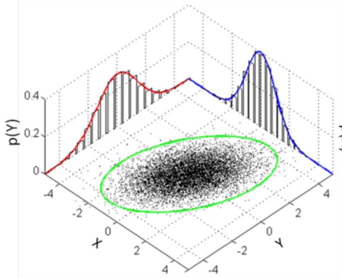
$$\mu_{AB} = \frac{\mu_A \sigma_B^2 + \mu_B \sigma_A^2}{\sigma_A^2 + \sigma_B^2} \qquad \sigma_{AB}^2 = \frac{\sigma_A^2 \sigma_B^2}{\sigma_A^2 + \sigma_B^2}$$

Multivariate Gaussian

- $\Omega = \mathbb{R}^D$
- $\theta = (\mu \in \mathbb{R}^D, \Sigma \in \mathbb{R}^{D \times D})$ `// Σ is positive definite

$$f(\mu, \Sigma)(x) = \frac{1}{\sqrt{2\pi^D |\Sigma|}} \exp\{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\}$$

$|\Sigma|$ is the determinant; figure from Wikipedia



- Axes are eigenvectors of Σ
- Axis-aligned if Σ is diagonal
- Round if Σ is identity

Fun facts about the multivariate Gaussian

Let's say our MVG has dimensions $1..D$, but we are interested in marginalizing some of them out, or conditioning some of them on particular values. Let's divide them into one set of dimensions $A = 1..K$ and another $B = K+1..D$. So, we can think of the parameters as

$$\mu = \begin{pmatrix} \mu_A \\ \mu_B \end{pmatrix} \quad \Sigma = \begin{pmatrix} \Sigma_{AA} & \Sigma_{AB} \\ \Sigma_{BA} & \Sigma_{BB} \end{pmatrix}$$

Marginalizing out dimensions A yields Gaussian on B with

$$\mu_B^m = \mu_B \quad \Sigma_B^m = \Sigma_{BB}$$

Conditioning on $B = b$ yields a Gaussian on A with

$$\mu_{A|B}^c = \mu_A + \Sigma_{AB}\Sigma_{BB}^{-1}(b - \mu_B) \quad \Sigma_{A|B}^c = \Sigma_{AA} - \Sigma_{AB}\Sigma_{BB}^{-1}\Sigma_{BA}$$

For random variables $X_1, \ldots, X_n$ that are jointly Gaussian with parameters μ, Σ:

- The mean of $c_0 + \sum_i c_i X_i$, where the c_i are constants, is $c_0 + \sum_i c_i \mu_i$
- The variance of $c_0 + \sum_i c_i X_i$ is $c^T \Sigma c$

Multivariate Gaussian is conjugate family

Product of MVGs:

- $\Omega_A = \mathbb{R}^D$ $\qquad\qquad$ $\Omega_B = \mathbb{R}^D$
- $\theta_A = (\mu_A, \Sigma_A)$ $\qquad\qquad$ $\theta_B = (\mu_B, \Sigma_B)$

If we multiply these functions on the same variable (e.g. during Bayes rule), then we get an MVG with

$$\mu_{AB} = \left(\Sigma_A^{-1} + \Sigma_B^{-1}\right)^{-1} \left(\Sigma_A^{-1}\mu_A + \Sigma_B^{-1}\mu_B\right) \qquad \Sigma_{AB} = \left(\Sigma_A^{-1} + \Sigma_B^{-1}\right)^{-1}$$

Can be useful to define <u>precision</u> : $\Lambda = \Sigma^{-1}$
Then $\Lambda_{AB} = \Lambda_A + \Lambda_B$ and

$$\mu_{AB} = (\Lambda_A + \Lambda_B)^{-1}(\Lambda_A \mu_A + \Lambda_B \mu_b)$$

Multivariate Gaussian is conjugate for joint

Product of MVGs on different domains

- $\Omega_A = \mathbb{R}^{D_A}$ $\qquad \Omega_B = \mathbb{R}^{D_B}$
- $\theta_A = (\mu_A, \Sigma_A)$ $\qquad \theta_B = (\mu_B, \Sigma_B)$

We get an MVG with dimension $D = D_A + D_B$, and

$$\mu = \begin{pmatrix} \mu_A \\ \mu_B \end{pmatrix} \quad \Sigma = \begin{pmatrix} \Sigma_A & 0 \\ 0 & \Sigma_B \end{pmatrix}$$

Gaussian Bayesian networks

Assume the conditional probability distribution for each node V has the form $V \sim \text{Normal}(w_V^0 + w_V^\mathsf{T} \cdot \text{pa}(V), \eta_V^2)$ where

- w_v is a vector of real-valued weights of length $N - 1$ (number of parents of V) and w_0 is a scalar offset
- η_V^2 is the variance of added noise at this node

then the joint distribution over all variables $V_1, \ldots, V_N, V$ is Gaussian.

- Assume the parents of node V are normally distributed with mean μ_P, Σ_P the distribution over V is normal with
- $\mu_V = w_V^0 + W_V^\mathsf{T} \mu_P$
- $\sigma_V^2 = \eta_V^2 + w^\mathsf{T} \Sigma_P w$

Gaussian Bayesian networks

- Assume distribution $V \sim \text{Normal}(w_V^0 + w_V^T \cdot \text{pa}(V), \eta_V^2)$
- Assume the parents of V are normally distributed with mean μ_P, Σ_P

then the joint distribution over all variables $V_1, \ldots, V_N, V$ is Gaussian with

- Mean: μ_P, μ_V
- Cov:

$$\begin{bmatrix} \Sigma_P & \Sigma_{PV} \\ \Sigma_{PV}^T & \sigma_V^2 \end{bmatrix}$$

where $\Sigma_{PV}[i] = \sum_j \Sigma_P[i, j]$

By induction, you can show that a whole Bayes net with this linear Gaussian structure defines a joint Gaussian distribution!

Hybrid networks

Some standard cases:

- Discrete parent of Gaussian nodes: mixture-of-Gaussians models
- Continuous parent of discrete node: apply sigmoid or softmax to get categorical distribution

Gaussian Factor graphs

Make a factor graph in which all potentials are described using μ, Σ over their neighbor variables.

- Joint distribution (suitably normalized) is a multivariate Gaussian
- If the graph is a tree, you can do belief propgation, using exactly the same algorithmic structure as sum-product, but using operations on Gaussian-PDF-form functions:
  - Multiply
  - Marginalize
- It turns out that it's usually easier to do it with messages representing the same information as μ, Σ but in a different ("canonical") form. We're not going to look at it in detail.

Next time

- Approximate inference via sampling