Graphical Models For Complex Health Data (P8124)

Daniel Malinsky

Columbia University
dsm2128@cumc.columbia.edu

Approximate Inference

Inference with graphical models

In (bio)statistics, “inference” usually refers to:

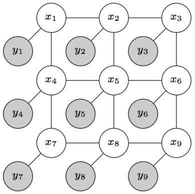
- Point estimates of some statistical parameter (with consistency properties)
- Confidence intervals for some parameter (with coverage property)
- Hypothesis testing (with error rate control)
- Estimating posterior distributions over parameters (Bayesian inference)
- Conclusions about population parameters, quantifying uncertainty

Inference with graphical models

In some GM literature (esp. computer science), "inference" refers to answering various "queries" based on a given model:

- Some marginal distribution $p(x_i, x_j, x_k)$ from a given joint $p(x_1, ..., x_p)$
- The probability of some *outcome* given some observed *evidence*: $p(x|y)$ where some variables Y are observed to be y.
  - Ex: "what is the probability of *pneumonia* given symptoms $y_1, ..., y_d$?"
- MAP: $\arg\max_x p(x|y)$
  - Ex: "what is the most likely disease given symptoms $y_1, ..., y_d$?"
  - Ex: "where will the car most likely be located given its current coordinates?"

Example: denoising with an MRF



Hidden variables $x_i \in \{-1, +1\}$ (black and white) and observed variables $y_i \in \mathbb{R}$ (noisy greyscale observations).

Want: $p(x|y)$ to determine for each pixel whether it is more likely black or white.

Exact inference

Given a graph $\mathcal{G}$ (usually DAG or UG) and the joint $p(x)$ that factorizes, how can I efficiently compute the query? There is large literature of algs that exploit graph structure to do computation w/ minimal steps.

The most famous approach is called *belief propagation*.

Unfortunately, introducing BP takes quite a bit of time and to be honest, it is much less often used in real data analysis these days compared to approximate inference approaches. So we will focus on approximate inference in this course instead.

Approximate inference

Approximate methods mostly fall into two categories:

- Variational Inference (VI): includes mean-field approx, loopy belief prop, tree-reweighting
- Sampling / Monte Carlo: includes MCMC and related (next time)

Which approach to use? Of course, it depends on the setting. VI is typically faster and easier to scale to large data. It is also more amenable to sophisticated optimization methods like stochastic optimization. Drawback: less developed theoretical properties, few guarantees, sometimes too approximate.

MCMC methods, on the other hand, come with guarantees about being asymptotically exact. May be better suited to smaller data sets but where we need more precise samples.

$\Rightarrow$ Both approaches are useful as tools!

Inference as an optimization problem

Goal: approximate some difficult distribution $p(x|y)$ with a new distribution $q(x)$ such that:

- $p(x|y)$ and $q(x)$ are "close" in some way
- computation on $q(x)$ is easy

Note: going to assume generically that X is the set of variables we care about (want to know distribution of) and $Y = y$ is the evidence we condition on. In some presentations, especially more Bayesian-oriented ones, people think of X as a set of latent (hidden) variables and Y is observed data, so you're approximating the "posterior" distribution of the latent variables, given your data. Alternatively it might be a posterior over a set of parameters, e.g., $p(\theta|y)$. VI is very popular in Bayesian inference.

Inference as an optimization problem

Assume $q(x)$ is chosen from some tractable family of distributions $\mathcal{Q}$. We assume q has some free parameters which we want to optimize to be "close" to p. How do we measure closeness?

A natural choice is the Kullback-Leibler divergence:

$$D_{KL}(q||p) = \sum_x q(x) \log \frac{q(x)}{p(x|y)}$$

Inference as an optimization problem

$D_{KL}(q||p) \neq D_{KL}(p||q)$. Which one should we choose to optimize?

$$D_{KL}(q||p) = \sum_x q(x) \log \frac{q(x)}{p(x|y)}$$

is called the I-projection (or "reverse" KL)

$$D_{KL}(p||q) = \sum_x p(x|y) \log \frac{p(x|y)}{q(x)}$$

is called the M-projection (or "forwards" KL)

Inference as an optimization problem

$D_{KL}(q||p) \neq D_{KL}(p||q)$. Which one should we choose to optimize?

$$D_{KL}(q||p) = \sum_x q(x) \log \frac{q(x)}{p(x|y)}$$

is called the I-projection (or "reverse" KL)

$$D_{KL}(p||q) = \sum_x p(x|y) \log \frac{p(x|y)}{q(x)}$$

is called the M-projection (or "forwards" KL)

$\Rightarrow$ the M-projection is hard to optimize because of expectation wrt $p(x|y)$

$\Rightarrow$ the I-projection is *zero forcing*, tends to under-estimate the supp(p).

$\Rightarrow$ the M-projection is *zero avoiding*, tends to over-estimate the supp(p).

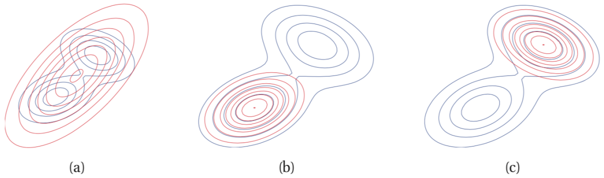


Figure 21.1 Illustrating forwards vs reverse KL on a bimodal distribution. The blue curves are the contours of the true distribution p. The red curves are the contours of the unimodal approximation q. (a) Minimizing forwards KL: q tends to "cover" p. (b-c) Minimizing reverse KL: q locks on to one of the two modes. Based on Figure 10.3 of (Bishop 2006b). Figure generated by `KLfwdReverseMixGauss`.

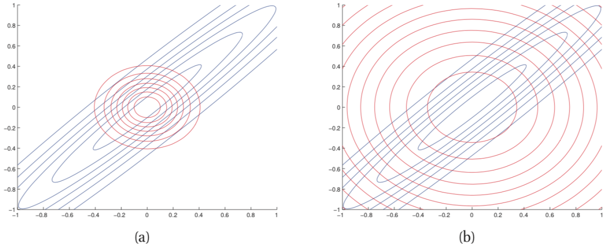


Figure 21.2 Illustrating forwards vs reverse KL on a symmetric Gaussian. The blue curves are the contours of the true distribution p. The red curves are the contours of a factorized approximation q. (a) Minimizing $\mathbb{KL}(q||p)$. (b) Minimizing $\mathbb{KL}(p||q)$. Based on Figure 10.2 of (Bishop 2006b). Figure generated by `KLpqGauss`.

Inference as an optimization problem

Typically we'll try to minimize $D_{KL}(q||p)$. But this is also difficult, since the normalizing constant $Z \equiv p(y)$ in $p(x|y)$ is expensive to compute. So define $\tilde{p}(x) \equiv p(x,y) = p(x|y)Z$ and optimize:

$$\begin{aligned} J(q) &\equiv D_{KL}(q||\tilde{p}) \\ &= \sum_x q(x) \log \frac{q(x)}{\tilde{p}(x)} \\ &= \sum_x q(x) \log \frac{q(x)}{p(x|y)Z} \\ &= \sum_x q(x) \log \frac{q(x)}{p(x|y)} - \log Z \\ &= D_{KL}(q||p) - \log Z \end{aligned}$$

Since Z is a constant, minimizing $J(q)$ forces q to become close to p.

Bounds on the evidence

$J(q)$ is sometimes called the variational free energy or Helmholtz free energy. It is an upper bound on the negative log likelihood of the evidence:

$$J(q) = D_{KL}(q||p) - \log Z \geq -\log Z = -\log p(y)$$

since $D_{KL}(\cdot||\cdot) \geq 0$.

Equivalently, we could **maximize** $L(q) \equiv -J(q)$

$$L(q) = -D_{KL}(q||p) + \log Z \leq \log Z = \log p(y)$$

which is called the evidence lower bound (ELBO). Also called the energy functional.

Mean field

So, we want to maximize $L(q)$, solve:

$$\arg\max_{q \in \mathcal{Q}} L(q)$$

The difficulty of this optimization problem depends on the chosen family $\mathcal{Q}$. It is often convenient to choose something simple for $\mathcal{Q}$, to make life (relatively) easy.
For example, the mean field method assumes

$$q(x) = \prod_{i=1}^{p} q_i(x_i)$$

$$\arg\max_{q_1,\ldots,q_p} L(q)$$

where we optimize over the parameters of each marginal q_j

$$
\begin{aligned}
L(q_j) &= \sum_x q(x) \log \frac{p(x,y)}{q(x)} \\
&= \sum_x \prod_{i=1}^{p} q_i(x_i) \left[\log p(x,y) - \sum_{k=1}^{p} \log q_k(x_k) \right] \\
&= \sum_{x_j} \sum_{x_{-j}} q_j(x_j) \prod_{i \neq j} q_i(x_i) \left[\log p(x,y) - \sum_{k=1}^{p} \log q_k(x_k) \right] \\
&= \sum_{x_j} q_j(x_j) \sum_{x_{-j}} \prod_{i \neq j} q_i(x_i) \log p(x,y) \\
&\quad - \sum_{x_j} q_j(x_j) \sum_{x_{-j}} \prod_{i \neq j} q_i(x_i) \left[\sum_{k \neq j} \log q_k(x_k) + \log q_j(x_j) \right] \\
&= \sum_{x_j} q_j(x_j) \log f_j(x_j) - \sum_{x_j} q_j(x_j) \log q_j(x_j) + \text{const}
\end{aligned}
$$

where $\log f_j(x_j) \equiv \sum_{x_{-j}} \prod_{i \neq j} q_i(x_i) \log p(x,y) = \mathbb{E}_{q_{-j}}[\log p(x,y)]$

So we average out all the variables except for x_j. We can rewrite $L(q_j)$ as

$$L(q_j) = -D_{KL}(q_j || f_j)$$

which we maximize by setting $q_j = f_j$

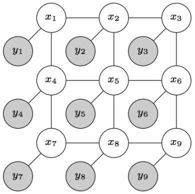
$$\log q_j(x_j) = \mathbb{E}_{q_{-j}}[\log p(x, y)] + \text{const}$$

Mean field

When updating q_j, we only need to reason about the variables which share a factor with x_j, i.e., the terms in j's Markov blanket; the other terms get absorbed into the constant term. Since we are replacing the neighboring values by their mean value, the method is known as mean field.

Let's do a concrete example.

Denoising with an MRF



Hidden variables $x_i \in \{-1, +1\}$ (black and white) and observed variables $y_i \in \mathbb{R}$ (noisy greyscale observations) with $\phi_i(x_i) = p(y_i|x_i) \equiv L_i(x_i)$ and $\phi_{ij}(x_i x_j) = W_{ij} x_i x_j$.

Denoising with an MRF

Hidden variables $x_i \in \{-1, +1\}$ (black and white) and observed variables $y_i \in \mathbb{R}$ (noisy greyscale observations) with $\phi_i(x_i) = p(y_i|x_i) \equiv L_i(x_i)$ and $\phi_{ij}(x_i x_j) = W_{ij} x_i x_j$.

Prior $p(x) = \frac{1}{Z_0} \exp(\sum_{i=1}^{p} \sum_{j \in \text{Ne}(i)} W_{ij} x_i x_j)$
(note: using Ne(i) here to denote *hidden* neighbors)

Likelihood $p(y|x) = \prod_i p(y_i|x_i) = \sum_i \exp(-L_i(x_i))$

Posterior $p(x|y) = \frac{1}{Z} \exp(\sum_{i=1}^{p} \sum_{j \in \text{Ne}(i)} W_{ij} x_i x_j + \sum_i L_i(x_i))$

Approximate this by $q(x) = \prod_i q(x_i; \mu_i)$ where μ_i is the mean value of node i. Want to find optimal values of μ_i.

Denoising with an MRF

Using $\log q_i(x_i) = \mathbb{E}_{q_{-i}}[\log p(x,y)] + \text{const}$

we get

$$q_i(x_i) \propto \exp\left(x_i \sum_{j\in \text{Ne}(i)} W_{ij}\mu_j + L_i(x_i)\right)$$

thus we replace the states of the neighbors by their average values. Let $m_i = \sum_{j\in\text{Ne}(i)} W_{ij}\mu_j$ be the mean field influence on node i. Then we approximate the marginal posteriors by

$q_i(x_i = 1) = \frac{\exp(m_i + L_i(1))}{\exp(m_i+L_i(1))+\exp(-m_i+L_i(-1))} = \text{expit}(2a_i)$
where $a_i \equiv m_i + 0.5(L_i(1) - L_i(-1))$

similarly $q_i(x_i = -1) = \text{expit}(-2a_i)$. From this we compute:
$\mu_i = E_{q_i}[x_i] = q_i(x_i=1)(1) + q_i(x_i=-1)(-1) =$
$\frac{1}{1+\exp(-2a_i)} - \frac{1}{1+\exp(2a_i)} = \tanh(a_i)$

Denoising with an MRF

So we set up update equations by:

$$\mu_i^t = \tanh\left(\sum_{j \in \mathrm{Ne}(i)} W_{ij}\mu_j^{t-1} + 0.5(L_i(1) - L_i(-1))\right)$$

typically people use "damped updates"

$$\mu_i^t = (1-\lambda)\mu_i^{t-1} + \lambda \tanh\left(\sum_{j \in \mathrm{Ne}(i)} W_{ij}\mu_j^{t-1} + 0.5(L_i(1) - L_i(-1))\right)$$

for some $0 < \lambda < 1$ which has better performance.

Example

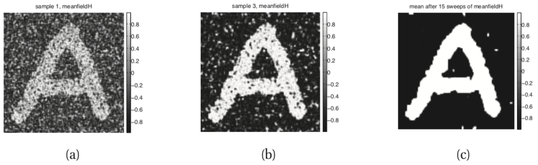


Figure 21.3 Example of image denoising using mean field (with parallel updates and a damping factor of 0.5). We use an Ising prior with $W_{ij} = 1$ and a Gaussian noise model with $\sigma = 2$. We show the results after 1, 3 and 15 iterations across the image. Compare to Figure 24.1. Figure generated by `isingImageDenoiseDemo`.

Structured mean field

Our assumption that

$$q(x) = \prod_{i=1}^{p} q_i(x_i)$$

is pretty naive; we could get better approximations by allowing for dependence among the q_i, and updating sets of variables together. This is called *structured* mean field.

Structured mean field

For example, for an HMM, a good approximation may be a product of chain-structured models:

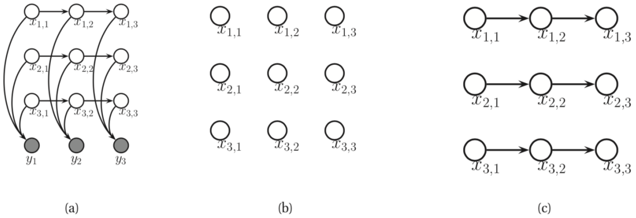


Figure 21.4 (a) A factorial HMM with 3 chains. (b) A fully factorized approximation. (c) A product-of-chains approximation. Based on Figure 2 of (Ghahramani and Jordan 1997).

Consider M chains each of length T. Each hidden node X_{tm} has K states. Y_t is observed and continuous. Want: posterior $p(x|y)$. Try: approximate by $q(x)$ with form in (c).

Structured mean field: HMM example

$$p(x, y) = \prod_{m=1}^{M} \prod_{t=1}^{T} p(x_{tm} | x_{t-1,m}) p(y_t | x_t)$$

where $p(x_{tm} = k | x_{t-1,m} = j) = A_{jmk}$ transition matrix and $p(x_{1m} = k | x_{0m}) = \pi_{mk}$ is initial state dist. for chain m.

$$Y_t \mid X_t = x_t \sim N(\sum_{m=1}^{M} W_m x_{tm}, \Sigma)$$

Structured mean field: HMM example

$$q(x|y) = \frac{1}{Z_q} \prod_{m=1}^{M} q(x_{1m}; \xi_{1m}) \prod_{t=2}^{T} q(x_{tm}|x_{t-1,m}; \xi_{tm})$$

$$q(x_{1m}; \xi_{1m}) = \prod_{k=1}^{K} (\xi_{1mk} \pi_{mk})^{x_{1mk}}$$

$$q(x_{tm}|x_{t-1,m}; \xi_{tm}) = \prod_{k=1}^{K} \left(\xi_{tmk} \prod_{j=1}^{K} (A_{mjk})^{x_{t-1,m,j}} \right)^{x_{tmk}}$$

Structured mean field: HMM example

Can show the posterior distribution has a simple form, function of x, ξ, π, A. Maximize ELBO and derive update equations of the form:

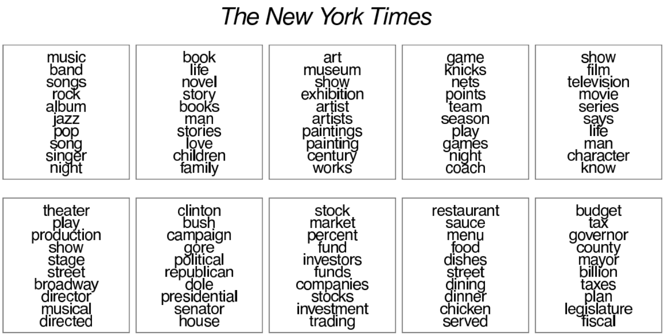
$$\xi_{tm} = \exp(W_m^T \Sigma^{-1} \tilde{y}_{tm} - \frac{1}{2}\delta_m)$$

$$\delta_m = \text{diag}(W_m^T \Sigma^{-1} W_m)$$

$$\tilde{y}_{tm} = y_t - \sum_{l \neq m}^{M} W_l \, \mathbb{E}[x_t, l]$$

See Murphy (2012, sec. 21.4) for details.

Application: topic modeling w/ stochastic variational inference



The New York Times

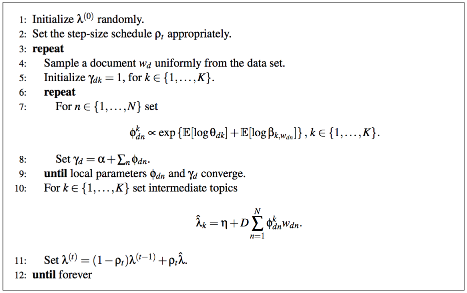
music	book	art	game	show
band	life	museum	knicks	film
songs	novel	show	nets	television
rock	story	exhibition	points	movie
album	books	artist	team	series
jazz	man	artists	season	says
pop	stories	paintings	play	life
song	love	painting	games	man
singer	children	century	night	character
night	family	works	coach	know

theater	clinton	stock	restaurant	budget
play	bush	market	sauce	tax
production	campaign	percent	menu	governor
show	gore	fund	food	county
stage	political	investors	dishes	mayor
street	republican	funds	street	billion
broadway	dole	companies	dining	taxes
director	presidential	stocks	dinner	plan
musical	senator	investment	chicken	legislature
directed	house	trading	served	fiscal

Posterior topics learned from 1.8M documents from NYT. 461M observed words from a vocab of 8K terms. Hoffman et al. (2013) "Stochastic Variational Inference," *Journal of Machine Learning*.

Application: topic modeling w/ stochastic variational inference

Hoffman et al. use basically the same LDA (latent Dirichlet allocation) model we discussed previously. They derive update equations using mean-field.



```
1:  Initialize $\lambda^{(0)}$ randomly.
2:  Set the step-size schedule $\rho_t$ appropriately.
3:  repeat
4:    Sample a document $w_d$ uniformly from the data set.
5:    Initialize $\gamma_{dk} = 1$, for $k \in \{1, \ldots, K\}$.
6:    repeat
7:      For $n \in \{1, \ldots, N\}$ set
            $\phi_{dn}^k \propto \exp\{\mathbb{E}[\log \theta_{dk}] + \mathbb{E}[\log \beta_{k,w_{dn}}]\}, k \in \{1, \ldots, K\}.$
8:      Set $\gamma_d = \alpha + \sum_n \phi_{dn}$.
9:    until local parameters $\phi_{dn}$ and $\gamma_d$ converge.
10:   For $k \in \{1, \ldots, K\}$ set intermediate topics
            $\hat{\lambda}_k = \eta + D \sum_{n=1}^{N} \phi_{dn}^k w_{dn}.$
11:   Set $\lambda^{(t)} = (1 - \rho_t)\lambda^{(t-1)} + \rho_t \hat{\lambda}$.
12: until forever
```

Figure 6: Stochastic variational inference for LDA. The relevant expectations for each update are found in Figure 5.

They use a stochastic gradient ascent version of mean-field to scale up VI to massive data.