Readings: 18.1, 18.2, 18.3

EM for BNs

Graphical Models – 10708
Carlos Guestrin
Carnegie Mellon University

November 24th, 2008

Thus far, fully supervised learning

- We have assumed fully supervised learning:

  ⟨A=t, F=f, S=t, H=t, N=f⟩

  ⋮

- Many real problems have missing data:

  ⟨A=t, F=f, S=?, H=t, N=?⟩

The general learning problem with missing data

- Marginal likelihood – **x** is observed, **z** is missing:

observed

$$\begin{aligned}
\ell(\mathcal{D} : \theta) &= \log \prod_{j=1}^{m} P(x^{(j)} \mid \theta) \\
&= \sum_{j=1}^{m} \log P(x^{(j)} \mid \theta) \\
&= \boxed{\sum_{j=1}^{m} \log \sum_{z} P(z, x^{(j)} \mid \theta)}
\end{aligned}$$

E-step

- **x** is observed, **z** is missing
- Compute probability of missing data given current choice of θ
  - Q(**z**|**x**$^{(j)}$) for each **x**$^{(j)}$
    - e.g., probability computed during classification step
    - corresponds to "classification step" in K-means

$$\boxed{Q^{(t+1)}(z \mid x^{(j)}) = P(z \mid x^{(j)}, \theta^{(t)})}$$

inference in a BN

Jensen's inequality

$$\ell(\mathcal{D}:\theta) = \boxed{\sum_{j=1}^{m} \log \sum_{z} P(z, x^{(j)} \mid \theta)}$$

- **Theorem**: log $\sum_z$ P(**z**) f(**z**) ≥ $\sum_z$ P(**z**) log f(**z**)

$\log 2 = \log(1+1) \geq \log 1 + \log 1 = 0$

Applying Jensen's inequality

EM: optimizes lower bound on $\ell(D:\theta)$

- Use: log $\sum_z$ P(**z**) f(**z**) ≥ $\sum_z$ P(**z**) log f(**z**)

$$\ell(\mathcal{D}:\theta^{(t)}) = \sum_{j=1}^{m} \log \sum_{z} \underbrace{Q^{(t+1)}(z \mid x^{(j)})}_{\text{"P(z)"}} \boxed{\frac{P(z, x^{(j)} \mid \theta^{(t)})}{Q^{(t+1)}(z \mid x^{(j)})}}$$

"f(z)"

$$\ell(D:\theta^{(t)}) \geq \sum_{j=1}^{m} \sum_{z} Q^{(t+1)}(z \mid x^{(j)}) \log \frac{P(z, x^{(j)} \mid \theta^{(t)})}{Q^{(t+1)}(z \mid x^{(j)})}$$

wrt. θ, Q

$$= \underbrace{\sum_{j=1}^{m} \sum_{z} Q^{(t+1)}(z \mid x^{(j)}) \log P(z, x^{(j)} \mid \theta^{(t)})}_{\text{weighted log-likelihood of fully observed data}} - \underbrace{\sum_{j=1}^{m} \sum_{z} Q^{(t+1)}(z \mid x^{(j)}) \log Q^{(t+1)}(z \mid x^{(j)})}_{m \hat{H}_{Q^{(t+1)}}(Z \mid X)}$$

The M-step maximizes lower bound on weighted data

constant

- Lower bound from Jensen's:

$$\ell(\mathcal{D}:\theta^{(t)}) \geq \underbrace{\sum_{j=1}^{m}\sum_{z} Q^{(t+1)}(z \mid x^{(j)}) \log P(z, x^{(j)} \mid \theta^{(t)})}_{\text{weighted LL completed data}} + \overset{m}{}H(Q^{(t+1)})$$

fix Q:

optimize θ

- Corresponds to weighted dataset:
  - <$\mathbf{x}^{(1)}$,**z**=1> with weight $Q^{(t+1)}(\mathbf{z}=1|\mathbf{x}^{(1)})$ .8
  - <$\mathbf{x}^{(1)}$,**z**=2> with weight $Q^{(t+1)}(\mathbf{z}=2|\mathbf{x}^{(1)})$ .15
  - <$\mathbf{x}^{(1)}$,**z**=3> with weight $Q^{(t+1)}(\mathbf{z}=3|\mathbf{x}^{(1)})$ .05
  - <$\mathbf{x}^{(2)}$,**z**=1> with weight $Q^{(t+1)}(\mathbf{z}=1|\mathbf{x}^{(2)})$ .1
  - <$\mathbf{x}^{(2)}$,**z**=2> with weight $Q^{(t+1)}(\mathbf{z}=2|\mathbf{x}^{(2)})$ .8
  - <$\mathbf{x}^{(2)}$,**z**=3> with weight $Q^{(t+1)}(\mathbf{z}=3|\mathbf{x}^{(2)})$ .1
  - …

The M-step

discovering hidden variables? → see book

$$\ell(\mathcal{D}:\theta^{(t)}) \geq \sum_{j=1}^{m}\sum_{z} Q^{(t+1)}(z \mid x^{(j)}) \log P(z, x^{(j)} \mid \theta^{(t)}) + H(Q^{(t+1)})$$

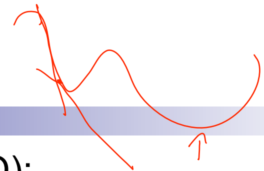
- Maximization step:

$$\theta^{(t+1)} \leftarrow \arg\max_{\theta} \sum_{j=1}^{m}\sum_{z} Q^{(t+1)}(z \mid x^{(j)}) \log P(z, x^{(j)} \mid \theta)$$

$$E_{Q^{(t+1)}}[\text{Count}(X=t, Z=f)] = \sum_{j=1}^{m} \mathbb{1}(x^{(j)}=t)\, Q^{(t+1)}(z=f \mid x^{(j)})$$

- Use expected counts instead of counts:
  - If learning requires Count(**x**,**z**)
  - Use $E_{Q(t+1)}$[Count(**x**,**z**)]

Convergence of EM



- Define potential function F(θ,Q):

$$\ell(\mathcal{D} : \theta^{(t)}) \geq F(\theta, Q) = \sum_{j=1}^{m} \sum_{z} Q(z \mid x^{(j)}) \log \frac{P(z, x^{(j)} \mid \theta)}{Q(z \mid x^{(j)})}$$

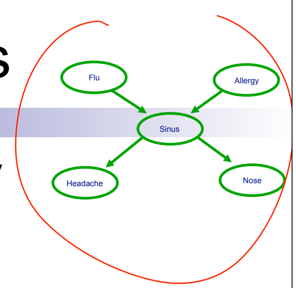
initialization can matter a lot ...

- EM corresponds to coordinate ascent on F
  - Thus, maximizes lower bound on marginal log likelihood
  - As seen in Machine Learning class last semester

fixing Q → max over θ

fixing θ → max over Q

Data likelihood for BNs



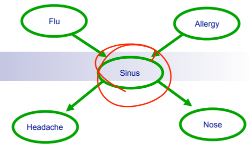
- Given structure, log likelihood of fully observed data:

$$\log P(\mathcal{D} \mid \theta_{\mathcal{G}}, \mathcal{G})$$

$$= \sum_{j=1}^{m} \log P(f^{(j)})\, P(a^{(j)})\, P(s^{(j)} \mid f^{(j)}, a^{(j)}) \ldots$$

decomposes

Marginal likelihood



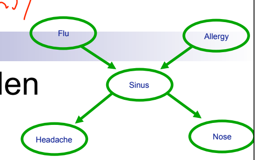
- What if S is hidden?

$$\log P(\mathcal{D} \mid \theta_{\mathcal{G}}, \mathcal{G})$$

$$= \sum_{j=1}^{m} \underbrace{\log \sum_{S} P(f^{(j)})\, P(a^{(j)})\, P(s \mid a^{(j)}, f^{(j)})\, P(h^{(j)} \mid s)\, P(n^{(j)} \mid s)}_{\text{no longer decomposes!!}}$$

Log likelihood for BNs with hidden data

$C, \{X_1, \ldots X_n\}$ $\quad H = \{X_1 \ldots X_n\} / O$



- Marginal likelihood – **O** is observed, **H** is hidden

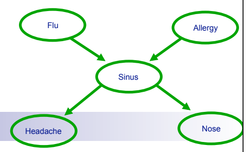
$$\ell(\theta : \mathcal{D}) = \sum_{j=1}^{m} \log P(\mathbf{o}^{(j)} \mid \theta)$$

$$= \underbrace{\sum_{j=1}^{m} \log \sum_{\mathbf{h}} P(\mathbf{h}, \mathbf{o}^{(j)} \mid \theta)}$$

What if I just want to compute this?

easy, e.g., Variable elimination

E-step for BNs



- E-step computes probability of hidden vars **h** given **o**

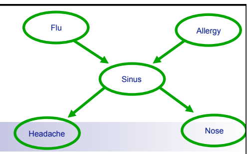
$$\underbrace{Q^{(t+1)}(\mathbf{h} \mid \mathbf{o}^{(j)})}_{\text{compute with:}} \leftarrow P(\mathbf{h} \mid \mathbf{o}^{(j)}, \theta^{(t)})$$

- Corresponds to inference in BN

Naively, must represent joint over $H \mid O^{(j)}$

if there are k hidden vars, then joint is huge!!

The M-step for BNs

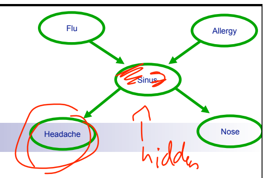


- Maximization step:

$$\theta^{(t+1)} \leftarrow \arg\max_{\theta} \sum_{j=1}^{m} \sum_{\mathbf{h}} Q^{(t+1)}(\mathbf{h} \mid \mathbf{o}^{(j)}) \log P(\mathbf{h}, \mathbf{o}^{(j)} \mid \theta)$$

- Use expected counts instead of counts:
  - If learning requires Count(**h**,**o**)
  - Use $E_{Q(t+1)}$[Count(**h**,**o**)]

M-step for each CPT



hidden

- M-step decomposes per CPT
  - Standard MLE:

$$P(X_i = x_i \mid \mathbf{Pa}_{X_i} = \mathbf{z}) = \frac{\text{Count}(X_i = x_i, \mathbf{Pa}_{X_i} = \mathbf{z})}{\text{Count}(\mathbf{Pa}_{X_i} = \mathbf{z})}$$

if var & parents hidden

$$\text{ExCount}[X_i = x_i, \mathbf{Pa}_{X_i} = \mathbf{z}] = \sum_{j=1}^{m} Q^{(t+1)}(x_i, z \mid o^{(j)})$$

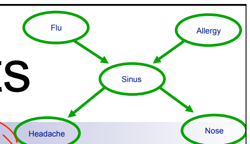
  - M-step uses expected counts:

$$P(X_i = x_i \mid \mathbf{Pa}_{X_i} = \mathbf{z}) = \frac{\text{ExCount}(X_i = x_i, \mathbf{Pa}_{X_i} = \mathbf{z})}{\text{ExCount}(\mathbf{Pa}_{X_i} = \mathbf{z})}$$

e.g., $P(H = t \mid S = f) \overset{MLE}{=} \dfrac{\text{Count}(H = t, S = f)}{\text{Count}(S = f)}$ — S is hidden

$$\text{ExCount}(H = t, S = f) = \sum_{j=1}^{m} \mathbb{1}(H^{(j)} = t)\, Q^{(t+1)}(S = f \mid h^{(j)}, f^{(j)}, a^{(j)}, n^{(j)})$$

Computing expected counts



$$P(X_i = x_i \mid \mathbf{Pa}_{X_i} = \mathbf{z}) = \frac{\text{ExCount}(\overbrace{X_i = x_i, \mathbf{Pa}_{X_i} = \mathbf{z}}^{A})}{\text{ExCount}(\mathbf{Pa}_{X_i} = \mathbf{z})}$$

requires inference for each datapoint

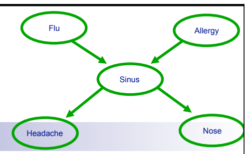
- M-step requires expected counts:
  - Observe **O=o**
  - For a set of vars **A**, must compute ExCount(**A=a**)
  - Some of **A** in example *j* will be observed
    - denote by $\mathbf{A_O} = \mathbf{a_O}^{(j)}$ — $A_O \subseteq O$
  - Some of **A** will be hidden
    - denote by $\mathbf{A_H}$ — $A_H \cap O = \emptyset$

all observed

- Use inference (E-step computes expected counts):
  - $\text{ExCount}^{(t+1)}(\mathbf{A_O} = \mathbf{a_O}, \mathbf{A_H} = \mathbf{a_H}) = \sum_{j=1}^{m} \mathbb{1}(A_O^{(j)} = a_O)\, Q^{(t+1)}(A_H = a_H \mid o^{(j)})$

if $A_O = \emptyset$, $= \sum_{j=1}^{m} Q^{(t+1)}(A = a \mid o^{(j)})$

Data need not be hidden in the same way



ugly notation

same everything

☺

- When data is fully observed
  - A data point is $\langle F=t, A=f, S=t \rangle$
- When data is partially observed $\langle F=f, A=f, S=f \rangle$
  - A data point is $\langle F=t, A=f, S=? \rangle$ $\langle F=f, A=f, S=? \rangle$
- But unobserved variables can be different for different data points
  - e.g., $\langle F=t, A=t, S=? \rangle$ $\langle F=?, A=?, S=t \rangle$ $\langle F=t, A=?, S=? \rangle$
- Same framework, just change definition of expected counts
  - Observed vars in point *j*, O_j ← changes for each j
  - Consider set of vars **A** $A_{O_j} = A \cap O_j$ $A_{H_j} = A / A_{O_j}$
  - ExCount$^{(t+1)}$(**A** = **a**)

$$= \sum_{j=1}^{m} \mathbb{1}(A_{O_j} = a_{O_j}) \, Q^{(t+1)}(A_{H_j} = a_{H_j} \mid O_j^{(j)})$$

Poster printing

facilities usually don't print on weekends

- Poster session:
  - ~~Friday~~ Monday Dec 1st, 3-6pm in the NSH Atrium.
  - There will be a popular vote for best poster. Invite your friends!
  - please be ready to set up your poster at 2:45pm sharp.
- We will provide posterboards, easels and pins.
  - The posterboards are 30x40 inches
  - We don't have a specific poster format for you to use.
    - You can either bring a big poster or a print a set of regular sized pages and pin them together.
- Unfortunately, we don't have a budget to pay for printing. If you are an SCS student, SCS has a poster printer you can use which prints on a 36" wide roll of paper.
  - If you are a student outside SCS, you will need to check with your department to see if there are printing facilities for big posters (I don't know what is offered outside SCS), or print a set of regular sized pages.
- We are looking forward to a great poster session!

EM for BNs & identifiability: a superficial discussion

- What happens if a leaf is never observed?

Y is observed but X is hidden

e.g., clustering

"Can learn"

$$\text{ExCount}(X=x, Y=y) = \sum_{j=1}^{m} \mathbb{1}(Y^{(j)}=y)\, Q(X=x \mid Y=y^{(j)})$$

Y is hidden but X is observed?

$X \to Y$

$$\text{ExCount}(X=x, Y=y) = \sum_{j=1}^{m} \mathbb{1}(X^{(j)}=x)\, Q(Y=y \mid X=x^{(j)})$$

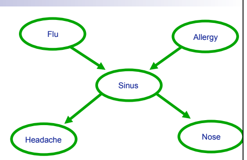
$$\ell(D:\theta) = \sum_{j=1}^{m} \log \sum_{y} P(y, x^{(j)} \mid \theta)$$

$$= \sum_{j=1}^{m} \log \underbrace{\sum_{y} P(y \mid x^{(j)}, \theta)}_{1 \text{ independently of } \theta} P(x^{(j)} \mid \theta)$$

Can't learn!!

Learning structure with missing data

[K&F 18.4]



- Known BN structure: Use expected counts, learning algorithm doesn't change
- Unknown BN structure:
  - Can use expected counts and score model as when we talked about structure learning
  - But, very slow...
    - e.g., greedy algorithm would need to redo inference for every edge we test…
- (Much Faster) **Structure-EM**: Iterate:
  - compute expected counts
  - do a some structure search (e.g., many greedy steps)
  - repeat
- **Theorem**: Converges to local optima of marginal log-likelihood
  - details in the book

What you need to know about learning BNs with missing data

- EM for Bayes Nets
- E-step: inference computes expected counts
  - Only need expected counts over X_i and $\mathbf{Pa}_{x_i}$
- M-step: expected counts used to estimate parameters
- Which variables are hidden can change per datapoint
  - Also, use labeled and unlabeled data $\rightarrow$ some data points are complete, some include hidden variables
- Structure-EM:
  - iterate between computing expected counts & many structure search steps

MNs & CRFs with missing data

- MNs with missing data
  - Models P(**X**), part of **X** hidden
  - Use EM to optimize
  - Same ideas as BN
- CRFs with missing data
  - Models P(**Y**|**X**)
  - What's hidden?
    - Part of **Y**: Y_O, Y_H, $\ell(D:\theta) = \sum_{j=1}^{m} \log \sum_{y_H} P(y_O^{(j)}, y_H \mid x^{(j)}, \theta)$, can use EM
    - All of **Y**: $\ell(D:\theta) = \sum_{j=1}^{m} \log \underbrace{\sum_{y} P(y \mid x^{(j)}, \theta)}_{1}$ $\therefore$ can't use EM
    - Part of **X**: can't use EM, because need to average over hidden X, but have no model of P(X)

Readings:
K&F: 6.1, 6.2, 6.3, 14.1, 14.2

Kalman Filters
Gaussian BNs

Graphical Models – 10708
Carlos Guestrin
Carnegie Mellon University

November 24th, 2008

Adventures of our BN hero

- Compact representation for probability distributions
- Fast inference
- Fast learning
- Approximate inference

- But… Who are the most popular kids?

1. Naïve Bayes

2 and 3.
Hidden Markov models (HMMs)
Kalman Filters

is an HMM with gaussian "CPTs"

The Kalman Filter

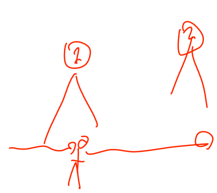
- An HMM with Gaussian distributions
- Has been around for at least 60 years
- Possibly the most used graphical model ever
- It's what
  - does your cruise control
  - tracks missiles
  - controls robots
  - …
- And it's so simple…
  - Possibly explaining why it's so used
- Many interesting models build on it…
  - An example of a Gaussian BN (more on this later)

25

Example of KF – SLAT
Simultaneous Localization and Tracking

[Funiak, Guestrin, Paskin, Sukthankar '06]

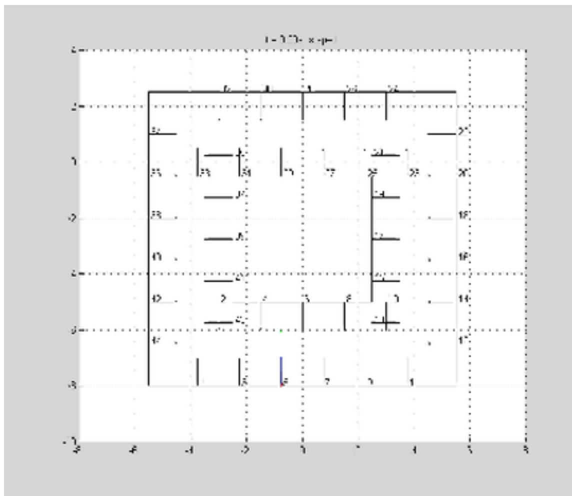
- Place some cameras around an environment, don't know where they are
- Could measure all locations, but requires lots of grad. student (Stano) time
- Intuition:
  - A person walks around
  - If camera 1 sees person, then camera 2 sees person, learn about relative positions of cameras



26

Example of KF – SLAT
Simultaneous Localization and Tracking

[Funiak, Guestrin, Paskin, Sukthankar '06]



27

Multivariate Gaussian

$$p(X_1, \ldots, X_n) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1}(\mathbf{x}-\mu)\right\}$$

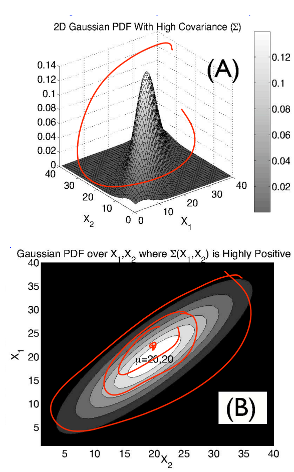
Mean vector:

$$\mu = \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_n \end{pmatrix}$$

Covariance matrix:

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_2^2 & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_3^2 \end{pmatrix}$$

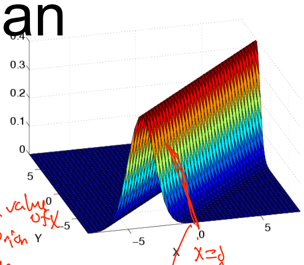
$\sigma_{32} = \sigma_{23}$

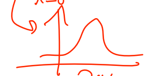


28

Conditioning a Gaussian

- Joint Gaussian:
  - p(X,Y) ~ *N*(μ;Σ)
- Conditional linear Gaussian:
  - p(Y|X) ~ *N*($\mu_{Y|X}$; $\sigma^2_{Y|X}$) ← gaussian





observed value of X

prior mean of X

$$\mu_{Y|X} = \mu_Y + \frac{\sigma_{YX}}{\sigma^2_X}(x - \mu_x)$$

↑ prior

$$\underbrace{\sigma^2_{Y|X}}_{\text{posterior variance}} = \sigma^2_Y - \underbrace{\frac{\sigma^2_{YX}}{\sigma^2_X}}_{\geq 0}$$

prior variance

← posterior variance doesn't depend on observed value!!

$\sigma^2_{Y|X} \leq \sigma^2_Y$ ($\sigma^2_{Y|X} = \sigma^2_Y$ iff $Y \perp X$)

observations always decrease variance