15-780: Grad AI
Lecture 19: Graphical models, Monte Carlo methods

Geoff Gordon (this lecture)
Tuomas Sandholm
TAs Erik Zawadzki, Abe Othman
Reminder: midterm March 29

Reminder: project milestone reports due March 31
Review: factor graphs

- Undirected, bipartite graph
  - one set of nodes represents variables
  - other set represents factors in probability distribution—tables of nonnegative numbers
  - need to compute normalizer in order to do anything useful
- Can convert back and forth to Bayes nets
- Hard v. soft constraints
Review: factor graphs

- Graphical test for independence
  - different results from Bayes net, even if we are representing the same distribution

- Inference by dynamic programming
  - instantiate evidence, eliminate nuisance nodes, normalize, answer query
    - elimination order matters
    - treewidth

- Relation to logic
Review: HMMs, DBNs

- Inference over time
  - same graphical template repeated once for each time step—conceptually infinite
- Inference: forward-backward algorithm (special case of belief propagation)
Review: numerical integration

- Integrate a difficult function over a high-dimensional volume
  - narrow, tall peaks contribute most of the integral—difficult search problem
- Central problem for approximate inference
  - e.g., computing normalizing constant in a factor graph
Uniform sampling

\[ \frac{2}{N} \sum_i f(x_i) \]
Importance sampling
Variance

- How does this help us control variance?
- Suppose \( f \) big \( \implies \) \( Q \) big
- And \( Q \) small \( \implies \) \( f \) small
- Then \( h = \frac{f}{Q} \) never gets too big
- Variance of each sample is lower \( \implies \) need fewer samples
- A good \( Q \) makes a good IS
Importance sampling, part II

Suppose

\[ f(x) = R(x)g(x) \]

\[ \int f(x) \, dx = \int R(x)g(x) \, dx \]

\[ = \mathbb{E}_R[g(x)] \]
Importance sampling, part II

- Use importance sampling w/ proposal $Q(X)$:
  - Pick $N$ samples $x_i$ from $Q(X)$
  - Average $w_i g(x_i)$, where $w_i = \frac{R(x_i)}{Q(x_i)}$ is importance weight

\[
\mathbb{E}_Q(Wg(X)) = \int Q(x) \frac{R(x)}{Q(x)} g(x) \\
= \int R(x) g(x) dx \\
= \int f(x) dx
\]
Parallel IS

- Now suppose $R(x)$ is unnormalized (e.g., represented by factor graph)—know only $Z R(x)$
- Pick $N$ samples $x_i$ from proposal $Q(X)$
- If we knew $w_i = R(x_i)/Q(x_i)$, could do IS
- Instead, set

$$\hat{w}_i = \frac{Z R(x_i)}{Q(x_i)}$$
Parallel IS

\[ \mathbb{E}(\hat{W}) = \int Q(x) \frac{Z R(x)}{Q(x)} \, dx \]

\[ = \int Z R(x) \, dx \]

\[ = Z \]

\[ \bar{w} = \frac{1}{N} \sum_i \hat{w}_i \] is an unbiased estimate of \( Z \)
Parallel IS

- So, $\hat{w}_i/\bar{w}$ is an estimate of $w_i$, computed without knowing $Z$

- Final estimate:

$$\int f(x) \, dx \approx \frac{1}{n} \sum_i \frac{\hat{w}_i}{\bar{w}} g(x_i)$$
Parallel IS is biased

\[ E(\bar{W}) = Z, \text{ but } E(1/\bar{W}) \neq 1/Z \text{ in general} \]
\[ Q : (X, Y) \sim N(1, 1) \quad \theta \sim U(-\pi, \pi) \]
\[ f(x, y, \theta) = Q(x, y, \theta)P(o = 0.8 \mid x, y, \theta)/Z \]
Posterior \( E(X, Y, \theta) = (0.496, 0.350, 0.084) \)
MCMC
Integration problem

- Recall: wanted

\[
\int f(x) \, dx = \int R(x)g(x) \, dx
\]

- And therefore, wanted good importance distribution \( Q(x) \) (close to \( R \))
Back to high dimensions

- Picking a good importance distribution is hard in high-D
- Major contributions to integral can be hidden in small areas
  - recall, want (R big ==> Q big)
- Would like to search for areas of high R(x)
- But searching could bias our estimates
Markov-Chain Monte Carlo

- Design a randomized search procedure $M$ over values of $x$, which tends to increase $R(x)$ if it is small.
- Run $M$ for a while, take resulting $x$ as a sample.
- Importance distribution $Q(x)$?
Markov-Chain Monte Carlo

- Design a randomized search procedure \( M \) over values of \( x \), which tends to increase \( R(x) \) if it is small
- Run \( M \) for a while, take resulting \( x \) as a sample
- Importance distribution \( Q(x) \)?
  - \( Q = \) stationary distribution of \( M \)…
Stationary distribution

- Run HMM or DBN for a long time; stop at a random point
- Do this again and again
- Resulting samples are from stationary distribution
Designing a search chain

\[ \int f(x) \, dx = \int R(x) g(x) \, dx \]

- Would like \( Q(x) = R(x) \)
  - makes importance weight = 1
- Turns out we can get this exactly, using \textbf{Metropolis-Hastings}
Metropolis-Hastings

- Way of designing chain w/ \( Q(x) = R(x) \)
- Basic strategy: start from arbitrary \( x \)
- Repeatedly tweak \( x \) to get \( x' \)
- If \( R(x') \geq R(x) \), move to \( x' \)
- If \( R(x') \ll R(x) \), stay at \( x \)
- In intermediate cases, randomize
Proposal distribution

- Left open: what does “tweak” mean?
- Parameter of MH: $Q(x' | x)$
  - one-step proposal distribution
- Good proposals explore quickly, but remain in regions of high $R(x)$
- Optimal proposal?
MH algorithm

- Sample $x' \sim Q(x' \mid x)$
- Compute $p = \frac{R(x')}{R(x)} \frac{Q(x' \mid x)}{Q(x \mid x')}$
- With probability $\min(1, p)$, set $x := x'$
- Repeat for $T$ steps; sample is $x_1, \ldots, x_T$ (will usually contain duplicates)
MH algorithm

- Sample $x' \sim Q(x' \mid x)$
- Compute $p = \frac{R(x') \cdot Q(x' \mid x)}{R(x) \cdot Q(x \mid x')}$
- With probability $\min(1, p)$, set $x := x'$
- Repeat for $T$ steps; sample is $x_1, \ldots, x_T$ (will usually contain duplicates)

Note: we don’t need to know $Z$
MH example
Acceptance rate

- Moving to new $x'$ is accepting
- Want acceptance rate (avg $p$) to be large, so we don’t get big runs of the same $x$
- Want $Q(x' \mid x)$ to move long distances (to explore quickly)
- Tension between $Q$ and $P(\text{accept})$:

$$p = \frac{R(x')}{R(x)} \frac{Q(x' \mid x)}{Q(x \mid x')}$$
Mixing rate, mixing time

- If we pick a good proposal, we will move rapidly around domain of $R(x)$
- After a short time, won’t be able to tell where we started
- This is short **mixing time** = # steps until we can’t tell which starting point we used
- **Mixing rate** = $1 / (\text{mixing time})$
MH estimate

- Once we have our samples $x_1, x_2, \ldots$
- Optional: discard initial “burn-in” range
  - allows time to reach stationary dist’n
- Estimated integral:
  $$\frac{1}{N} \sum_{i=1}^{N} g(x_i)$$
In example

- $g(x) = x^2$
- True $E(g(X)) = 0.28…$
- Proposal: $Q(x' | x) = N(x' | x, 0.25^2 I)$
- Acceptance rate 55–60%
- After 1000 samples, minus burn-in of 100:
  - final estimate 0.282361
  - final estimate 0.271167
  - final estimate 0.322270
  - final estimate 0.306541
  - final estimate 0.308716
Gibbs sampler

- Special case of MH
- Divide $\mathbf{X}$ into blocks of r.v.s $B(1), B(2), \ldots$
- Proposal $Q$:
  - pick a block $i$ uniformly (or round robin, or any other schedule)
  - sample $\mathbf{X}_{B(i)} \sim P(\mathbf{X}_{B(i)} | \mathbf{X}_{\neg B(i)})$
Gibbs example
Gibbs example
Why is Gibbs useful?

- For Gibbs, $p =$

$$
\frac{P(x'_i, x'_{-i})}{P(x_i, x_{-i})} \frac{P(x_i \mid x'_{-i})}{P(x'_i \mid x_{-i})}
$$
Gibbs derivation

\[
\frac{P(x'_i, x'_{-i})}{P(x_i, x_{-i})} \frac{P(x_i \mid x'_{-i})}{P(x'_i \mid x_{-i})} = \frac{P(x'_i, x_{-i})}{P(x_i, x_{-i})} \frac{P(x_i \mid x_{-i})}{P(x'_i \mid x_{-i})}
\]

\[
= \frac{P(x'_i, x_{-i})}{P(x_i, x_{-i})} \frac{P(x_i, x_{-i})}{P(x'_i, x_{-i})} = 1
\]
Gibbs in practice

- Proof of $p=1$ means Gibbs is often easy to implement
- Often works well
  - if we choose good blocks (but there may be no good blocking!)
- Fancier version: adaptive blocks, based on current $\mathbf{x}$
Gibbs failure example
Sequential sampling

In an HMM or DBN, to sample $P(\mathbf{X}_T)$, start from $\mathbf{X}_1$ and sample forward step by step

$\mathbf{X}_{t+1} \sim P(\mathbf{X}_{t+1} | \mathbf{X}_t)$

$P(\mathbf{X}_{1:T}) = P(\mathbf{X}_1) P(\mathbf{X}_2 | \mathbf{X}_1) P(\mathbf{X}_3 | \mathbf{X}_2) \ldots$
Particle filter

- Can sample $\mathbf{X}_{t+1} \sim P(\mathbf{X}_{t+1} \mid \mathbf{X}_t)$ using any algorithm from above.
- If we use parallel importance sampling to get $N$ samples at once from each $P(\mathbf{X}_t)$, we get a **particle filter**
  - also need one more trick: **resampling**
- Write $\mathbf{x}_{t,i}$ ($i = 1 \ldots N$) for sample at time $t$
Particle filter

- Want one sample from each of $P(X_{t+1} | x_{t,i})$
- Have only $Z P(X_{t+1} | x_{t,i})$
- For each $i$, pick $x_{t+1,i}$ from proposal $Q(x)$
- Compute unnormalized importance weight

$$
\hat{\omega}_i = Z P(x_{t+1,i} | x_{t,i}) / Q(x_{t+1,i})
$$
Particle filter

- Normalize weights:

\[ \bar{w} = \frac{1}{N} \sum_{i} \hat{w}_i \quad w_i = \frac{\hat{w}_i}{\bar{w}} \]

- Now, \((w_i, x_{t+1,i})\) is an approximate weighted sample from \(P(X_{t+1})\)

- To get an unweighted sample, resample
Resampling

- Sample $N$ times (with replacement) from $x_{t+1,i}$ with probabilities $w_i/N$
  - alternately: deterministically take $\text{floor}(w_i)$ copies of $x_{t+1,i}$ and sample only from fractional part $[w_i - \text{floor}(w_i)]$
- Each $x_{t+1,i}$ appears $w_i$ times on average, so we’re still a sample from $P(X_{t+1})$
Particle filter example
Learning
Learning

○ Basic learning problem: given some experience, find a new or improved model

○ Experience: a sample $x_1, \ldots, x_N$

○ Model: want to predict $x_{N+1}, \ldots$
Example

- **Experience** = range sensor readings & odometry from robot
- **Model** = map of the world
Example

- **Experience** = physical measurements of surveyed specimens & expert judgements of their true species

- **Model** = factor graph relating species to measurements
### Sample data

<table>
<thead>
<tr>
<th></th>
<th>sepal length</th>
<th>sepal width</th>
<th>petal length</th>
<th>petal width</th>
<th>species</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>3.5</td>
<td>1.4</td>
<td>0.2</td>
<td>Iris setosa</td>
<td></td>
</tr>
<tr>
<td>5.6</td>
<td>3.0</td>
<td>4.5</td>
<td>1.5</td>
<td>Iris versicolor</td>
<td></td>
</tr>
<tr>
<td>4.9</td>
<td>3.0</td>
<td>1.4</td>
<td>0.2</td>
<td>Iris setosa</td>
<td></td>
</tr>
<tr>
<td>6.4</td>
<td>2.8</td>
<td>5.6</td>
<td>2.1</td>
<td>Iris virginica</td>
<td></td>
</tr>
<tr>
<td>5.8</td>
<td>2.7</td>
<td>4.1</td>
<td>1.0</td>
<td>Iris versicolor</td>
<td></td>
</tr>
</tbody>
</table>
Factor graph

- One of many possible factor graphs
- Values of Φs not shown, but part of model
Factor graph
Factor graph

\[ \Phi \]

\[ \Phi_1 \text{ params} \]
In general

- For our purposes, a model $M$ is exactly a distribution $P(X \mid M)$ over possible samples.

- When is $M$ better than $M'$? When $P(X \mid M)$ is more accurate than $P(X \mid M')$.

- Bayes rule encodes this: from prior $P(M)$ and evidence $X$, compute posterior $P(M \mid X)$.
  - $P(M \mid X) = \frac{P(X \mid M) \cdot P(M)}{P(X)}$
  - Better predictions (higher $P(X \mid M)$) yield higher posterior.
Conditional model

- Split variables into \((X, Y)\)
- Suppose we always observe \(X\)
- Two ways \(P(X, Y)\) and \(P'(X, Y)\) can differ:
  - \(P(X) \neq P'(X)\), and/or
  - \(P(Y \mid X) \neq P'(Y \mid X)\)
- First way doesn’t matter for decisions
- **Conditional model**: only specifies \(P(Y \mid X, M)\)
Conditional model example

- Experience = samples of \((X, Y)\)
- \(X\) = features of object
- \(Y\) = whether object is a “framling”
- Model = rule for deciding whether a new object is a framling
Sample data & possible model

<table>
<thead>
<tr>
<th>tall</th>
<th>pointy</th>
<th>blue</th>
<th>framling</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

\[ H = \text{tall} \land \lnot \text{blue} \]
Hypothesis space

- Hypothesis space $\mathcal{H} =$ set of models we are willing to consider
  - for philosophical or computational reasons
- E.g., all factor graphs of a given structure
- Or, all conjunctions of up to two literals
- Prior is a distribution over $\mathcal{H}$
A simple learning algorithm

- Conditional learning: samples \((x_i, y_i)\)
- Let \(\mathcal{H}\) be a set of propositional formulae
  - \(\mathcal{H} = \{ H_1, H_2, \ldots \}\)
- \(H\) is **consistent** if \(H(x_i) = y_i\) for all \(i\)
- **Version space** \(V = \{ \text{all consistent } H \} \subseteq \mathcal{H}\)
- **Version space algorithm**: predict \(y = \text{majority vote of } H(x)\) over all \(H \in V\)
Framlings

- $H = \{ \text{conjunctions of up to 2 literals} \} = \{ T, F, \text{tall, pointy, blue, } \neg \text{tall, } \neg \text{pointy, } \neg \text{blue, tall } \land \text{ pointy, tall } \land \text{ blue, pointy } \land \text{ blue, } \neg \text{tall } \land \text{ pointy, ... } \}$
Framlings

<table>
<thead>
<tr>
<th>tall</th>
<th>pointy</th>
<th>blue</th>
<th>framling</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>
Analysis

- **Mistake** = make wrong prediction

- If some $H \in \mathcal{H}$ is always right, eventually we’ll eliminate all competitors, and make no more mistakes.

- If no $H \in \mathcal{H}$ is always right, eventually $V$ will become empty.

- e.g., if *label noise* or *feature noise*
Analysis

- Suppose $| \mathcal{H} | = N$
- How many mistakes could we make?
Suppose $|\mathcal{H}| = N$

How many mistakes could we make?

Since we predict with majority of $V$, after any mistake, we eliminate half (or more) of $V$

Can’t do that more than $\log_2(N)$ times
Discussion

- In example, $N = 20$, $\log_2(N) = 4.32$
- Made only 2 mistakes
- Mistake bound: limits wrong decisions, as desired
- But, required strong assumptions (no noise, true $H$ contained in $H$)
- Could be very slow!
Bayesian Learning
Recall iris example

- $\mathcal{H} =$ factor graphs of given structure
- Need to specify entries of $\phi$s
## Factors

<table>
<thead>
<tr>
<th></th>
<th>( \phi_0 )</th>
<th>( \phi_{1-4} )</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>setosa</strong></td>
<td>( p )</td>
<td>( p_i )</td>
</tr>
<tr>
<td><strong>versicolor</strong></td>
<td>( q )</td>
<td>( q_i )</td>
</tr>
<tr>
<td><strong>virginica</strong></td>
<td>( 1-p-q )</td>
<td>( r_i )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( s_i )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( u_i )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( v_i )</td>
</tr>
</tbody>
</table>
### Continuous factors

<table>
<thead>
<tr>
<th></th>
<th>lo</th>
<th>m</th>
<th>hi</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>set.</strong></td>
<td>p₁</td>
<td>q₁</td>
<td>l–p₁–q₁</td>
</tr>
<tr>
<td><strong>vers.</strong></td>
<td>r₁</td>
<td>s₁</td>
<td>l–r₁–s₁</td>
</tr>
<tr>
<td><strong>vir.</strong></td>
<td>u₁</td>
<td>v₁</td>
<td>l–u₁–v₁</td>
</tr>
</tbody>
</table>

\[
\Phi_1(\ell, s) = \exp\left(-\frac{(\ell - \ell_s)^2}{2\sigma^2}\right)
\]

parameters \(\ell_{set}, \ell_{vers}, \ell_{vir}\); constant \(\sigma^2\)

Discretized petal length  Continuous petal length
Simpler example

<table>
<thead>
<tr>
<th></th>
<th>$p$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>H</strong></td>
<td></td>
</tr>
<tr>
<td><strong>T</strong></td>
<td>$1-p$</td>
</tr>
</tbody>
</table>

Coin toss
Parametric model class

- $\mathcal{H}$ is a **parametric** model class: each $H$ in $\mathcal{H}$ corresponds to a vector of parameters $\theta = (p)$ or $\theta = (p, q, p_1, q_1, r_1, s_1, \ldots)$

- $H_\theta: \mathbf{X} \sim P(\mathbf{X} \mid \theta)$ (or, $Y \sim P(Y \mid \mathbf{X}, \theta)$)

- Contrast to **discrete** $\mathcal{H}$, as in version space

- Could also have **mixed** $\mathcal{H}$: discrete choice among parametric (sub)classes
Continuous prior

- E.g., for coin toss, $p \sim \text{Beta}(a, b)$:

$$P(p \mid a, b) = \frac{1}{B(a, b)} p^{a-1} (1 - p)^{b-1}$$

- Specifying, e.g., $a = 2, b = 2$:

$$P(p) = 6p(1 - p)$$
Prior for $p$
Joint dist’n of parameter $p$ and data $x_i$:

$$P(p, x) = P(p) \prod_i P(x_i \mid p)$$

$$= 6p(1 - p) \prod_i p^{x_i} (1 - p)^{1-x_i}$$
Coin flip posterior

\[ P(p \mid x) = P(p) \prod_{i} P(x_i \mid p)/P(x) \]

\[ = \frac{1}{Z} p(1 - p) \prod_{i} p^{x_i} (1 - p)^{1-x_i} \]

\[ = \frac{1}{Z} p^{1+\sum_i x_i} (1 - p)^{1+\sum_i (1-x_i)} \]

\[ = \text{Beta}(2 + \sum_i x_i, 2 + \sum_i (1 - x_i)) \]
Prior for $p$
Posterior after 4 H, 7 T
Posterior after 10 H, 19 T
Predictive distribution

- Posterior is nice, but doesn’t tell us directly what we need to know
- We care more about $P(x_{N+1} \mid x_1, \ldots, x_N)$
- By law of total probability, conditional independence:

$$P(x_{N+1} \mid D) = \int P(x_{N+1}, \theta \mid D) d\theta$$

$$= \int P(x_{N+1} \mid \theta) P(\theta \mid D) d\theta$$
Coin flip example

- After 10 H, 19 T: $p \sim \text{Beta}(12, 21)$
- $E(x_{N+1} | p) = p$
- $E(x_{N+1} | \theta) = E(p | \theta) = \frac{a}{a+b} = \frac{12}{33}$
- So, predict 36.4% chance of H on next flip
Approximate Bayes
Approximate Bayes

- Coin flip example was easy
- In general, computing posterior (or predictive distribution) may be hard
- Solution: use the approximate integration techniques we’ve studied!
Bayes as numerical integration

- Parameters $\theta$, data $D$
- $P(\theta \mid D) = P(D \mid \theta) P(\theta) / P(D)$
- Usually, $P(\theta)$ is simple; so is $Z \ P(D \mid \theta)$
- So, $P(\theta \mid D) \propto Z \ P(D \mid \theta) P(\theta)$
- Perfect for MH
\[ P(y \mid x) = \sigma(ax + b) \]

\[ \sigma(z) = \frac{1}{1 + \exp(-z)} \]
Posterior

\[ P(a, b \mid x_i, y_i) = \]
\[ Z P(a, b) \prod_i \sigma(ax_i + b)^{y_i} \sigma(-ax_i - b)^{1-y_i} \]

\[ P(a, b) = N(0, I) \]
Sample from posterior
Expanded factor graph

original factor graph:
Cheaper approximations
Getting cheaper

- Maximum a posteriori (MAP)
- Maximum likelihood (MLE)
- Conditional MLE / MAP

- Instead of true posterior, just use single most probable hypothesis
MAP

\[ \arg \max_{\theta} P(D | \theta) P(\theta) \]

- Summarize entire posterior density using the maximum
MLE

\[ \arg \max_{\theta} P(D | \theta) \]

- Like MAP, but ignore prior term
Conditional MLE, MAP

\[ \arg \max_{\theta} P(y \mid x, \theta) \]

\[ \arg \max_{\theta} P(y \mid x, \theta)P(\theta) \]

- Split \( D = (x, y) \)
- Condition on \( x \), try to explain only \( y \)
Iris example: MAP vs. posterior
Irises: MAP vs. posterior
Too certain

- This behavior of MAP (or MLE) is typical: we are too sure of ourselves
- But, often gets better with more data
- Theorem: MAP and MLE are consistent estimates of true $\theta$, if “data per parameter” $\to \infty$