Probabilistic Graphical Models

Inference

Sampling Methods

Simple Sampling
Sampling-Based Estimation

\[ \mathcal{D} = \{x[1], \ldots, x[M]\} \]

If \( P(X=1) = p \)

Estimator for \( p \): \( T_D = \frac{1}{M} \sum_{m=1}^{M} x[m] \)

More generally, for any distribution \( P \), function \( f \):

\[ E_P[f] \approx \frac{1}{M} \sum_{m=1}^{M} f(x[m]) \]
Sampling from Discrete Distribution

$$\text{Val}(X) = \{x^1, \ldots, x^k\} \quad P(x^i) = \theta^i$$
Sampling-Based Estimation

Hoeffding Bound:

\[ P_D(T_D \notin [p - \epsilon, p + \epsilon]) \leq 2e^{-2M\epsilon^2} \]

Chernoff Bound:

\[ P_D(T_D \notin [p(1 - \epsilon), p(1 + \epsilon)]) \leq 2e^{-Mp\epsilon^2/3} \]
Sampling-Based Estimation

\[ T_D = \frac{1}{M} \sum_{m=1}^{M} X[m] \]

**Hoeffding Bound:**

\[ P_D(T_D \notin [p - \epsilon, p + \epsilon]) \leq 2e^{-2M\epsilon^2} \]

For additive bound \( \epsilon \) on error with probability > 1-\( \delta \):

\[ M \geq \frac{\ln(2/\delta)}{2\epsilon^2} \]

**Chernoff Bound:**

\[ P_D(T_D \notin [p(1 - \epsilon), p(1 + \epsilon)]) \leq 2e^{-Mp\epsilon^2/3} \]

For multiplicative bound \( \epsilon \) on error with probability > 1-\( \delta \):

\[ M \geq 3 \frac{\ln(2/\delta)}{p\epsilon^2} \]

Daphne Koller
Forward sampling from a BN

$P(D, I, G, S, L)$

<table>
<thead>
<tr>
<th>$d^0$</th>
<th>$d^1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6</td>
<td>0.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$i^0$</th>
<th>$i^1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.7</td>
<td>0.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g^1$</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>$i^0, d^0$</td>
</tr>
<tr>
<td>$i^0, d^1$</td>
</tr>
<tr>
<td>$i^1, d^0$</td>
</tr>
<tr>
<td>$i^1, d^1$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Letter</th>
</tr>
</thead>
<tbody>
<tr>
<td>$l^0$</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>$g^1$</td>
</tr>
<tr>
<td>$g^2$</td>
</tr>
<tr>
<td>$g^3$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s^0$</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>$i^0$</td>
</tr>
<tr>
<td>$i^1$</td>
</tr>
</tbody>
</table>

Topological order: parents before children

Daphne Koller
Forward Sampling for Querying

- **Goal:** Estimate $P(Y=y)$
  - Generate samples from BN
  - Compute fraction where $Y=y$

For additive bound $\epsilon$ on error with probability $> 1-\delta$: 
$$M \geq \frac{\ln(2/\delta)}{2\epsilon^2}$$

For multiplicative bound $\epsilon$ on error with probability $> 1-\delta$: 
$$M \geq 3 \frac{\ln(2/\delta)}{P(y)\epsilon^2}$$
Queries with Evidence

- **Goal:** Estimate $P(Y=y \mid E=e)$
- **Rejection sampling algorithm**
  - Generate samples from BN
  - Throw away all those where $E \neq e$
  - Compute fraction where $Y = y$

Expected fraction of samples kept $\sim P(e)$

# samples needed grows exponentially with # of observed variables

Daphne Koller
Summary

• Generating samples from a BN is easy
• \((\varepsilon, \delta)\)-bounds exist, but usefulness is limited:
  – Additive bounds: useless for low probability events
  – Multiplicative bounds: \# samples grows as \(1/P(y)\)
• With evidence, \# of required samples grows exponentially with \# of observed variables

• Forward sampling generally infeasible for MNs
Probabilistic Graphical Models

Inference
Sampling Methods

Markov Chain Monte Carlo
A Markov chain defines a probabilistic transition model $T(x \rightarrow x')$ over states $x$:

- for all $x$: \[ \sum_{x'} T(x \rightarrow x') = 1 \]
Temporal Dynamics

\[ P^{(t+1)}(X^{(t+1)} = x') = \sum_{x} P^{(t)}(X^{(t)} = x)T(x \rightarrow x') \]

<table>
<thead>
<tr>
<th></th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>+1</th>
<th>+2</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P^{(0)} )</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( P^{(1)} )</td>
<td>0</td>
<td>.25</td>
<td>.5</td>
<td>.25</td>
<td>0</td>
</tr>
<tr>
<td>( P^{(2)} )</td>
<td>.25^2=.0625</td>
<td>2 \times (.5 \times .25) = .25</td>
<td>5^2 + 2 \times .25^2 = .375</td>
<td>2 \times (.5 \times .25) = .25</td>
<td>.25^2=.0625</td>
</tr>
</tbody>
</table>

Daphne Koller
Stationary Distribution

\[ P^{(t)}(x') \approx P^{(t+1)}(x') = \sum_x P^{(t)}(x)T(x \rightarrow x') \]

\[ \pi(x') = \sum_x \pi(x)T(x \rightarrow x') \]

\[
\begin{align*}
\pi(x^1) &= 0.25\pi(x^1) + 0.5\pi(x^3) & \pi(x^1) &= 0.2 \\
\pi(x^2) &= 0.7\pi(x^2) + 0.5\pi(x^3) & \pi(x^2) &= 0.5 \\
\pi(x^3) &= 0.75\pi(x^1) + 0.3\pi(x^2) & \pi(x^3) &= 0.3 \\
\pi(x^1) + \pi(x^2) + \pi(x^3) &= 1
\end{align*}
\]
Regular Markov Chains

• A Markov chain is regular if there exists $k$ such that, for every $x, x'$, the probability of getting from $x$ to $x'$ in exactly $k$ steps is $> 0$.

• Theorem: A regular Markov chain converges to a unique stationary distribution regardless of start state.
Regular Markov Chains

• A Markov chain is regular if there exists $k$ such that, for every $x, x'$, the probability of getting from $x$ to $x'$ in exactly $k$ steps is $> 0$

- $k$ is a distance between furthest $x, x'$

• Sufficient conditions for regularity:
  - Every two states are connected with path of $p(x) > 0$
  - For every state, there is a self-transition
Probabilistic Graphical Models

Inference
Sampling Methods

Using a Markov Chain
Using a Markov Chain

• **Goal:** compute $\mathbb{P}(x \in S)$ – but $P$ is too hard to sample from directly

• Construct a Markov chain $T$ whose unique stationary distribution is $P$

• Sample $x^{(0)}$ from some $P^{(0)}$

• For $t = 0, 1, 2, \ldots$
  – Generate $x^{(t+1)}$ from $T(x^{(t)} \to x')$
Using a Markov Chain

• We only want to use samples that are sampled from a distribution close to $P$.

• At early iterations, $P(t)$ is usually far from $P$.

• Start collecting samples only after the chain has run long enough to “mix” $P(t)$ close enough to $P$. 

Daphne Koller
Mixing

• How do you know if a chain has mixed or not?
  – In general, you can never “prove” a chain has mixed
  – But in many cases you can show that it has NOT

• How do you know a chain has not mixed?
  – Compare chain statistics in different windows within a single run of the chain
  – and across different runs initialized differently

Daphne Koller
Initialized from an arbitrary state

Initialized from a high-probability state

Mixing?

Maybe

NO

Daphne Koller
- Each dot is a statistic (e.g., \( P(x \in S) \))
- x-position is its estimated value from chain 1
- y-position is its estimated value from chain 2

Mixing?

NO

Maybe
Using the Samples

• Once the chain mixes, all samples \( x^{(t)} \) are from the stationary distribution \( \pi \)
  – So we can (and should) use all \( x^{(t)} \) for \( t > T_{\text{mix}} \)

• However, nearby samples are correlated!
  – So we shouldn’t overestimate the quality of our estimate by simply counting samples \( \not\text{iid} \)

• The faster a chain mixes, the less correlated (more useful) the samples
MCMC Algorithm Summary I

- For c=1,...,C
  - Sample $x^{(c,0)}$ from $P^{(0)}$

- Repeat until mixing
  - For c=1,...,C
    - Generate $x^{(c,t+1)}$ from $T(x^{(c,t)} \rightarrow x')$
    - Compare window statistics in different chains to determine mixing
  - $t := t+1$
MCMC Algorithm Summary II

• Repeat until sufficient samples
  – D := ∅
  – For c=1,…,C
    • Generate \(x^{(c, t+1)}\) from \(T(x^{(c, t)} \rightarrow x')\)
    • D := D ∪ \{x^{(c, t+1)}\}
  – t := t+1
• Let D = \{x[1],...,x[M]\}
• Estimate
  \[ E_P[f] \approx \frac{1}{M} \sum_{m=1}^{M} f(x[m]) \]
Summary

• Pros:
  – Very general purpose
  – Often easy to implement
  – Good theoretical guarantees as $t \to \infty$

• Cons:
  – Lots of tunable parameters / design choices
  – Can be quite slow to converge
  – Difficult to tell whether it’s working
MCMC for PGMs: The Gibbs Chain
Gibbs Chain

- Target distribution $P_{\Phi}(X_1, \ldots, X_n)$
- Markov chain state space: complete assignments $x$ to $X = \{X_1, \ldots, X_n\}$
- Transition model given starting state $x$:
  - For $i = 1, \ldots, n$
    - Sample $x_i \sim P_{\Phi}(X_i \mid x_{-i})$
  - Set $x' = x$
Example

\[
P(d | i^0, g^1, l^0, s^0)
\]

\[
p(g | d^0, l^0, s^0)
\]

\[
P(i | g^1, g^2, s^0)
\]

\[
\begin{array}{c|c|c}
  & d^0 & d^1 \\
\hline
i^0 & 0.6 & 0.4 \\
i^1 & 0.7 & 0.3 \\
\end{array}
\]

\[
\begin{array}{c|c|c|c|c}
  & g^1 & g^2 & g^3 \\
\hline
i^0, d^0 & 0.3 & 0.4 & 0.3 \\
i^0, d^1 & 0.05 & 0.25 & 0.7 \\
i^1, d^0 & 0.9 & 0.08 & 0.02 \\
i^1, d^1 & 0.5 & 0.3 & 0.2 \\
\end{array}
\]

\[
\begin{array}{c|c|c}
  & s^0 & s^1 \\
\hline
i^0 & 0.95 & 0.05 \\
i^1 & 0.2 & 0.8 \\
\end{array}
\]

Daphne Koller
Computational Cost

• For i=1,...,n
  – Sample $x_i \sim P_\Phi(X_i \mid x_{-i})$

$$P_\Phi(X_i \mid x_{-i}) = \frac{P_\Phi(X_i, x_{-i})}{P_\Phi(x_{-i})} = \frac{\tilde{P}_\Phi(X_i, x_{-i})}{\tilde{P}_\Phi(x_{-i})}$$

Daphne Koller
Another Example

\[ P_\Phi(A \mid b, c, d) = \frac{\tilde{P}_\Phi(a, b, c, d)}{\sum_{A'} \tilde{P}_\Phi(A', b, c, d)} \]

Normalizing constant 

Factors that involve A
Computational Cost Revisited

• For $i=1,...,n$
  - Sample $x_i \sim P_{\Phi}(X_i \mid x_{-i})$

\[
P_{\Phi}(X_i \mid x_{-i}) = \frac{P_{\Phi}(X_i, x_{-i})}{P_{\Phi}(x_{-i})} = \frac{\tilde{P}_{\Phi}(X_i, x_{-i})}{\tilde{P}_{\Phi}(x_{-i})}
\]

only $x_i$ and its neighbors

\[
\propto \prod_{j : X_i \in \text{Scope}[C_j]} \phi_j(X_i, x_j, -i)
\]
Gibbs Chain and Regularity

• If all factors are positive, Gibbs chain is regular
• However, mixing can still be very slow

<table>
<thead>
<tr>
<th>X₁</th>
<th>X₂</th>
<th>Y</th>
<th>Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.25</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.25</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0.25</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.25</td>
</tr>
</tbody>
</table>
Summary

- Converts the hard problem of inference to a sequence of “easy” sampling steps

- Pros:
  - Probably the simplest Markov chain for PGMs
  - Computationally efficient to sample

- Cons:
  - Often slow to mix, esp. when probabilities are peaked
  - Only applies if we can sample from product of factors
Probabilistic Graphical Models

Inference
Sampling Methods

Metropolis-Hastings Algorithm
Reversible Chains

Theorem: If detailed balance holds, and $T$ is regular, then $T$ has a unique stationary distribution $\pi$.

Proof:

$$\sum_x \pi(x)T(x \rightarrow x') = \sum_x \pi(x')T(x' \rightarrow x) = \pi(x) \cdot \sum T(x' \rightarrow x)$$

$$\sum_x \pi(x)T(x \rightarrow x') = \pi(x')$$

definition of $T$
Metropolis Hastings Chain

Proposal distribution \( Q(x \rightarrow x') \)

Acceptance probability: \( A(x \rightarrow x') \)

- At each state \( x \), sample \( x' \) from \( Q(x \rightarrow x') \)
- Accept proposal with probability \( A(x \rightarrow x') \)
  - If proposal accepted, move to \( x' \)
  - Otherwise stay at \( x \)

\[
T(x \rightarrow x') = Q(x \rightarrow x') A(x \rightarrow x') \quad \text{if } x' \neq x
\]

\[
T(x \rightarrow x) = Q(x \rightarrow x) + \sum_{x' \neq x} Q(x \rightarrow x') (1-A(x \rightarrow x'))
\]
Acceptance Probability

\[
\pi(x)T(x \rightarrow x') = \pi(x')T(x' \rightarrow x)
\]

Construct a s.t. \( \pi(x)Q(x \rightarrow x')A(x \rightarrow x') = \pi(x')Q(x' \rightarrow x)A(x' \rightarrow x) \)

\[
A(x \rightarrow x') = \begin{cases} p & \text{if } A(x' \rightarrow x) = 1 \\ \frac{\pi(x')Q(x' \rightarrow x)}{\pi(x)Q(x \rightarrow x')} & \end{cases} = p < 1
\]

\[
A(x \rightarrow x') = \min \left[ 1, \frac{\pi(x')Q(x' \rightarrow x)}{\pi(x)Q(x \rightarrow x')} \right]
\]
Choice of $Q$

$$A(x \rightarrow x') = \min \left[ 1, \frac{\pi(x')Q(x' \rightarrow x)}{\pi(x)Q(x \rightarrow x')} \right]$$

• $Q$ must be reversible:
  - $Q(x \rightarrow x') > 0 \Rightarrow Q(x' \rightarrow x) > 0$

• Opposing forces
  - $Q$ should try to spread out, to improve mixing
  - But then acceptance probability often low
MCMC for Matching

$X_i = j$ if $i$ matched to $j$

$P(X_1 = v_1, \ldots, X_4 = v_4) \propto \begin{cases} 
\exp \left( - \sum_i \text{dist} (i, v_i) \right) & \text{if every } X_i \text{ has different value} \\
0 & \text{otherwise}
\end{cases}$
MH for Matching: Augmenting Path

1) randomly pick one variable $X_i$
2) sample $X_i$, pretending that all values are available
3) pick the variable whose assignment was taken (conflict), and return to step 2
• When step 2 creates no conflict, modify assignment to flip augmenting path
Example Results

Gibbs

MH proposal 1

MH proposal 2
Summary

• MH is a general framework for building Markov chains with a particular stationary distribution
  – Requires a proposal distribution
  – Acceptance computed via detailed balance

• Tremendous flexibility in designing proposal distributions that explore the space quickly
  – But proposal distribution makes a big difference
  – and finding a good one is not always easy