A Quiz Question

Consider an exponential family on $x_1, x_2 \in \{0, 1\}$ with $T(x_1, x_2) = I[x_1 = 1, x_2 = 1]$.
Suppose you use the data below to estimate maximum likelihood parameters:

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

At the maximum likelihood estimate $\theta^*$, what will be $P_{\theta^*}(X_1 = 1, X_2 = 1)$?
Message Passing by Autodiff

To be concrete, let’s consider a pairwise MRF:

\[ p_{\theta}(x) = \frac{1}{Z(\theta)} \prod_{(i,j) \in E} \exp(\theta_{ij}^a \cdot I[x_i = a, x_j = b]), \quad \theta = (\theta_{ij}^a)_{(i,j) \in E, a \in \text{Val}(x_i), b \in \text{Val}(x_j)} \]

Fact (from previous lectures): \[ \frac{\partial}{\partial \theta_{ij}^a} \log Z(\theta) = P_{\theta}(X_i = a, X_j = b) \]

Algorithm

1. Compute \( Z(\theta) \) by variable elimination (forward pass)
2. Compute \( \mu = \nabla_{\theta} \log Z(\theta) \) by autodiff
3. Recover marginals: \( P_{\theta}(X_i = a, X_j = b) = \mu_{ij} \)

Message Passing by Autodiff: Observations

- The total running-time is \( \sim 2x \) the cost of the forward pass (variable elimination), which is the same as message passing. The backward pass of autodiff is equivalent to the distribute phase of message-passing.
- It still works for general MRFs

\[ p_{\theta}(x) = \frac{1}{Z(\theta)} \prod_{c \in C} \exp(\theta_{c}^a \cdot I[x_c = a]), \quad \theta = (\theta_{c}^a)_{c \in C, a \in \text{Val}(x_c)} \]

- Compute \( \mu = \nabla_{\theta} \log Z(\theta) \) by variable elimination and autodiff, then recover marginals as \( P_{\theta}(X_c = a) = \mu_c^a \). (No junction trees here!)
Chain Example

Let's use a chain on \(x_1, \ldots, x_n\) to see that the backward autodiff pass is equivalent to the backward pass of message-passing.

Suppose the forward messages are computed as

\[
m_{i \rightarrow i+1}(x_{i+1}) = \sum_{x_i} \phi_{i,i+1}(x_i, x_{i+1}) m_{i-1 \rightarrow i}(x_i) \quad \forall i, x_i
\]

\[
Z = \sum_{x_n} m_{n-1 \rightarrow n}(x_n)
\]

If we ran the backward pass of message-passing, we would compute

\[
m_{n+1 \rightarrow n}(x_n) = 1 \quad \forall x_n
\]

\[
m_{i+1 \rightarrow i}(x_i) = \sum_{x_{i+1}} \phi_{i,i+1}(x_i, x_{i+1}) m_{i+2 \rightarrow i+1}(x_{i+1}) \quad \forall i < n, x_i
\]

**Induction step**: assume the claim holds for \(j = i + 1\). By the formula for \(m_{i \rightarrow i+1}\) and the induction hypothesis

\[
\frac{d \log Z}{dm_{i-1 \rightarrow i}(a)} = \sum_{x_{i+1}} \frac{d \log Z}{dm_{i \rightarrow i+1}(x_{i+1})} \frac{dm_{i \rightarrow i+1}(x_{i+1})}{dm_{i-1 \rightarrow i}(a)}
\]

\[
= \sum_{x_{i+1}} \frac{1}{Z} m_{i+2 \rightarrow i+1}(x_{i+1}) \phi_{i,i+1}(a, x_{i+1})
\]

\[
= \frac{1}{Z} m_{i+1 \rightarrow i}(a)
\]

Instead, we can use backpropagation to compute \(\frac{d \log Z}{dv}\) for all intermediate quantities \(v\) in the forward pass.

**Claim**: the derivative of \(\log Z\) with respect to the forward messages gives the backward messages

\[
\frac{d \log Z}{dm_{i-1 \rightarrow i}(a)} = \frac{1}{Z} m_{i+1 \rightarrow i}(a)
\]

**Proof** (by induction)

**Base case**: from the formula for \(Z\), we see

\[
\frac{d \log Z}{dm_{n-1 \rightarrow n}(a)} = \frac{1}{Z} \frac{dZ}{dm_{n-1 \rightarrow n}(a)} = \frac{1}{Z} = \frac{1}{Z} m_{n+1 \rightarrow n}(a)
\]
Motivation

Computing expectations is important!

\[ \mathbb{E}_{p(x)}[f(X)] = \int p(x)f(x)dx \]

**Example**: suppose \( p(x) \) is an MRF, then

\[ P(X_u = a, X_v = b) = \mathbb{E}_{p(x)}[I[X_u = a, X_v = b]] \]

In general, computing expectations is hard, so we need an approximation.

Monte Carlo methods

In a Monte Carlo method, we approximate an expected value by a sample average. Draw \( N \) samples \( X_1, \ldots, X_N \sim p(x) \), then

\[ \mathbb{E}_{p(x)}[f(X)] \approx \frac{1}{N} \sum_{n=1}^{N} f(X_n). \]

Nice properties:
- Unbiased
- Variance decreases like \( \frac{1}{N} \).
- Measure arbitrary properties by choosing \( f \).

Not nice properties: sampling is algorithmically/computationally hard in general.

Examples

Suppose we have \( p(x) = 12(x^2 - x^3) \), where \( x \in [0, 1] \). Or suppose we have an MRF with a cycle.

\[ p(x) \]

**Question**: How do we sample from these distributions? **Answer**: We need an algorithm.

Gibbs Sampling
Markov Chain Monte Carlo Overview

- Markov chain Monte Carlo (MCMC) methods iteratively construct samples from a given “target distribution” \( p(x) \).
- They require only access to the unnormalized distribution, so can apply easily to models like MRFs.
- Formally, they work by constructing a Markov chain that has the target distribution \( p(x) \) as its limiting distribution.
- We’ll introduce one MCMC method today, and then start to develop some of the theory needed to understand the algorithm.
- Importance / applications: statistical physics, econometrics, ecology, epidemiology, weather modeling, ... 

The Gibbs Sampler

A simple and powerful algorithm! Assume \( X = (X_1, \ldots, X_D) \).
Initialize all variables arbitrarily, then repeatedly update each variable by sampling from its conditional distribution given all other variables.

Gibbs sampler

- Initialize \( x_1, \ldots, x_D \)
- Repeat
  - For \( i = 1 \) to \( D \), resample \( x_i \sim p(X_i \mid X_{-i} = x_{-i}) \)
  - Record \( x = (x_1, \ldots, x_D) \) as one sample

One sample is generated after each loop through all of the variables.

Example: Cycle MRF

Suppose \( p(x) = \prod_{i=1}^{n} \phi(x_i, x_{i+1}) \) (mod \( n \))

Then \( p(x_i \mid x_{-i}) \propto \phi(x_{i-1}, x_i)\phi(x_i, x_{i+1}) \) (factor reduction!)
For a general MRF: \( p(x_i \mid x_{-i}) \propto \prod_{c:i \in c} \phi_c(x_i, x_c \setminus i) \)
Markov Chain Theory

A discrete Markov chain is a set of states with transition probabilities between each pair of states. Example (note: not a graphical model!)

Transition Matrix

- The probabilistic transitions in the state diagram can also be represented by an equivalent matrix of transition probabilities.
- The “from” states are rows and the “to” states are columns.

Markov Chains: Simulation and State Sequences

- To simulate a Markov chain, we draw $x_0 \sim p_0$, then repeatedly sample $x_{t+1}$ given the current state $x_t$ according to the transition probabilities $T$. 

Example:

```
From  To
0.8  0.1  0.1
0.2  0.3  0.5
0.5  0.2  0.3
```
Markov Chain: Formal Definition

By repeatedly making random transitions from a starting state, we generate a chain of random variables \(X_0, X_1, X_2, X_3, \ldots\).

Formally, a Markov chain is specified by:

- A set of states \(\{1, 2, \ldots, D\}\)
- A starting distribution \(p_0(i) = P(X_0 = i)\).
- Transition probabilities \(T_{ij} = P(X_{t+1} = j | X_t = i)\) for all \(i, j \in \{1, 2, \ldots, D\}\).

A Markov chain assumes the Markov property:

\[
P(X_t = x_t | X_0 = x_0, X_1 = x_1, \ldots, X_{t-1} = x_{t-1}) = P(X_t = x_t | X_{t-1} = x_{t-1})
\]

Markov Chain Questions

Three important questions:

1. What is the joint probability of a sequence of states of length \(N\)?
2. What is the marginal probability distribution over states after a given number of steps \(t\)?
3. What happens to the probability distribution over states in the limit as \(t\) goes to infinity?

Markov Chain Factorization

**Question:** What is the joint probability over the state sequence \(x_0, \ldots, x_N\)?

**Answer:** by the Markov property:

\[
P(X_1 = x_1, \ldots, X_N = x_N | X_0 = x_0) = P(X_1 = x_1 | X_0 = x_0) \times P(X_2 = x_2 | X_1 = x_1) \times \cdots \times P(X_N = x_N | X_{N-1} = x_{N-1})
\]

Shorter version:

\[
p(x_1, x_2, \ldots, x_N | x_0) = p(x_1 | x_0)p(x_2 | x_1) \cdots p(x_N | x_{N-1})
\]

\[
= T_{x_0x_1} \times T_{x_1x_2} \times \cdots \times T_{x_{N-1}x_N}
\]

The \(t\)-Step Distribution for Fixed \(x_0\)

**Question:** What is the marginal probability distribution after \(t\) steps given that the chain starts at \(x_0\)? i.e., what is \(p(x_t | x_0)\)?

**Examples:**

\[
p(x_1 | x_0) = T_{x_0x_1},
\]

\[
p(x_2 | x_0) = \sum_{x_1} p(x_1, x_2 | x_0) = \sum_{x_1} p(x_1 | x_0) T_{x_1x_2}.
\]

In general, we have the recursive expression:

\[
p(x_2 | x_0) = \sum_{x_1} p(x_1 \cdot x_2 | x_0) = \sum_{x_1} p(x_1 | x_0) T_{x_1x_2}.
\]

\[
p(x_t | x_0) = \sum_{x_{t-1}} p(x_{t-1} \cdot x_t | x_0) = \sum_{x_{t-1}} p(x_{t-1} | x_0) T_{x_{t-1}x_t}.
\]
The \( t \)-Step Distribution for Random \( X_0 \)

**Question:** What is the marginal probability distribution after \( t \) steps given that \( X_0 \sim p_0 \)? I.e., what is \( p(x_t) \)?

By similar logic:

\[
p(x_1) = \sum_{x_0} p(x_0, x_1) = \sum_{x_0} p(x_0) T_{x_0 x_1},
\]

\[
p(x_1) = \sum_{x_0} p(x_1, x_2) = \sum_{x_1} p(x_1) T_{x_1 x_2}.
\]

In general:

\[
p(x_t) = \sum_{x_{t-1}} p(x_{t-1}, x_t) = \sum_{x_{t-1}} p(x_{t-1}) T_{x_{t-1} x_t}.
\]

\( t \)-Step Recurrence as Matrix-Vector Multiplication

The recurrences for the \( t \)-step distributions can be expressed using matrix-vector multiplication. Let \( p_t \) be the row-vector

\[
p_t = [P(X_t = 1), P(X_t = 2), \ldots, P(X_t = D)].
\]

Then, since \( T_{ij} = P(X_t = j|X_{t-1} = i) \), we can write the above recursive relationship as

\[
p_t = p_{t-1} T.
\]

\( t \)-Step Distribution as Matrix Power

By unrolling the recurrence, the \( t \)-step distribution can be obtained as a matrix power

\[
p_t = p_{t-1} T
\]

\[
= (p_{t-2}) T
\]

\[
= (p_{t-3}) T T
\]

\[
= (p_{t-4}) T T T
\]

\[
= \ldots
\]

\[
= p_0 T^t.
\]

\( t \) times
Thus

\[ p_t = p_0 T^t. \]

This also implies that \( T^t \) is the \( t \)-step transition matrix

\[ (T^t)_{ij} = P(X_t = j | X_0 = i) = P(X_{s+t} = j | X_s = i) \]