STAT 31512 Spring 2024 Scribe: Robert Koort
March 18, 2024 Lecturer: Frederic Koehler
These notes have not received the scrutiny of publication. They could be missing important references, etc.

Lecture 1 - Logistics and Intro to Sampling

1 Introduction

The goal of the course is to answer the following question: given a probability distribution P, how should we generate a sample $x \sim P$ on a computer?

Definition 1. Probability Distribution: description of a distribution, sometimes implicit (a way to compute the density $p(x)$ for any x), sometimes just a characterization of P
In this class, will usually work with energy-based models.

Definition 2. Energy based models are probability distributions that take the form $p(x) = \frac{1}{Z}\exp(H(x))$ where $H(x)$ is our log-likelihood/energy. We generally assume $H(x)$ is easy to compute

Definition 3. The partition function is denoted Z in the energy-based model. This is often hard to compute and we generally regard as a constant. Sometimes we instead may work in terms of use $\log Z$ and we denote that "free energy" / "cumulant generating function"

Definition 4. Exponential Family models, which include most common distributions, take form $p_\theta(x) = \frac{1}{Z}\exp(\langle\theta, F(x)\rangle)$ where θ the is called the canonical parameter, $F(x)$ the sufficient statistic

1.1 Example 1: Ising Model

Let:
$x \in \{\pm 1\}^n$ where n is our dimension / "number of sites."
$p(X) = \frac{1}{Z}\exp(\frac{1}{2}\langle X, J_x\rangle + \langle h, x\rangle)$ where J, h are parameters.

Note that it kinda looks like the gaussian $p(x) \propto \exp(-\frac{1}{2}\langle x - \mu, \Sigma^{-1}(x-\mu)\rangle)$
If it helps, we can relate J, Σ^{-1} - they have a similar role. They encode interaction/covariation between dimensions respectively.

1.1.1 2d Ising Model

In the 2d case, we define $p(x) \propto \exp(\frac{\beta}{2}\sum_{i\sim j} x_i x_j)$[1]
Here, β is our inverse temperature. So, as β gets bigger, the temperature gets cooler.

We can use a water metaphor to frame this:

1. At high temperature, water becomes gas
2. At medium temperature, water is a liquid
3. at low temperature, waster is a solid
4. At specific critical temperature in between states, properties change very rapidly (boiling point and freezing point)

[1] notation Note: when talking about points in a graph, we use notation $i \sim j$ to denote neighbors on the graph

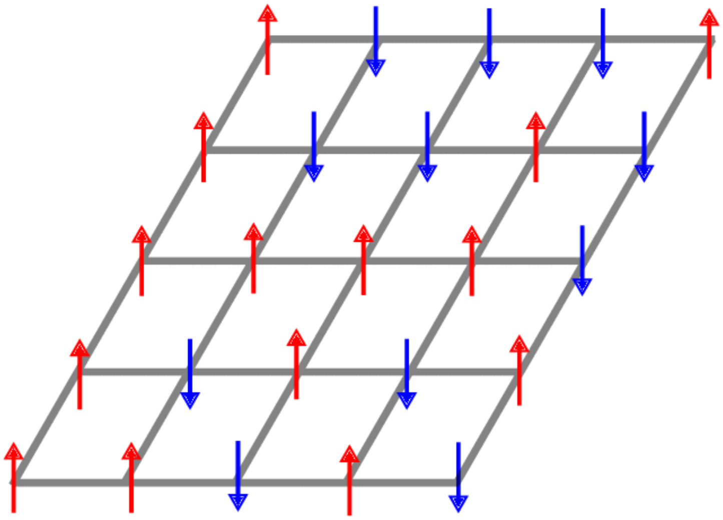

Figure 1: Visual example of a sample from the 2d Ising model. Each node on our lattice has either positive (+1) or negative (−1) magnetic spin

Our Ising model (and many others) works similarly.
At temperatures higher than the critical temperature, it looks basically IID.
But at temperatures lower than the critical temperature, it has "plus phase" and "minus phase" where nodes are largely of the respective state with small pockets of the other.

The takeaway is as follows: with the same parameters, but different temperatures, the model can perform very differently.
We need to be aware of this: one sampling algorithm might work well in one regime but not another.

As an aside: we can introduce a new term, $\sigma_{i,j} \sim \text{Uniform}\{\pm 1\}$ such that $p(x) \propto \exp(\frac{\beta}{2}\sum_{i\sim j}\sigma_{i,j}x_i x_j)$
This is called the **Edwards-Anderson Model**, or spin glass model. We don't know how to sample from in in 3d but, for scientific reasons people believe they know how it will behave. Cool.

1.2 Example 2: Uniformly Random Minimum Spanning Tree

we have a graph $G = (V, E)$, where V denotes our set of vertices and E our set of edges.
we have a probability measure $P(S) = \frac{1}{Z}\mathbb{1}(S \text{ is a spanning tree})$ with $S \subseteq E$
So, sampling from this measure would generate a uniformly random spanning tree from the graph. But, we need to define some terms first.

Definition 5. A spanning tree is a tree (connected graph with no cycles) that spans the graph (all vertices are included)
The naive approach to sampling a uniformly random spanning tree is to enumerate over all subsets of edges to find all the spanning trees. That's a terrible algorithm: in worst case, requires evaluating 2^{n^2} subsets.

1.2.1 Down-Up Walk

The "best" (fastest) algorithm is Down-Up Walk.
Starting with a spanning tree T on our graph G, the Down-Up part only consists of two steps:

1. Down step: pick an edge $e \in T$ uniformly at random (u.a.r.). Remove it from the subgraph.

2. Up step: select an edge e u.a.r. from E such that its addition creates a spanning tree. Add it to the subgraph.

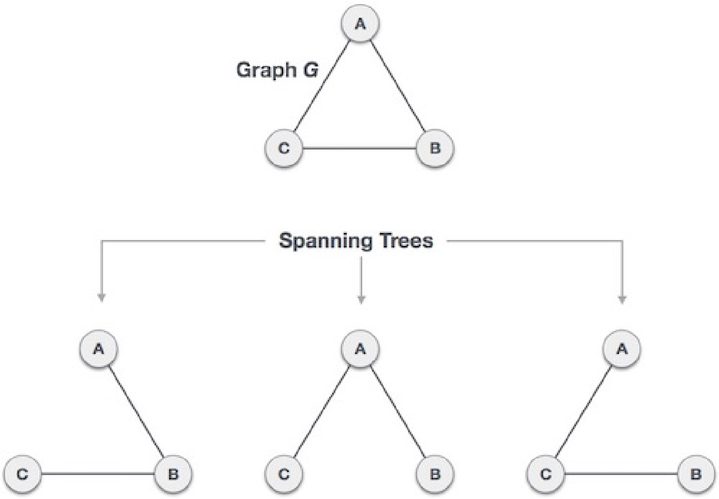

Figure 2: Spanning trees of a fully connected graph with three nodes.

The full algorithm is as follows:

Algorithm 1 Down-Up Algorithm

```
for $t := 1$ to $t_{\text{final}}$ do
    $T_t = \text{DownUp}(T_{t-1})$
end for
```

where T_0 is any arbitrary spanning tree and out output is the last tree generated.
if we pick $t_{\text{final}} \approx O(n \log n)$, then we end up with a uniformly random spanning tree!

2 Determining an algorithm works

When evaluating a sampling algorithm, if we can show we end up sampling $x \sim P$, then we have exact sampling.
In practice, we sample $x \sim P'$ such that $d(P, P') \leq \epsilon$

2.1 Example: 1d Ising model

The 1d Ising model gives us the spin of nodes on a line, interacting with neighbors. Nodes take values $x \in \{\pm 1\}$
The pdf is defined $p(x) = \frac{1}{Z} \exp(\beta \sum\limits_{i \sim j} x_i x_j)$ where $\beta \geq 0$ is our inverse temperature

As it turns out, computing our normalizing constant Z is actually really close to sampling!

We know, from the pdf, that

$$Z = \sum_{x \in \{\pm 1\}^n} \exp(\beta \sum_{i \sim j} x_i x_j)$$

We we can compute this, but it's very slow - $O(2^n)$.
in 1d, it's easy to explicity write what it means to be a neighbor.

$$Z = \sum_{x \in \{\pm 1\}^n} \exp(\beta \sum_{i=1}^{n-1} x_i x_{i+1})$$

Now, we can reframe out problem in terms of "edges" $y_i = x_i x_{i+1} = 1$ if $x_i = x_{i+1}$, else -1
Importantly, (x_1, y) is a bijection of $(x_1, ..., x_n)$
pf:
Given x_1, y
$x_2 = x_1^2 x_2 = y_1 x_1$
$x_3 = x_2^2 x_3 = y_2 x_2$
and so on...
The consequence is that, without losing any information, we can write:

$$Z = \sum_{x_1} \sum_{y \in} \exp(\sum_{i=1}^{n-1} \beta y_i)$$

In this form, it still takes $O(2^n)$ to calculate. But, it's nicer - we just have a linear term y_i in the exponential in the inner sum instead of a quadratic term $x_i x_{i+1}$.
We can keep reformulating:

$$\begin{aligned} Z &= \sum_{x_1} \prod_{i=1}^{n-1} (e^{\beta} + e^{-\beta}) \\ &= 2(e^{\beta} + e^{-\beta})^{n-1} \\ &= 2^n \cosh(\beta)^{n-1} \end{aligned}$$

So, we have a closed form for Z!.

We also know $\Pr(X_1 = x_1, y = y) = p(X) \propto \exp(\beta \sum_{i \sim j} x_i x_j) = \prod_{i=1}^{n-1} \exp(\beta y_i)$ - We rewrite using the same work as we used to find Z.
This factorizes!
So, we have:

$$\begin{aligned} P(Y_1 = y_1) &= \frac{\exp(\beta y_1)}{\exp(\beta) + \exp(-\beta)} \\ E[y_1] &= \frac{\exp(\beta) - \exp(-\beta)}{\exp(\beta) + \exp(-\beta)} = \tanh(\beta) \end{aligned}$$

which we can use to exactly sample from $p(X)$!

What are the takeaways?

1. It's not immediately obvious from definition of the distribution how to sample from it

2. But finding the partition function ("counting") gives us a big hint on how to sample from the distribution