# Deep generative models for biologists

Clayton W. Seitz

January 18, 2022

# Outline

Generative Models

Probabilistic Graphical Models

References

# The logic of generative modeling

Say we have a set of variables $\mathbf{x} = (x_1, x_2, ..., x_n)$ which might have some statistical dependence

The variable $\mathbf{x}$ might be an amino acid sequence, DNA sequence, microscopy image, etc.

- Often we are handed a batch of empirical samples $\{\mathbf{x}_i\}_{i=1}^N$
- We want to know the generating distribution $p(\mathbf{x})$

In supervised generative learning, we try to explicity learn the joint distribution $p(\mathbf{x}) = p(x_1|x_2, ..., x_n)p(x_2|x_3, ..., x_n), ..., p(x_n)$, which is generally more difficult than discriminative learning.

# Sampling from a model

To find $p(\mathbf{x})$ we might fit a parametric model with parameters $\theta$ with MLE or some other method

Lets assume we already know the model type and parameters $\theta$

As a toy example, perhaps $x \sim \mathcal{N}(\mu, \sigma^2)$ and we know $\theta = (\mu, \sigma)$

In this simple case, we can draw samples by rejection sampling

# Rejection sampling with the uniform distribution

Let $\Omega$ be the state space or *support* of $x$. Let $U(\Omega)$ be the uniform distribution over $\Omega$

Also notice that $p(x) \leq 1 \ \ \forall x \in \Omega$

The following procedure produces a sample $x \sim p(x)$.

1. Sample $u \sim U(\Omega)$
2. Sample $y \sim U([0,1])$
3. If $y < p(u)$ return $y$ as a sample of $p(x)$

This algorithm suffers from the curse of dimensionality. Generally, sampling becomes

# The sampling problem

Dimensionality can make it difficult to sample from $p(\mathbf{x})$ directly. For example, the multivariate Gaussian distribution

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

Sampling must be achieved in another way e.g., Cholesky decomposition or Gibbs sampling

# The sampling problem

We also may not know the proper normalization constant or partition function $Z$. Say we have

$$p(\mathbf{x}) = \frac{1}{Z}\tilde{p}(\mathbf{x})$$

where $p(\mathbf{x})$ is easy to compute but $Z$ is (too) hard to compute.

This very important situation arises in several contexts:

1. In Bayesian models where $p(x_1, x_2) := p(x_1|x_2)p(x_2)$ is easy to compute but $Z = \int p(x_1|x_2)p(x_2)dx_2$ can be very difficult or impossible to compute.

2. In models from statistical physics, e.g. the Ising model, we only know $\tilde{p}(\mathbf{x}) = e^{-H(\mathbf{x})}$ where $H(\mathbf{x})$ is the Hamiltonian

# Sampling the joint distribution

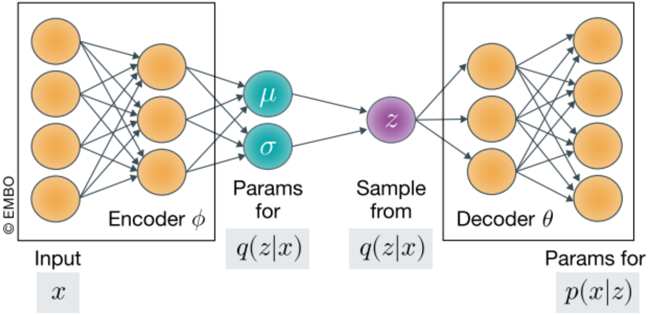
How to generate samples depends on our model

Variational methods can also be used to evaluate $p(\mathbf{x})$ by autoencoding $\mathbf{x}$ (called a variational autoencoder or VAE)

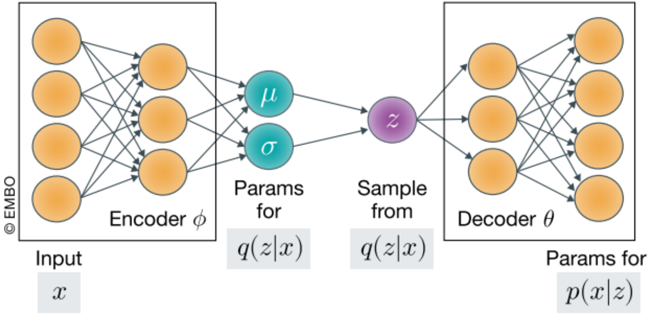
If we have the model parameters we could use Monte-Carlo Markov Chain (MCMC) methods

We will discuss both in the following slides

# The variational autoencoder (VAE)



# Theory of the VAE



# Monte-Carlo Markov Chain (MCMC)

- MCMC algorithms were originally developed in the 1940's by physicists at Los Alamos
- They were interested in modeling the probabilistic behavior of collections of atomic particles
- Simulation was difficult – the normalization constant $Z$ was not known
- The term "Monte-Carlo" was coined at Los Alamos.
- Ulam and Metropolis overcame this problem by constructing a Markov chain for which the desired distribution was the stationary distribution
- Introduced to statistics and generalized with the Metropolis-Hastings algorithm (1970) and the Gibbs sampler of Geman and Geman (1984).

# Markov Chains

For a state space $\Omega$ s.t. $\mathbf{x}_t \in \Omega$. $\mathbf{x}_t$ is a Markov process if:

$$P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_{t-2}, ..., \mathbf{x}_{t-N}) = P(\mathbf{x}_t|\mathbf{x}_{t-1})$$

which is commonly called the *memoryless property*.

- $\mathbf{x}_t$ can be generally be $N$-dimensional
- The chain is called *homogeneous* if $T(\mathbf{x}_t|\mathbf{x}_{t-1})$ is time-invariant.
- For discrete $\Omega$, $T$ is a matrix of probabilities with $T_{ij} = \Pr(i \to j)$
- For continuous $\Omega$, $T$ is the joint probability density $T(x_t, x_{t-1})$

# Markov Chains

The Chapman-Kolmogorov equation marginalizes $T(x_t, x_{t-1})$:

$$\begin{aligned} P(\mathbf{x}_t) &= \int T(x_t, x_{t-1}) d\mathbf{x}_{t-1} \\ &= \int T(x_t | x_{t-1}) P(\mathbf{x}_{t-1}) d\mathbf{x}_{t-1} \end{aligned}$$

The chain satisfies *detailed balance* if

$$T(x_t, x_{t-1}) P(x_t) = T(x_{t-1}, x_t) P(x_{t-1})$$

which guarantees there is a unique stationary distribution $P_0(x_t)$

# Monte-Carlo Markov Chain (MCMC)

A stationary distribution satisfies

$$P_0(\mathbf{x}_t) = \int T(x_t|x_{t-1})P_0(\mathbf{x}_{t-1})d\mathbf{x}_{t-1}$$

- If a process is Markov e.g., Brownian motion, Ornstein-Uhlenbeck, $P_0(x_t)$ is a solution to the SDE
- We can also design $T(x_t, x_{t-1})$ s.t. $P_0(x_t)$ is a distribution we cannot sample from easily such as the Ising model
- The notion of "time" in the second case is artificial
- There are several MCMC algorithms, we will focus on Gibbs MCMC

# Gibbs sampling

- Suppose $p(\mathbf{x})$ is a p.d.f. or p.m.f. that is difficult to sample from directly.
- Suppose, though, that we *can* easily sample from the conditional distributions e.g., $p(x_1|x_2, ..., x_n)$.
- The Gibbs sampler proceeds as follows:
    1. set $\mathbf{x}$ to some initial starting values
    2. then sample $x_1|x_2, ..., x_n$, then sample $x_2|x_1, ..., x_n$, and so on.

# Gibbs sampling

0. Set $(x_0, y_0)$ to some starting value.
1. Sample $x_1 \sim p(x|y_0)$, that is, from the conditional distribution $X \mid Y = y_0$.
   Current state: $(x_1, y_0)$
   Sample $y_1 \sim p(y|x_1)$, that is, from the conditional distribution $Y \mid X = x_1$.
   Current state: $(x_1, y_1)$
2. Sample $x_2 \sim p(x|y_1)$, that is, from the conditional distribution $X \mid Y = y_1$.
   Current state: $(x_2, y_1)$
   Sample $y_2 \sim p(y|x_2)$, that is, from the conditional distribution $Y \mid X = x_2$.
   Current state: $(x_2, y_2)$
   
   $\vdots$

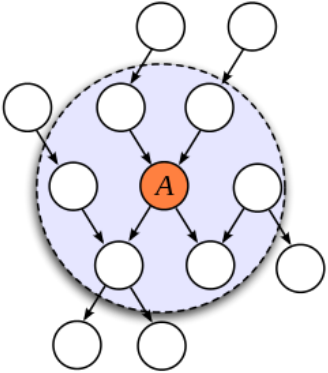
Repeat iterations 1 and 2, M times.

# Bayesian inference using Gibbs sampling

Joint distributions factor according to

$$P(\mathbf{x}) = P(x_1|x_2,...,x_n)P(x_2|x_3,...,x_n),...,P(x_n)$$

$P(x_1|x_2,...,x_n)$ may not include all $n-1$ variables

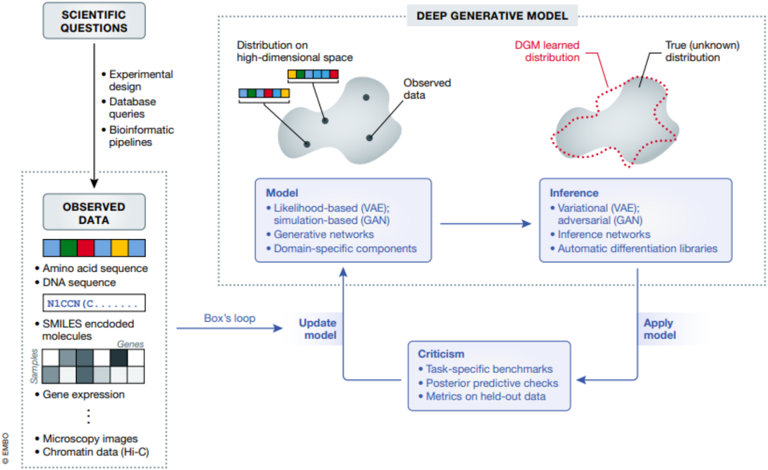



The useful information is called a Markov blanket

# Learning graph structure

Learning the graph structure $\mathcal{G} = (V, E)$ is a common task in machine learning.

# Applying deep generative models to biological data



# Cool biological applications of VAEs

Sequencing, Imaging, Other stuff

# References I