15-780 – Graduate Artificial Intelligence: Probabilistic inference

J. Zico Kolter (this lecture) and Ariel Procaccia
Carnegie Mellon University
Spring 2017

Outline

Probabilistic graphical models

Probabilistic inference

Exact inference

Sample-based inference

Bayesian modeling

A brief look at deep generative models

Outline

Probabilistic graphical models

Probabilistic inference

Exact inference

Sample-based inference

Bayesian modeling

A brief look at deep generative models

Probabilistic graphical models

Probabilistic graphical models are all about representing distributions

$$p(X)$$

where X represents some large *set* of random variables

Example: suppose $X \in \{0,1\}^n$ (n-dimensional random variable), would take $2^n - 1$ parameters to describe the full joint distribution

Graphical models offer a way to represent these same distributions more compactly, by exploiting *conditional independencies* in the distribution

Note: I'm going to use "probabilistic graphical model" and "Bayesian network" interchangeably, even though there are differences

Bayesian networks

A Bayesian network is defined by

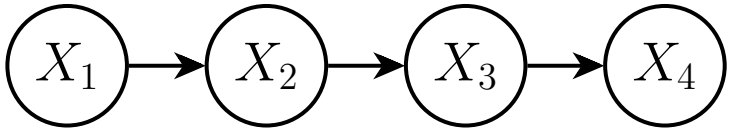
1. A directed acyclic graph, $G = \{V = \{X_1, \ldots, X_n\}, E\}$
2. A set of conditional distributions $p(X_i|\text{Parents}(X_i))$

Defines the joint probability distribution

$$p(X) = \prod_{i=1}^{n} p(X_i|\text{Parents}(X_i))$$

Equivalently: each node is conditionally independent of all non-descendants given its parents

Example Bayesian network



Conditional independencies let us simply the joint distribution:

$$p(X_1, X_2, X_3, X_4) = p(X_1)p(X_2|X_1)p(X_3|X_1, X_2)p(X_4|X_1, X_2, X_3)$$

$2^4 - 1 = 15$ parameters (assuming binary variables)

$$= p(X_1)p(X_2|X_1)p(X_3|X_2)p(X_4|X_3)$$

1 parameter

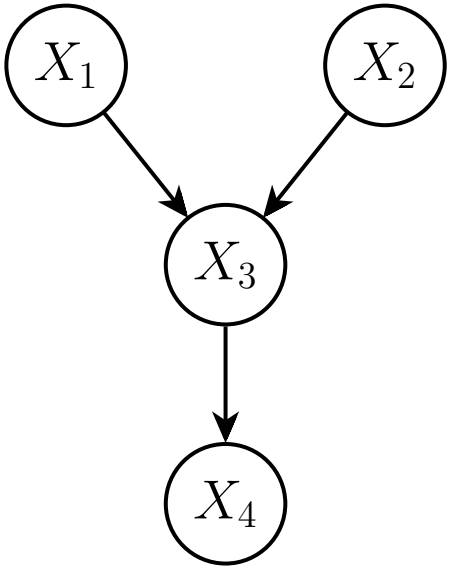
2 parameters

7 parameters

Poll: Simple Bayesian network

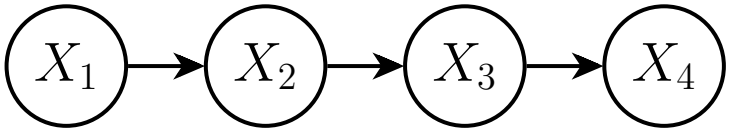
What conditional independencies exist in the following Bayesian network?

1. X_1 and X_2 are marginally independent
2. X_4 is conditionally independent of X_1 given X_3
3. X_1 is conditionally independent of X_4 given X_3
4. X_1 is conditionally independent of X_2 given X_3



Generative model

Can also describe the probabilistic distribution as a sequential "story", this is called a *generative model*



$$
\begin{aligned}
X_1 &\sim \text{Bernoulli}\left(\phi^{(1)}\right) \\
X_2 | \; X_1 = x_1 &\sim \text{Bernoulli}\left(\phi^{(2)}_{x_1}\right) \\
X_3 | \; X_2 = x_2 &\sim \text{Bernoulli}\left(\phi^{(3)}_{x_2}\right) \\
X_4 | \; X_3 = x_3 &\sim \text{Bernoulli}\left(\phi^{(3)}_{x_3}\right)
\end{aligned}
$$

"First sample X_1 from a Bernoulli distribution with parameter $\phi^{(1)}$, then sample X_2 from a Bernoulli distribution with parameter $\phi^{(2)}_{x_1}$, where x_1 is the value we sampled for X_1, then sample X_3 from a Bernoulli …"

More general generative models

This notion of a “sequential story” (generative model) is extremely powerful for describing very general distributions

Naive Bayes:

$$Y \sim \text{Bernoulli}(\phi)$$
$$X_i | Y = y \sim \text{Categorical}\left(\phi_y^{(i)}\right)$$

Gaussian mixture model:

$$Z \sim \text{Categorical}(\phi)$$
$$X | Z = z \sim \mathcal{N}(\mu_z, \Sigma_z)$$

More general generative models

Linear regression:

$$Y|X = x \sim \mathcal{N}(\theta^T x, \sigma^2)$$

Changepoint model:

$$X \sim \text{Uniform}(0,1)$$

$$Y|X = x \sim \begin{cases} \mathcal{N}(\mu_1, \sigma^2) \text{ if } x < t \\ \mathcal{N}(\mu_2, \sigma^2) \text{ if } x \geq t \end{cases}$$

Latent Dirichlet Allocation: M documents, K topics, N_i words/document

$$\theta_i \sim \text{Dirichlet}(\alpha) \text{ (topic distributions per document)}$$
$$\phi_k \sim \text{Dirichlet}(\beta) \text{ (word distributions per topic)}$$
$$z_{i,j} \sim \text{Categorical}(\theta_i) \text{ (topic of } i\text{th word in document)}$$
$$w_{i,j} \sim \text{Categorical}(\phi_{z_i,j}) \text{ (}i\text{th word in document)}$$

Outline

Probabilistic graphical models

Probabilistic inference

Exact inference

Sample-based inference

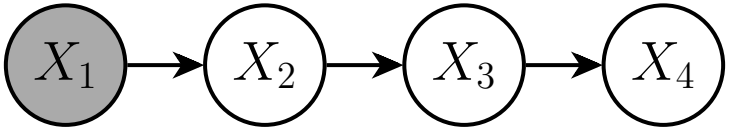
Bayesian modeling

A brief look at deep generative models

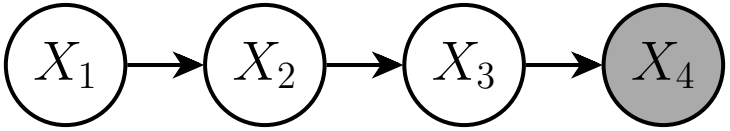
The inference problem

Given observations (i.e., knowing the value of some of the variables in a model), what is the distribution over the other (hidden) variables?

A relatively "easy" problem if we observe variables at the "beginning" of chains in a Bayesian network:



If we observe the value of X_1, then X_2, X_3, X_4 have the same distribution as before, just with X_1 "fixed"



But if we observe X_4 what is the distribution over X_1, X_2, X_3?

Many types of inference problems

Marginal inference: given a generative distribution for $p(\mathrm{X})$ over $X = \{X_1, \ldots, X_n\}$, determine $p(X_{\mathcal{I}})$ for $\mathcal{I} \subseteq \{1, \ldots, n\}$

MAP inference: determine assignment with the maximum probability

Conditional variants: solve either of the two variants conditioned on some observable variables, e.g.

$$p(X_{\mathcal{I}} | X_{\mathcal{E}} = x_{\mathcal{E}})$$

Approaches to inference

There are three categories of common approaches to inference (more exist, but these are most common)

1. Exact methods: Bayes' rule or variable elimination methods

2. Sampling approaches: draw samples from the the distribution over hidden variables, without construction them explicitly

3. Approximate variational approaches: approximate distributions over hidden variables using "simple" distributions, minimizing the difference between these distributions and the true distributions

Outline

Probabilistic graphical models

Probabilistic inference

Exact inference

Sample-based inference

Bayesian modeling

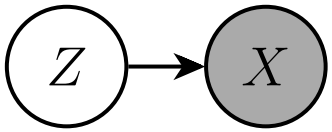
A brief look at deep generative models

Exact inference example

Mixture of Gaussians model:

$$Z \sim \text{Categorical}(\phi)$$
$$X|Z = z \sim \mathcal{N}(\mu_z, \Sigma_z)$$

Task: compute $p(Z|x)$



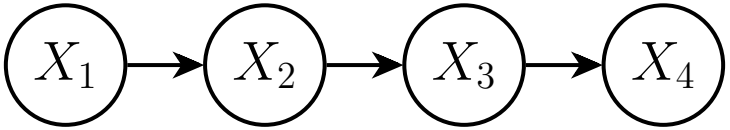
In this case, we can solve inference exactly with Bayes' rule:

$$p(Z|x) = \frac{p(x|Z)p(Z)}{\sum_z p(x|z)p(z)}$$

Exact inference in graphical moels

In some cases, it's possible to exploit the structure of the graphical model to develop efficient exact inference methods

Example: how can I compute $p(X_4)$?

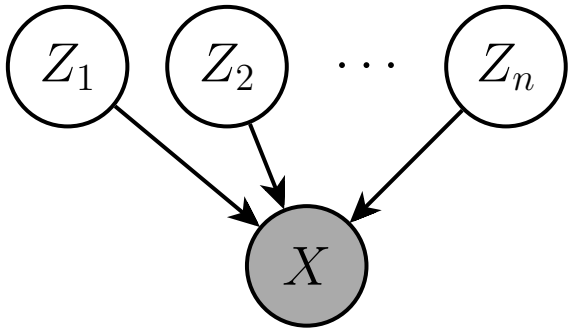


$$p(X_4) = \sum_{x_1, x_2, x_3} P(x_1)P(x_2|x_1)P(x_3|x_2)P(X_4|x_3)$$

Need for approximate inference

In most cases, the exact distribution over hidden variables cannot be computed, would require representing an exponentially large distribution over hidden variables (or infinite, in continuous case)

$$Z_i \sim \text{Bernoulli}(\phi_i), \qquad i = 1, \ldots, n$$
$$X|Z = z \sim \mathcal{N}(\theta^T z, \sigma^2)$$



Distribution $P(Z|x)$ is a full distribution over n binary random variables

Outline

Probabilistic graphical models

Probabilistic inference

Exact inference

Sample-based inference

Bayesian modeling

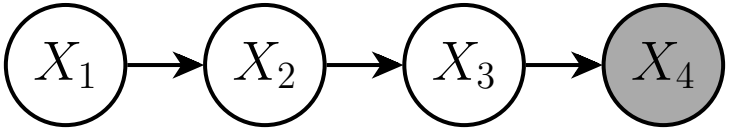
A brief look at deep generative models

Sample-based inference

If we can draw samples from a posterior distribution, then we can approximate arbitrary probabilistic queries about that distribution

A naive strategy (rejection sampling): draw samples from the generative model until we find one that matches the observed data, distribution over other variables will be samples of the hidden variables given observed variables

As we get more complex models, and more observed variables, probability that we see our exact observations goes to zero



Markov Chain Monte Carlo

Let's consider a generic technique for generating samples from a distribution $p(X)$ (suppose distribution is complex so that we cannot directly compute or sample)

Our strategy is going to be to generate samples X^t via some conditional distribution $p(X^{t+1}|X^t)$, constructed to guarantee that $p(X^t) \rightarrow p(X)$

Metropolis-Hastings Algorithm

One of the workhorses of modern probabilistic methods

1. Pick some x^0 (e.g., completely randomly)

2. For $t = 1,2, \ldots$

   Sample:

$$\tilde{x}^{t+1} \sim q(X'|X = x^t)$$

   Set:

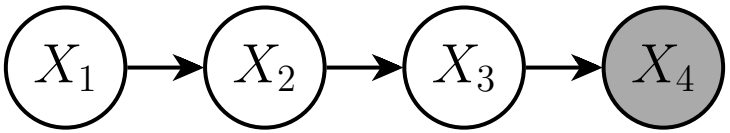
$$x^{t+1} := \begin{cases} \tilde{x}^{t+1} & w.p. \ \min\left(1, \dfrac{p(\tilde{x}^{t+1})q(x^t|\tilde{x}^{t+1})}{p(x^t)q(\tilde{x}^{t+1}|x^t)}\right) \\ x^t & \text{otherwise} \end{cases}$$

Notes on MH

We choose $q(X'|X)$ so that we can *easily* sample from the distribution (e.g., for continuous distributions, it's common to choose)

$$q(X'|X = x) = \mathcal{N}(x'|x; I)$$

Note that even if we cannot compute the probabilities $p(x^t)$ and $p(\tilde{x}^{t+1})$ we *can* often compute their ratio $p(\tilde{x}^{t+1})/p(x^t)$ (requires only being able to compute the unnormalized probabilities), e.g., consider the case



Proof of MH algorithm

Theorem: For samples generated by MH, $p(X^t) \to p(X)$ as $t \to \infty$

Proof: We'll proceed in two parts.

1. (Detailed balance equations) First, we show that given any distribution $p(X)$ and a conditional distribution $p(X'|X)$, then if

$$p(X)p(X'|X) = p(X')p(X|X')$$

and if $p(X'|X) > 0, \forall x, x'$ then repeatedly sampling $x^{t+1} \sim p(X'|X = x^t)$ gives $p(X^t) \to p(X)$

2. The Metropolis-Hastings update gives a distribution that satisfies the detailed balance equations

Proof of MH algorithm (cont)

Part 1: *(not a complete proof)*, detailed balance says that for x^t, x^{t+1}

$$p(x^t)p(x^{t+1}|x^t) = p(x^{t+1})p(x^t|x^{t+1})$$

Summing both sizes over over x^t gives

$$\sum_{x^t} p(x^t)p(x^{t+1}|x^t) = p(x^{t+1})$$

which is equivalent to the fact that $p(X)$ is a *stationary distribution* of the conditional distribution $p(X'|X)$

Under some properties of conditional distributions that we won't cover, repeated sampling from the conditional will converge to the stationary distribution, assuming e.g. conditional has positive probabilities

Proof of MH algorithm (cont)

Part 2: First, note that detailed balance is trivially satisfied for $x^{t+1} = x^t$

$$p(x^{t+1}|x^t)p(x^t) = p(x^t|x^{t+1})p(x^{t+1})$$

Now assuming $x^{t+1} \neq x^t$, suppose that (opposite case proceeds in exactly the same manner)

$$p(x^t)q(x^{t+1}|x^t) \leq p(x^{t+1})q(x^t|x^{t+1})$$

Then:

$$\min\left(1, \frac{p(x^{t+1})q(x^t|x^{t+1})}{p(x^t)q(x^{t+1}|x^t)}\right) = 1$$

$$\min\left(1, \frac{p(x^t)q(x^{t+1}|x^t)}{p(x^{t+1})q(x^t|x^{t+1})}\right) p(x^{t+1})q(x^t|x^{t+1}) = p(x^t)q(x^{t+1}|x^t)$$

Proof of MH algorithm (cont)

So finally, note that

$$
\begin{aligned}
p(x^t)p(x^{t+1}|x^t) &= p(x^t)q(x^{t+1}|x^t)\min\left(1, \frac{p(x^{t+1})q(x^t|x^{t+1})}{p(x^t)q(x^{t+1}|x^t)}\right) \\
&= p(x^t)q(x^{t+1}|x^t) \\
&= \min\left(1, \frac{p(x^t)q(x^{t+1}|x^t)}{p(x^{t+1})q(x^t|x^{t+1})}\right) p(x^{t+1})q(x^t|x^{t+1}) \\
&= p(x^{t+1})p(x^t|x^{t+1})
\end{aligned}
$$

Which shows that the transition probabilities satisfy detailed balance equations.

■

Poll: Metropolis-Hastings

Given the following true distributions p and sampling distributions q would result in creating accurate samples from the true distribution?

1. $p(x) = \mathcal{N}(0,1),\ q(x') = U[0,1]$

2. $p(x) = U[0,1],\ q(x') = \mathcal{N}(0,1)$

3. $p(x) = \mathcal{N}(0,1),\ q(x'|x) = x + U[0,1]$

Gibbs sampling

An application of MH to graphical models leads to what is called *Gibbs sampling*

Suppose we want to draw a sample from $p(Z|X = x)$ (i.e., sample over unobserved variables given observed variables)

1. Initialize z randomly

2. Repeat: pick some i and sample

$$z_i \sim P(Z_i|Z_{\neg i} = z_{\neg i}, X = x)$$

Practical to implement as long as we can sample from a variable given a fixed value of all other variables (can exploit independence structure)

Gibbs as Metropolis-Hastings

We can derive Gibbs sampling as an application of the MH algorithm, with the proposal distribution (omitting X terms for simplicity)

$$q_i(Z'|Z) = \begin{cases} z'_i \sim P(Z_i|Z_{\neg i} = z_{\neg i}) \\ z'_j = z_j \end{cases}$$

Under this distribution, proposal is *always* accepted:

$$\frac{p(z')q_i(z|z')}{p(z)q_i(z'|z)} = \frac{p(z'_i|z'_{\neg i})p(z'_{\neg i})p(z_i|z'_{\neg i})}{p(z_i|z_{\neg i})p(z_{\neg i})p(z'_i|z_{\neg i})} = \frac{p(z'_i|z'_{\neg i})p(z'_{\neg i})p(z_i|z'_{\neg i})}{p(z_i|z'_{\neg i})p(z_{\neg i})p(z'_i|z'_{\neg i})} = 1$$

Technically, this uses a *different* q_i selected at random for each Z_i variable, but we can show that the product of all these individual q_i's lead to a single "global" q that still has all the necessary properties

Outline

Probabilistic graphical models

Probabilistic inference

Exact inference

Sample-based inference

Bayesian modeling

A brief look at deep generative models

Maximum likelihood estimation

Our discussion of probabilistic modeling thus far has maintained a separation between *variables* and *parameters*

Roughly speaking: variables are the things we take expectations over (or sample), and parameters are the things we optimize

E.g. maximum likelihood estimation required that we solve the problem (given observed data $x^{(i)}$):

$$\underset{\theta}{\text{maximize}} \sum_{i=1}^{m} \log p(x^{(i)}; \theta)$$

Bayesian statistics

In Bayesian statistics, everything (including "parameters" θ) is a random variable, we write likelihoods now as

$$p\left(x^{(i)}\middle|\theta\right)$$

In order for these probabilities to be well-defined, we need to define ***prior distribution*** $p(\theta;\alpha)$ on the "parameters" themselves, where α are hyperparameters (typically fixed and not estimated at all)

Instead of finding a point estimate of θ, in Bayesian statistics we try to quantify the *distribution* of $\theta|X$ (θ given the observed data), called the ***posterior distribution***

$$p(\theta|X) = \frac{p(X|\theta)p(\theta;\alpha)}{\int p(X|\theta)p(\theta;\alpha)d\theta}$$

Bayesian linear regression

Bayesian linear regression model

$$\theta \sim \mathcal{N}(0, \rho I)$$
$$Y|\theta, x \sim \mathcal{N}(\theta^T x, \sigma^2)$$

Theorem: the posterior distribution is given by

$$\theta|x^{(1:m)}, y^{(1:m)} \sim \mathcal{N}(\mu, \Sigma)$$
$$\Sigma = \left(\frac{1}{\rho} I + \frac{1}{\sigma^2} X^T X\right)^{-1}$$
$$\mu = \frac{1}{\sigma^2} \Sigma X^T y$$

Where X and y and the normal matrix/vector of inputs/outputs

Key point: *posterior* distribution over θ is also Gaussians

Bayesian linear regression

Proof:

$$
\begin{aligned}
p(\theta|x^{(1:m)}, y^{(1:m)}) &= \frac{p(y^{(1:m)}|x^{(1:m)}, \theta)p(\theta)}{\int p(y^{(1:m)}|x^{(1:m)}, \theta')p(\theta')d\theta'} \\
&= c_1 \cdot p(y^{(1:m)}|x^{(1:m)}, \theta)p(\theta) \\
&= c_2 \cdot \prod_{i=1}^{m} \exp\left(\frac{1}{2\sigma^2}\left\|y^{(i)} - \theta^T x^{(i)}\right\|_2^2\right) \exp\left(\frac{1}{2\rho}\|\theta\|_2^2\right) \\
&= c_3 \cdot \exp\left(\theta^T\left(\frac{1}{2\rho}I + \frac{1}{2\sigma^2}X^T X\right)\theta^T - \frac{2}{2\sigma^2}\theta^T X^T y\right) \\
&= c_4 \cdot \exp\left(\frac{1}{2}(\theta-\mu)^T\Sigma^{-1}(\theta-\mu)\right)
\end{aligned}
$$

where $\Sigma = \left(\frac{1}{\rho}I + \frac{1}{\sigma^2}X^T X\right)^{-1}, \mu = \frac{1}{\sigma^2}\Sigma X^T y$, which is a Gaussian distribution with the given mean and covariance. ■

Conjugate priors

You may hear this term if you read about Bayesian statistics

All this is saying is the following: suppose

$$\theta \sim F(\alpha) \quad (F \text{ is some distribution})$$
$$X|\theta \sim G(\theta) \ (G \text{ some other distribution})$$

Then if F is a ***conjugate prior*** for G

$$\theta|X \sim F(\alpha')$$

i.e., the posterior has the same type of distribution as the prior

This is quite useful, as it represents just about the only case where we can represent the posterior distribution exactly

Conjugate priors and limitations

Example: Normal distribution is conjugate for mean parameter of Normal (see Bayesian linear regression), Inverse Gamma is conjugate for variance parameter

Example: Beta distribution is conjugate prior for Bernoulli, Dirichlet is conjugate for categorical

In the vast majority of cases, you won't use exact conjugate priors, meaning you can't come up with a closed form distribution for the parameters given the data

Need to resort to approximate inference methods, often sampling

(Simplified) Bayesian changepoint detection

Changepoint detection:

$$X \sim \text{Uniform}(0,1)$$

$$Y|x \sim \begin{cases} \mathcal{N}(\mu_1, \sigma^2) \text{ if } x < t \\ \mathcal{N}(\mu_2, \sigma^2) \text{ if } x \geq t \end{cases}$$

Bayesian changepoint detection:

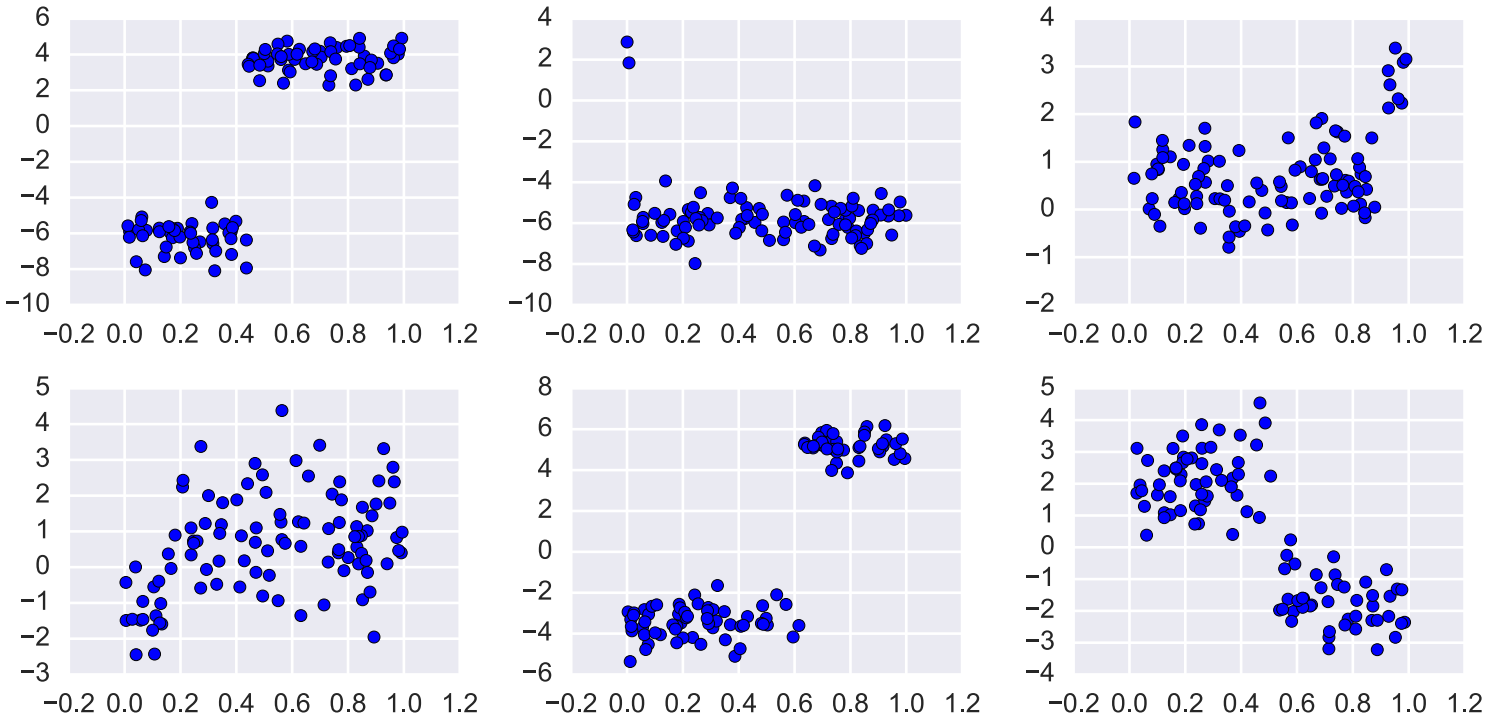
$$t \sim \text{Uniform}(0,1)$$

$$\mu_1, \mu_2 \sim \mathcal{N}(0, \nu^2)$$

$$\sigma^2 \sim \text{InverseGamma}(\alpha, \beta)$$

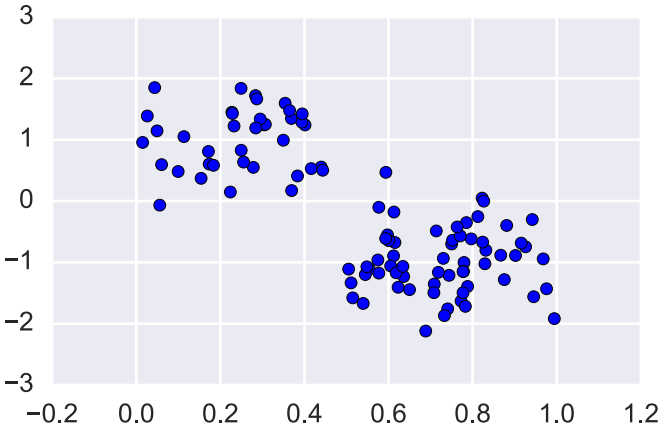
$$Y|x \sim \begin{cases} \mathcal{N}(\mu_1, \sigma^2) \text{ if } x < t \\ \mathcal{N}(\mu_2, \sigma^2) \text{ if } x \geq t \end{cases}$$

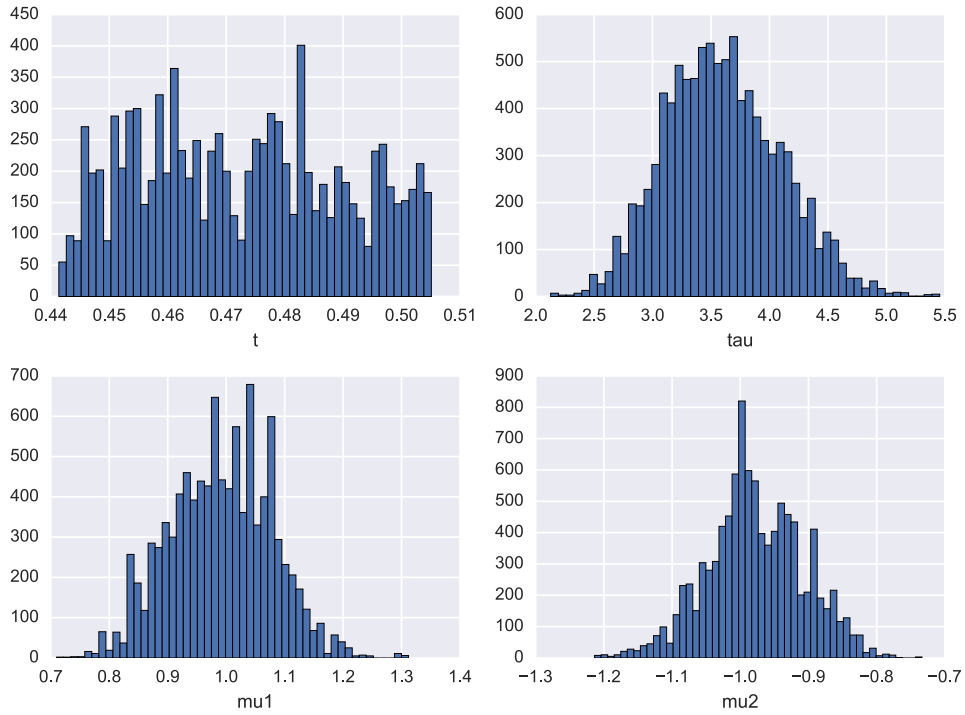
Samples from Generative model



Adding observations

Now suppose we observe the following pairs of (x, y) samples, this updates the posterior over samples





Probabilistic programming

In recent years, there has been substantial effort to “automate” the specification of probabilistic models and inference within these models

In probabilistic programming languages, users specify the model similar to writing code, specify the observed variables (if any), and then perform inference (usually sampling-based) to compute posterior

Some common examples: PyMC, Stan, Edward

Outline

Probabilistic graphical models

Probabilistic inference

Exact inference

Sample-based inference

Bayesian modeling

A brief look at deep generative models

Deep generative models

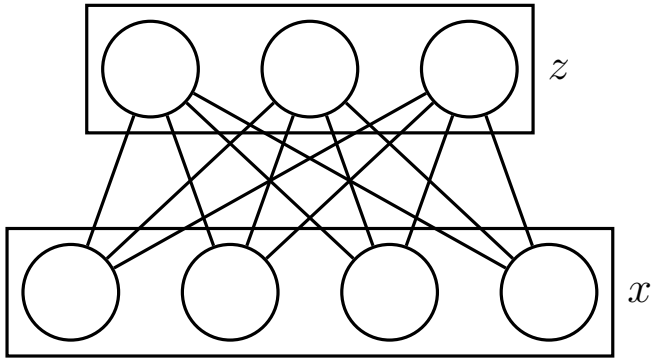
Probabilistic models + deep learning (what could be better?)

A huge landscape, going back many years, and we will just briefly mention a few models

- (Deep) restricted Boltzmann Machines
- Deep directed models (e.g. variational autoencoders)
- Generative adversarial networks

Restricted Boltzmann machine

An early *undirected* graphical model (Smolensky, 1986) that captures joint distribution over (Bernoulli) observed variables x and hidden variables z



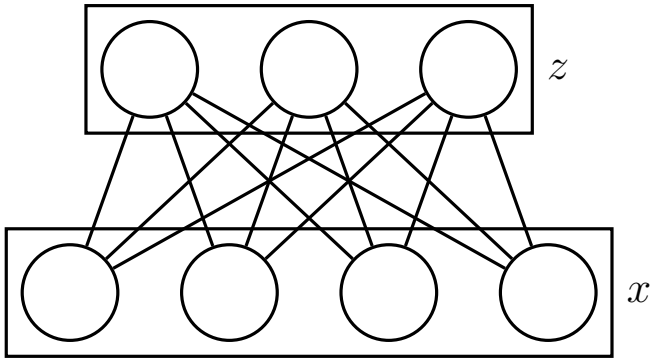
$$p(x, z; \theta) \propto \exp(x^T W z + b_1^T x + b_2^T z)$$
$$\theta = \{W, b_1, b_2\}$$

Training involves maximum likelihood estimation

$$\underset{\theta}{\text{maximize}} \sum_{i=1}^{m} \log p(x^{(i)}; \theta) \equiv \underset{\theta}{\text{maximize}} \sum_{i=1}^{m} \log \sum_{z} p(x^{(i)}, z; \theta)$$

Restricted Boltzmann machine (cont)

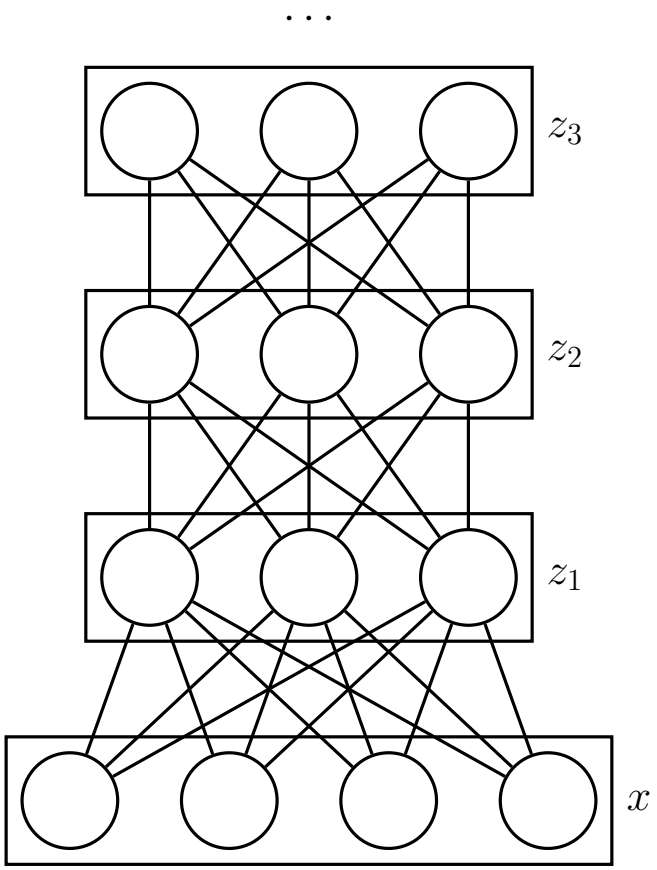
Sampling from the distribution $p(x, z; \theta)$ is also non-trivial, need to resort to MCMC methods



But, (block) Gibbs sampling has a nice form for such models: sample from $p(x|z;\theta) = \prod_i p(x_i|z;\theta)$ then from $p(z|x;\theta) = \prod_i p(z_i|x;\theta)$

This sample-based inference is typically used in training

Deep RBMs



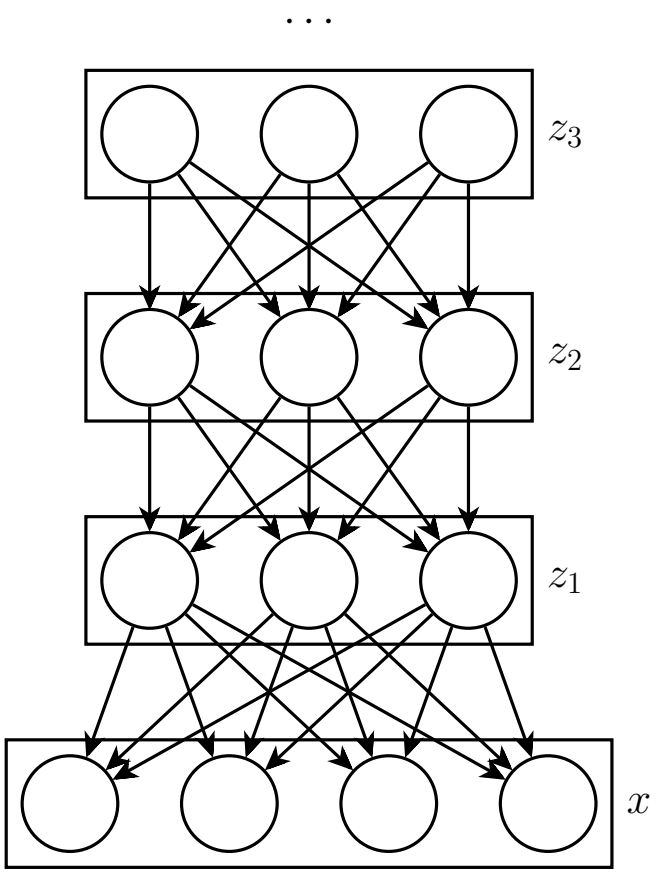
We can extend the RBM model to the "deep" setting

Training and sampling are still both "hard" (in fact even harder now), both involve MCMC sampling

Despite this, a lot of interesting tricks for training, e.g. in layer-wise fashion (e.g. Hinton 2006)

(Above paper partly responsible for resurgence of deep learning interest)

Deep directed models



Replace undirected models with directed (generative) model

$$p(x, z_{1:k}; \theta) = p(z_k; \theta) p(z_{k-1} | z_k; \theta) \cdots p(x | z_1; \theta)$$

Sampling is now "easy": supposing some simple distribution for $p(z_k)$ (e.g., independent Bernoulli) and conditionals, just a matter of simple random sampling

Training is still challenging, need inference to compute posterior distribution

$$\underset{\theta}{\text{maximize}} \sum_{i=1}^{m} \log \sum_{z} p(x^{(i)}, z; \theta)$$

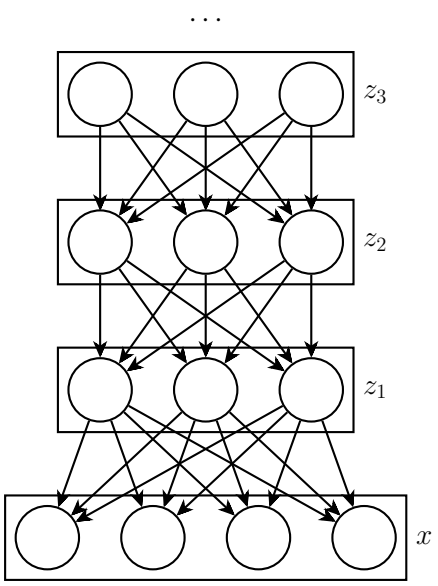
Generative adversarial models (GANs)

An alternative approach to training deep directed models: try to build a classifier that can "tell apart" generated samples from real data

$$\underset{\theta_g}{\text{minimize}} \ \underset{\theta_d}{\text{maximize}} \ \frac{1}{m}\sum_{i=1}^{m} \log p(x^{(i)}\,;\theta_d) + \mathbf{E}_{x\sim p(x,z;\theta_g)}[\log(1-p(x\,;\theta_d))]$$

Training requires solving a min-max optimization problem but current results suggest that it can generate very realistic samples

Has generated a lot of interest in the past year, some impressive results



Examples of GANs

Samples of bedrooms (no training example looks like these in training set)

From (Radford et al., 2016)

Text to image generating using GANs

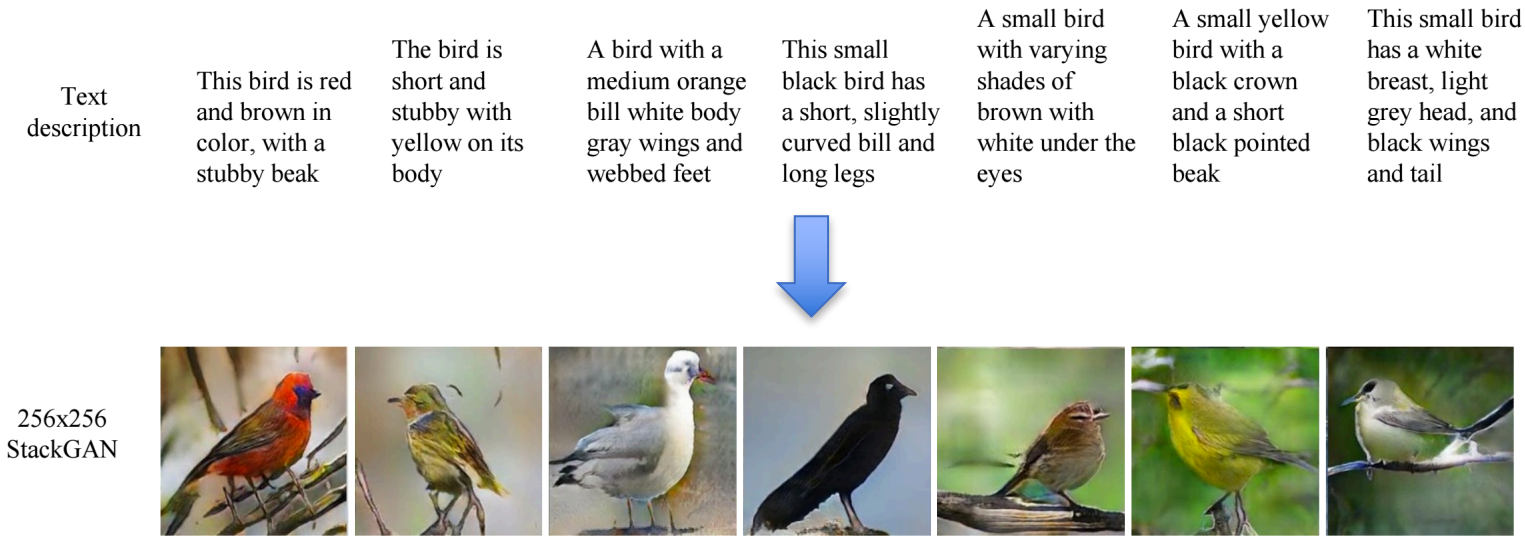


Figure from (Zhang et al., 2016)

Trained on data set of birds and captions, but again, no images just like this in the training set