PGM foundations - Part 2

Priors, Generative processes and Mixture models

Francisco Pereira

Filipe Rodrigues

DTU Management Engineering
Department of Management Engineering

Outline

- PGMs in continuous domain
- Generative processes
- Mixture models
- Summary: The big picture so far

Learning objectives

At the end of this lecture, you should be able to:

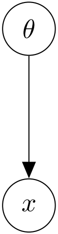
- Understand the concept of continuous random variable, and its specification in a PGM
- Understand the role of the prior, the importance of its form, and the concept of conjugate prior in inference
- Apply the generative process principles in the creation of a PGM and perform ancestral sampling with it
- Understand the concept mixture model, its representation, and inference challenges

PGM in continuous domain

- Thus far, we've been using only discrete variables
- Conditional Probability Tables
- Extension to continuous domain is intuitive...
- But with it, some concepts become more relevant
    - Prior
    - Conjugate prior

PGMs in continuous domain

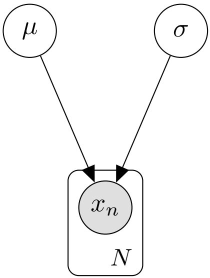
- General form



- We use functions instead of tables
- Typically, each function is a well-known distribution (or combination of them)
- Every distribution is parameterized by a set θ

PGMs in continuous domain

- Gaussian distribution



- A well-known example is the Gaussian (or *Normal)* distribution
- In this PGM, we assume to have observations x_n, that follow a Gaussian distribution
- It has two parameters (mean μ, variance σ^2)
- Inference
  - It has a well-known likelihood function

$$L(\mu, \sigma) = \prod_{i}^{N} \frac{1}{\sqrt{2\pi}\sigma} e^{\left(-\frac{(x_i-\mu)^2}{2\sigma^2}\right)}$$

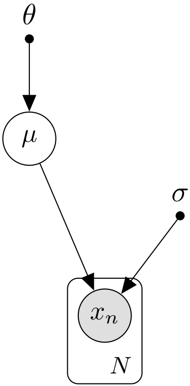
  - Corresponding log version

$$LL(\mu, \sigma) = -\frac{N}{2}(\log(2\pi) + \log(\sigma^2)) - \frac{1}{2\sigma^2}\sum_{i}(x_i - \mu)^2$$

PGMs in continuous domain

- A Graphical Model allows for a full Bayesian treatment
  - We can assign *priors* to the parameters
  - We can use domain knowledge
  - Good to prevent overfitting
  - What would be the form of those priors?

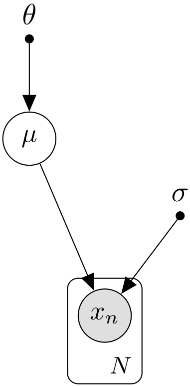
Gaussian distribution case



- To simplify, let's assume we know σ but not μ
- Can we pick *any* distribution, $D(\mu|\theta)$?
- Our joint distribution would become:

$$p(\mu, \mathbf{x}|\theta, \sigma) = D(\mu|\theta) \prod_{n=1}^{N} p(x_n|\mu, \sigma)$$

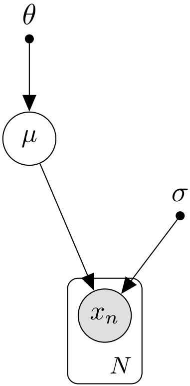
Gaussian distribution case



- To simplify, let's assume we know σ but not μ
- Can we pick *any* distribution, $D(\mu|\theta)$?
- **Common simplification** to unclutter notation:

$$p(\mu, \mathbf{x}|\theta, \sigma) = D(\mu|\theta) \prod_{n=1}^{N} p(x_n|\mu, \sigma)$$

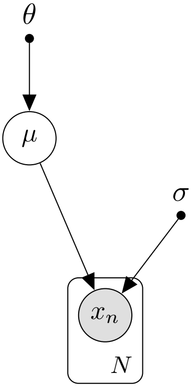
Gaussian distribution case



- To simplify, let's assume we know σ but not μ
- Can we pick *any* distribution, $D(\mu|\theta)$?
- **Common simplification** to unclutter notation:

$$p(\mu, \mathbf{x}) = D(\mu|\theta) \prod_{n=1}^{N} p(x_n|\mu, \sigma)$$

Gaussian distribution case

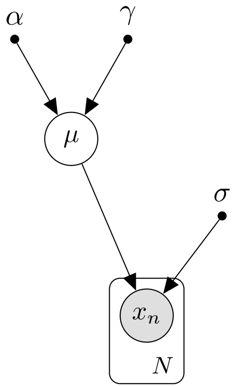


- To simplify, let's assume we know σ but not μ
- Can we pick *any* distribution, $D(\mu|\theta)$?
- Our joint distribution would become:

$$p(\mu, \mathbf{x}) = D(\mu|\theta) \prod_{n=1}^{N} p(x_n|\mu, \sigma)$$

- If $D(\mu|\theta)$ is normal, then $p(\mu, \mathbf{x})$ is normal too!

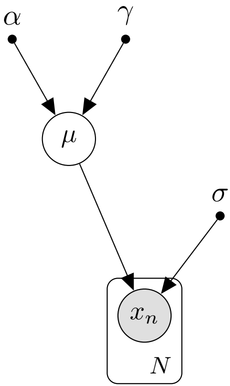
Gaussian distribution case



- If $D(\mu|\theta)$ is normal, then $p(\mu, \mathbf{x})$ is normal too!

$$D(\mu|\theta) = \mathcal{N}(\mu|\alpha, \gamma)$$

Gaussian distribution case



- If $D(\mu|\theta)$ is normal, then $p(\mu, \mathbf{x})$ is normal too!

$$D(\mu|\theta) = \mathcal{N}(\mu|\alpha, \gamma)$$

- the log probability of our PGM would be:

$$\begin{aligned} LL(\mu, \alpha, \gamma, \sigma) = -\frac{N}{2}(\log(2\pi) + \log(\sigma^2)) \\ -\frac{1}{2\sigma^2}\sum_i (x_i - \mu)^2 \\ -\frac{\log(2\pi)}{2} - \frac{\log(\gamma^2)}{2} - \frac{(\alpha - \mu)^2}{2\gamma^2} \end{aligned}$$

Playtime!

- Open notebook "3-PGM fundamentals.ipynb"
- Do part 1 (est. duration=30 min)

Conjugate priors

- For many known distributions, there is a corresponding *conjugate prior*, P, that preserves its form under multiplication. I.e., if we have distribution L and its conjugate prior P_0, we should have

$$P_1 = L \times P_0$$

- where P_1 has the same form as P_0
- For example, the Beta distribution is the conjugate prior of Bernoulli; and we've seen that the Normal is the conjugate for the mean of the Normal (when variance is known).
- If we have a known closed form for model, inference is generally more efficient!
- **This is great for online learning (why?)!**

Conjugate priors

- We usually use a table

Discrete distributions [edit]

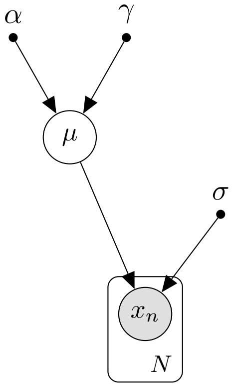
Likelihood	Model parameters	Conjugate prior distribution	Prior hyperparameters	Posterior hyperparameters	Interpretation of hyperparameters[note 1]	Posterior predictive[note 2]
Bernoulli	*p* (probability)	Beta	$\alpha,\ \beta$	$\alpha + \sum_{i=1}^n x_i,\ \beta + n - \sum_{i=1}^n x_i$	$\alpha - 1$ successes, $\beta - 1$ failures[note 1]	$p(\tilde{x} = 1) = \frac{\alpha'}{\alpha' + \beta'}$
Binomial	*p* (probability)	Beta	$\alpha,\ \beta$	$\alpha + \sum_{i=1}^n x_i,\ \beta + \sum_{i=1}^n N_i - \sum_{i=1}^n x_i$	$\alpha - 1$ successes, $\beta - 1$ failures[note 1]	$\operatorname{BetaBin}(\tilde{x}\|\alpha', \beta')$ (beta-binomial)
Negative binomial with known failure number, *r*	*p* (probability)	Beta	$\alpha,\ \beta$	$\alpha + \sum_{i=1}^n x_i,\ \beta + rn$	$\alpha - 1$ total successes, $\beta - 1$ failures[note 1] (i.e., $\frac{\beta - 1}{r}$ experiments, assuming r stays fixed)	
Poisson	λ (rate)	Gamma	$k,\ \theta$	$k + \sum_{i=1}^n x_i,\ \frac{\theta}{n\theta + 1}$	k total occurrences in $\frac{1}{\theta}$ intervals	$\operatorname{NB}(\tilde{x}\|k', \theta')$ (negative binomial)
			$\alpha,\ \beta$[note 3]	$\alpha + \sum_{i=1}^n x_i,\ \beta + n$	α total occurrences in β intervals	$\operatorname{NB}\left(\tilde{x}\|\alpha', \frac{1}{1+\beta'}\right)$ (negative binomial)
Categorical	***p*** (probability vector), *k* (number of categories; i.e., size of ***p***)	Dirichlet	$\boldsymbol{\alpha}$	$\boldsymbol{\alpha} + (c_1, \ldots, c_k)$, where c_i is the number of observations in category *i*	$\alpha_i - 1$ occurrences of category *i*[note 1]	$p(\tilde{x} = i) = \frac{\alpha_i'}{\sum_i \alpha_i'} = \frac{\alpha_i + c_i}{\sum_i \alpha_i + n}$

Figure: From Wikipedia

Some conjugate priors to remember...

Likelihood	**Prior**
Normal with known variance	Normal
Normal with known mean	Inverse Gamma
Multivariate normal, known mean	Inverse Wishart
Multivariate normal, unknown mean and variance	Normal-inverse-Wishart
Exponential	Gamma
Bernoulli	Beta
Mulitnomial	Dirichlet
Poisson	Gamma

Gaussian distribution case



- For our Gaussian example, the posterior $p(\mu|\mathbf{X}) = \mathcal{N}(\tilde{\alpha}, \tilde{\gamma})$ will be directly

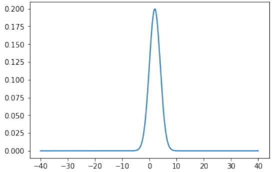
$$\tilde{\alpha} = \frac{1}{\gamma^{-2} + \frac{N}{\sigma^2}} \left(\frac{\alpha}{\gamma^2} + \frac{\sum_{i=1}^{N} x_i}{\sigma^2} \right)$$

$$\tilde{\gamma} = \sqrt{\left(\gamma^{-2} + \frac{N}{\sigma^2} \right)^{-1}}$$

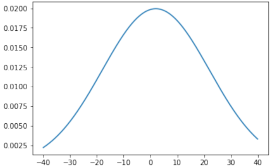
- We just followed the conjugate priors table
- Calculation in constant time, no need to optimize anything!
- We could use this as the next prior!...
- BUT if $p(\mu, \mathbf{x})$ is not a known distribution, we may have trouble deriving it (analytically)...

Last note on priors

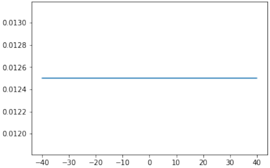
- Depending on what you know of the problem (or the constraints you want to impose...):



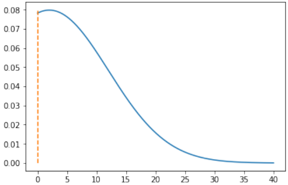
Proper informative prior



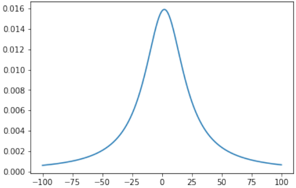
Proper weakly informative prior



Inproper uninformative prior



Proper bounded prior



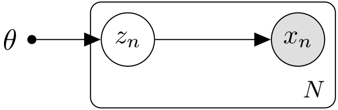
Proper uninformative prior (notice scale and fat tails)

Generative processes

- By now, you understand that you can combine variables in multiple ways in your graphical model
- On the other hand, you may be overwhelmed about where to start doing your own
  - Small models, with few variables, are simple
  - What if you have a lot of variables, assumptions, domain knowledge?...
- You need to think from a generative perspective...

"Generative story" of data

- How is a data point generated?



- Given a parameter θ
- For $n = 1..N$, do

  1. Draw a random latent variable, $z_n \sim p(z|\theta)$
  2. Given z_n, generate x_n such that $x_n \sim p(x|\theta, z_n)$

- In fact, this resembles a program structure!

A more complex example - Dwell time prediction

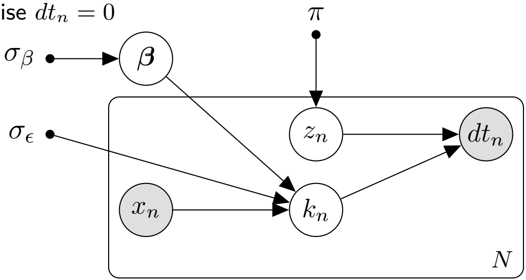
For a given bus stop, that serves a single line, can we predict the amount of time the next bus will be stopped there to load/unload passengers (the *dwell* time)?

- Our dataset contains $\{x_n = \{0, 1\}$-representing peak/non-peak hour, dt_n - dwell time$\}$.
- Notice that, sometimes, the bus does not stop at all!
- When it stops, we measure the duration as dt
- When it doesn't stop, $dt = 0$

Dwell time prediction

Given N, σ_β, σ_ϵ and π

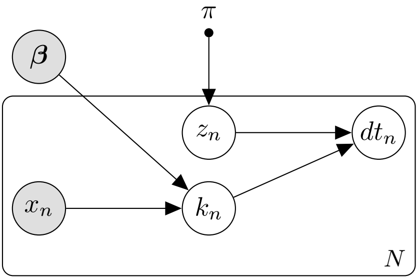
1. Draw a pair of parameters[1], $\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{0}, I\sigma_\beta)$
2. For $n = 1..N$
   1. Draw one value for z_n, such that $z_n \sim Bern(\pi)$.
      - If $z_n = 1$, the bus has stopped ($z_n = 0$ otherwise).
      - Distributed as Bernoulli, with parameter π
   2. Draw one value for k_n, such that $k_n \sim \mathcal{N}(\beta_0 + \beta_1 x_n, \sigma_\epsilon)$
   3. If $z_n = 1$, $dt_n = k_n$,
      - otherwise $dt_n = 0$



[1] We need two values for β, one for the intercept, another for the peak/non-peak information.

Dwell time prediction

- After you define your model, you need to estimate it. I.e. infer the following:
  - Distribution of $\boldsymbol{\beta}$
  - Optimal values of σ_ϵ, σ_β, and π (we defined them as constants!)
- Of course, when you have them, you can make your predictions!
- Your model will look different:



"Generative story" of data

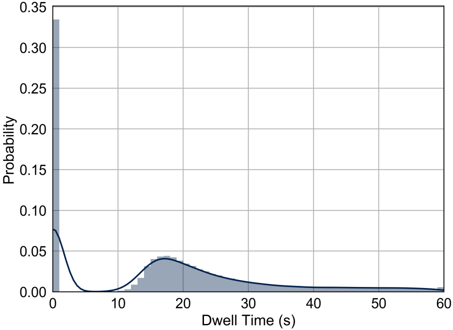
- Set up the building blocks, as per available knowledge
- Easy to change data distributions inside the model
- Can be used to *actually* generate data!
  - Ancestral sampling
  - Do *prior predictive checks*!

Playtime!

- Open notebook "3-PGM fundamentals.ipynb"
- Do part 2 (est. duration=30 min)

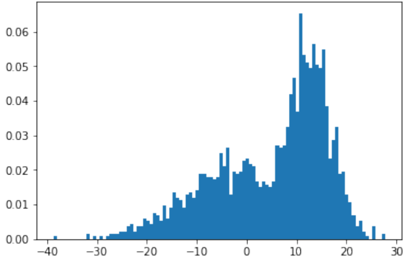
Mixture models

- A PGM is composed of observed and latent variables, parameters, constants.
- In this course, we'll approach some examples from this very large family
- Mixture models are pervasive in data modelling in general
- Problem:
  - Sub-populations of data
  - Data generated from combination/competition of multiple sources
  - Number of sources usually discrete and finite

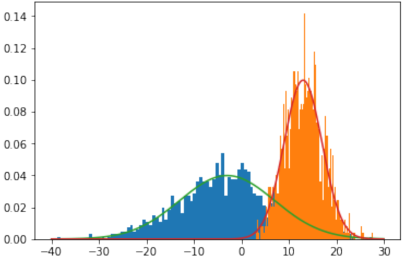


The canonical example: Gaussian Mixture

- What we observe



- What really happens

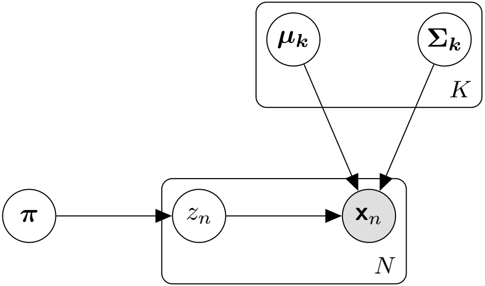


Generative story

Given:

- A dataset with N points (or vectors) $(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n)$ and a value K

1. Draw $\boldsymbol{\pi}$, and $(\boldsymbol{\mu_k}, \boldsymbol{\Sigma_k})$ for all K gaussians
2. For $n = 1, 2, ..., N$
    1. Draw $z_n \sim Multinomial(\boldsymbol{\pi})$
       where $\boldsymbol{\pi}$ is a vector $(1 \times K)$ with the probabilities of each class
    2. Define $k = z_n$. Generate $\mathbf{x}_n$, from the k-th Gaussian,
       $\mathbf{x}_n \sim \mathcal{N}(\boldsymbol{\mu_k}, \boldsymbol{\Sigma_k})$



Generative story

Given:

- A dataset with N points (or vectors) $(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n)$ and a value K

1. Draw $\boldsymbol{\pi}$, and $(\boldsymbol{\mu_k}, \boldsymbol{\Sigma_k})$ for all K gaussians
2. For $n = 1, 2, ..., N$
    1. Draw $z_n \sim Multinomial(\boldsymbol{\pi})$
       where $\boldsymbol{\pi}$ is a vector $(1 \times K)$ with the probabilities of each class
    2. Define $k = z_n$. Generate $\mathbf{x}_n$, from the k-th Gaussian,
       $\mathbf{x}_n \sim \mathcal{N}(\boldsymbol{\mu_k}, \boldsymbol{\Sigma_k})$

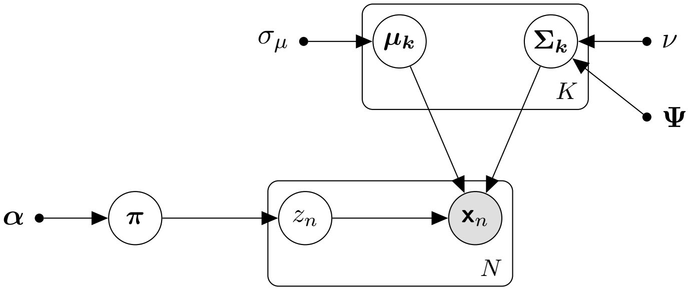
Factorization:

$$p(\boldsymbol{\pi}) \left(\prod_{k=1}^{K} p(\boldsymbol{\mu_k}) p(\boldsymbol{\Sigma_k}) \right) \prod_{n=1}^{N} \sum_{k=1}^{K} p(z_n = k | \boldsymbol{\pi}) p(\mathbf{x}_n | \boldsymbol{\mu_k}, \boldsymbol{\Sigma_k})$$

Note: in practice we need to be exhaustive

...particularly in probabilistic programming (e.g. STAN)

- $\boldsymbol{\pi} \sim Dir(\boldsymbol{\alpha})$
- $\boldsymbol{\mu_k} \sim \mathcal{N}(\mathbf{0}, I\sigma_\mu)$
- $\boldsymbol{\Sigma_k} \sim \mathcal{W}^{-1}(\boldsymbol{\Psi}, \nu)$
    - Typically, $\nu =$ degrees of freedom (typically number of dimensions of $\mathbf{x}$), and $\boldsymbol{\Psi} = I$



Note: in practice we need to be exhaustive

...particularly in probabilistic programming (e.g. STAN)

- $\boldsymbol{\pi} \sim Dir(\boldsymbol{\alpha})$
- $\boldsymbol{\mu_k} \sim \mathcal{N}(\mathbf{0}, I\sigma_\mu)$
- $\boldsymbol{\Sigma_k} \sim \mathcal{W}^{-1}(\boldsymbol{\Psi}, \nu)$
  - Typically, $\nu =$ degrees of freedom (typically number of dimensions of $\mathbf{x}$), and $\boldsymbol{\Psi} = I$

The factorization becomes:

$$Dir(\boldsymbol{\pi}|\boldsymbol{\alpha})\left(\prod_{k=1}^{K} \mathcal{N}(\boldsymbol{\mu_k}|\mathbf{0}, I\sigma_\mu)\mathcal{W}^{-1}(\boldsymbol{\Sigma_k}|\boldsymbol{\Psi}, \nu)\right)\prod_{n=1}^{N}\sum_{k=1}^{K}\pi_k\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu_k}, \boldsymbol{\Sigma_k})$$

In log format:

$$ln(Dir(\boldsymbol{\pi}|\boldsymbol{\alpha}))+\sum_{k=1}^{K}\left(ln\mathcal{N}(\boldsymbol{\mu_k}|\mathbf{0}, I\sigma_\mu)+ln\mathcal{W}^{-1}(\boldsymbol{\Sigma_k}|\boldsymbol{\Psi}, \nu)\right)+\sum_{n=1}^{N} ln\left(\sum_{k=1}^{K}\pi_k\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu_k}, \boldsymbol{\Sigma_k})\right)$$

Playtime!

- Open notebook "3-PGM fundamentals.ipynb"
- Do part 3 (est. duration=45 min)

The big picture so far

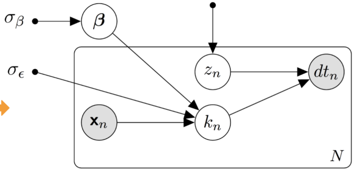
- Probability and statistics recap
  - Probability theory at the center of everything that we do
  - Allows to capture uncertainty
- Probabilistic graphical models (PGMs)
  - Intuitive and compact way of representing the structure of a prob. model
  - Relationships between variables and conditional independencies
  - How the joint distribution factorizes
- Generative processes
  - A "story" of how the observed data was generated
  - Explicit description of how the different variables in the model are related
  - Complementary to PGM representation: more detailed, but less intuitive
- Joint probability distribution and Bayesian inference
  - Joint probability of the model: central object for all computations
  - Bayesian inference: model $+$ data $\rightarrow$ patterns
  - Important concepts: likelihood, prior, posterior, conjugate prior, etc.

Step back: The big picture so far

- Everything is related...

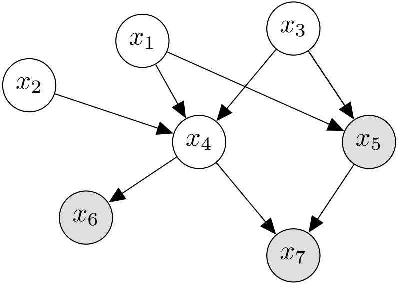
$$p(\boldsymbol{\beta}, \mathbf{z}, \mathbf{k}, \mathbf{dt}) =$$
$$p(\boldsymbol{\beta}|\sigma_\beta) \prod_{n=1}^{N} p(k_n|\mathbf{x}_n, \boldsymbol{\beta}, \sigma_\epsilon)\, p(z_n|\pi)\, p(dt_n|z_n, k_n)$$

1. Draw a pair of parameters[1], $\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{0}, I\sigma_\beta)$
2. For $n = 1..N$
    1. Draw one value for z_n, such that $z_n \sim Bern(\pi)$.
        - If $z_n = 1$, the bus has stopped ($z_n = 0$ otherwise)
        - Distributed as Bernoulli, with parameter π
    2. Draw one value for k_n, such that $k_n \sim \mathcal{N}(\mathbf{x}_n^T \boldsymbol{\beta}, \sigma_\epsilon)$
    3. If $z_n = 1$, $dt_n = k_n$,
        - otherwise $dt_n = 0$



The problem of inference

- **Model + Data $\rightarrow$ Insights**
- Answer various types of questions about the data by computing the posterior distribution of the latent variables given the observed ones



- Example: $p(x_2|x_5, x_6, x_7) = ?$

The problem of inference

- Inference in general: given a set of latent variables $\mathbf{z} = \{z_m\}_{m=1}^{M}$ and observed variables $\mathbf{x} = \{x_n\}_{n=1}^{N}$, compute $p(\mathbf{z}|\mathbf{x})$
- Two classes of approaches:
  - Exact inference (Bayes' theorem)

$$\overbrace{p(\mathbf{z}|\mathbf{x})}^{\text{posterior}} = \frac{\overbrace{p(\mathbf{x},\mathbf{z})}^{\text{joint}}}{p(\mathbf{x})} = \frac{\overbrace{p(\mathbf{x}|\mathbf{z})}^{\text{likelihood}}\ \overbrace{p(\mathbf{z})}^{\text{prior}}}{\underbrace{p(\mathbf{x})}_{\text{evidence}}}$$

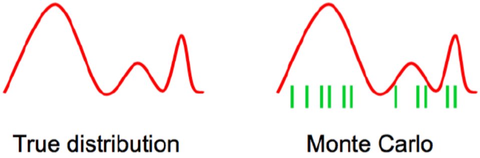
    - For most problems of interest, it is often infeasible to evaluate posterior exactly or to compute expectations with respect to it
  - Approximate Inference
    - STAN uses approximate inference!
    - Stochastic vs. variational methods

Approximate Inference

- Stochastic
- Variational

Approximate Inference

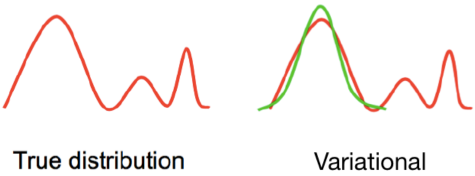
- Stochastic
  - We try to sample from the posterior distribution
  - Samples provide approximate representation of the true posterior
  - We can use samples to compute expectations w.r.t. the posterior
  - Example: Markov Chain Monte Carlo (MCMC) methods



- Variational

Approximate Inference

- Stochastic
- Variational
  - Approximate intractable distribution with a simpler, tractable one
  - Goal: find the parameters of the simpler distribution that make it as similar as possible to the true distribution
  - Similar in what sense?
    - E.g. using Kullback-Leibler (KL) divergence
  - Becomes an optimization problem (of minimizing the difference between *true* and *approximate* distribution)



Approximate Inference

- Stochastic
- Variational

- STAN can use:
  - MCMC (Hamiltonian Monte Carlo or NUTS)
  - Automatic Differentiation Variational Inference (ADVI) - a variational approach with a stochastic component...

References

- **Main reading**: Chapter 8.1 “Bayesian Networks”, pages 363-366, and Chapter 9.2: “Mixture Models and EM”, pages 430-435 of Chris Bishop’s book, “Pattern Recognition and Machine Learning” (PRML) URL: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/05/prml-web-sol-2009-09-08.pdf)
- More on Mixture Models: Chapter 11: “Mixture models and the EM algorithm”, pages 337-345 of Kevin Murphy’s book “Machine Learning: A Probabilistic Perspective”
- (Koller and Friedman, 2009) Koller, D., and Friedman, N. Probabilistic graphical models: principles and techniques. MIT press. (2009).