PGM foundations Part II

Representation in continuous domain, approximate inference

Francisco Pereira

Filipe Rodrigues

DTU Management Engineering
Department of Management Engineering

Outline

- PGMs in continuous domain
- Generative approach
- Approximate inference

PGM in continuous domain

- Thus far, we've been using only discrete variables
- Conditional Probability Tables
- Extension to continuous domain is almost trivial...

PGM in continuous domain

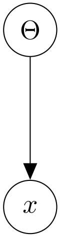
- Thus far, we've been using only discrete variables
- Conditional Probability Tables
- Extension to continuous domain is almost trivial...
- But with it, some concepts become more relevant

PGM in continuous domain

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

PGMs in continuous domain

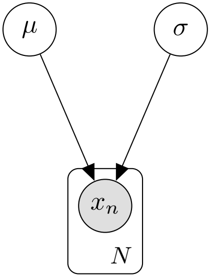
- General form



- 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

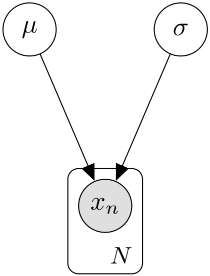
- Guassian distribution



- 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)

PGMs in continuous domain

- Guassian 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 log likelihood function

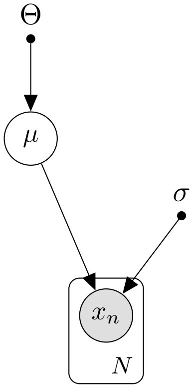
PGMs in continuous domain

- 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

PGMs in continuous domain

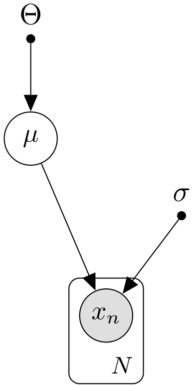
- A Graphical Model allows for a full Bayesian treatment:
  - We can assign *priors* to the parameters
  - We can use domain knowledge
  - Good to prevent overfitting
  - What would be the form of those priors?

Gaussian distribution case



- To simplify, let's assume we know σ but not μ

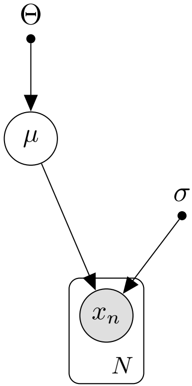
Gaussian distribution case



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

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

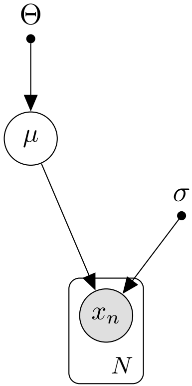
Gaussian distribution case



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

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

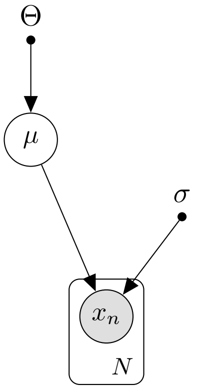
Gaussian distribution case



- 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

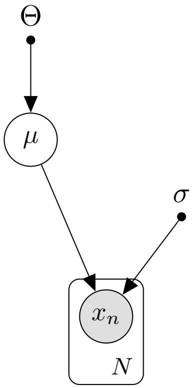


- To simplify, let's assume we know σ but not μ
- Can we pick *any* distribution, $D(\mu|\Theta)$?
- Our joint distribution would become:

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

  - If $D(\mu|\Theta)$ is normal, then $p(\mu, \mathbf{X})$ is normal too!
- If $p(\mu, \mathbf{X})$ is not a known distribution, we may have trouble deriving it

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!
- If $p(\mu, \mathbf{X})$ is not a known distribution, we may have trouble deriving it (analytically)...

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!

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]	$\mathrm{BetaBin}(\tilde{x} \mid \alpha', \beta')$ (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	$\mathrm{NB}(\tilde{x} \mid k', \theta')$ (negative binomial)
			$\alpha,\, \beta$[note 3]	$\alpha + \sum_{i=1}^{n} x_i,\, \beta + n$	α total occurrences in β intervals	$\mathrm{NB}\left(\tilde{x} \mid \alpha', \frac{1}{1 + \beta'}\right)$ (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]	$p(\tilde{x} = i) = \frac{\alpha_i{}'}{\sum_i \alpha_i{}'} = \frac{\alpha_i + c_i}{\sum_i \alpha_i + n}$

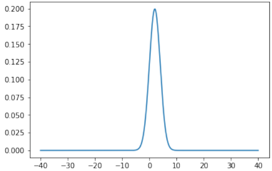
Figure: From Wikipedia

Some conjugate priors to remember...

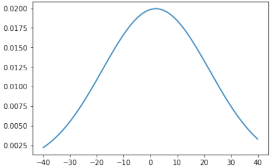
Likelihood	Prior
Normal with known variance	Normal
Normal with known mean	Inverse Gamma
Multivariate normal, known mean	Inverse Wishart
Multivariate normal, unknown mean and variance	Normal-inverse-Wishart
Exponential	Gamma
Bernoulli	Beta
Mulitnomial	Dirichlet
Poisson	Gamma

Last note on priors

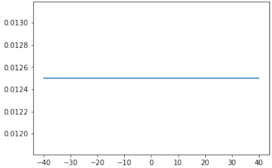
- Depending on what you know of the problem (or the constraints you want to impose...):



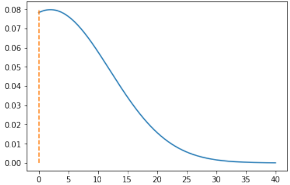
Proper informative prior



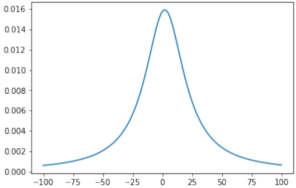
Proper weakly informative prior



Inproper uninformative prior



Proper bounded prior



Proper uninformative prior (notice scale and fat tails)

Playtime!

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

Generative approach

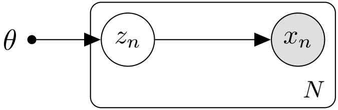
- By now, you understand that you can combine variables in multiple ways in your graphical model
- On the other hand, you may be overwhelmed about where to start doing your own
    - Small models, with few variables, are simple
    - What if you have a lot of variables, assumptions, domain knowledge?...

Generative approach

- 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...

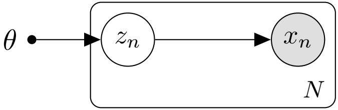
"Generative story" of data

- How is a data point generated?



"Generative story" of data

- How is a data point generated?

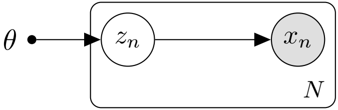


- Given a parameter θ
- For $n = 1..N$, do

  1. Draw a random latent variable, $z_n \sim p(z|\theta)$

"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

Dwell time prediction

Given N, σ_β, σ_ϵ and π

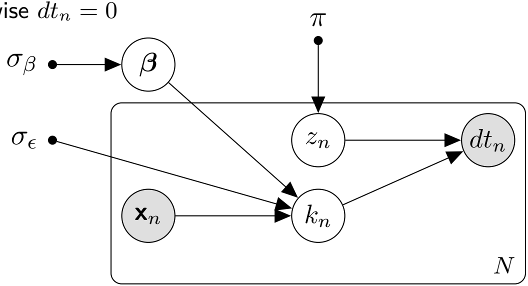
1. Draw a pair of parameters[1], $\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{0}, I\sigma_\beta)$
2. For $n = 1..N$
   1. Draw one value for z_n, such that $z_n \sim Bern(\pi)$.
      - If $z_n = 1$, the bus has stopped ($z_n = 0$ otherwise).
      - Distributed as Bernoulli, with parameter π
   2. Draw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

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 $k_n \sim \mathcal{N}(\mathbf{x}_n^T \boldsymbol{\beta}, \sigma_\epsilon)$
   3. If $z_n = 1$, $dt_n = k_n$,
      - otherwise $dt_n = 0$



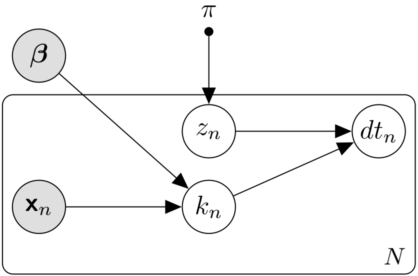
[1] We need two values for β, one for the intercept, another for the peak/non-peak information.

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:

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

Playtime!

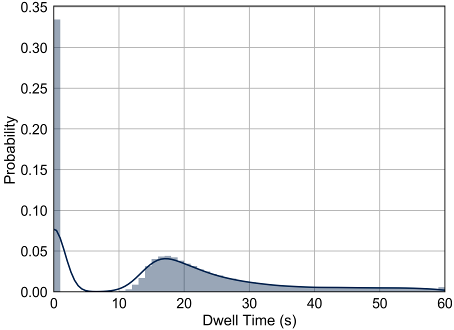
- Open notebook "3-PGM fundamentals.ipynb"
- Do part 2 (est. duration=30 min)

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

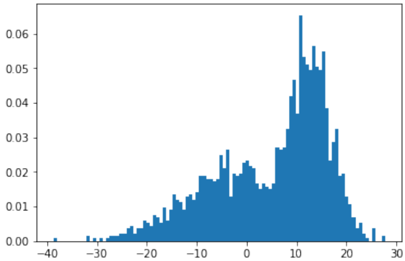
Mixture models

- 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



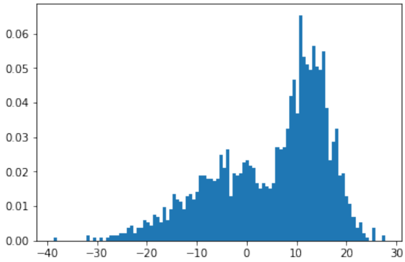
The canonical example: Gaussian Mixture

- What we observe

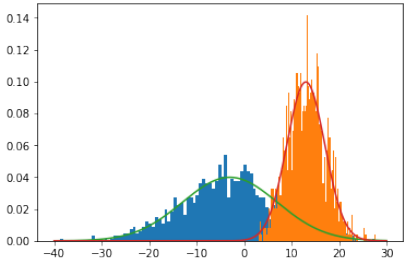


The canonical example: Gaussian Mixture

- 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 $z_n \sim Multinomial(\boldsymbol{\pi})$
        - $\boldsymbol{\pi}$ 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,

$$\mathbf{x}_n \sim \mathcal{N}(\boldsymbol{\mu_k}, \boldsymbol{\Sigma_k})$$

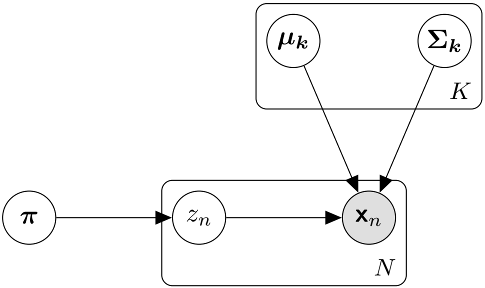
Generative story

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 $z_n \sim Multinomial(\boldsymbol{\pi})$
      - $\boldsymbol{\pi}$ 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,

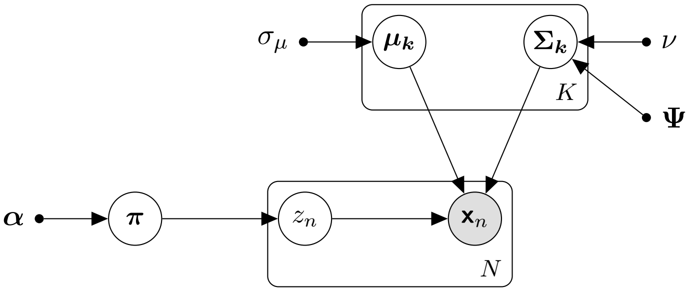
$$\mathbf{x}_n \sim \mathcal{N}(\boldsymbol{\mu_k}, \boldsymbol{\Sigma_k})$$



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, $\nu =$ number of dimensions, and $\boldsymbol{\Psi} = I$



Playtime!

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

The problem of inference

- ...your last exercise should show that we need efficient inference methods
  - Complex distribution (e.g. involving log of sum; an unknown form; etc.)
  - High dimensionality (e.g. more than a couple of parameters is often too many!)
  - Continuous dimensions instead of discrete

The problem of inference

- ...your last exercise should show that we need efficient inference methods
  - Complex distribution (e.g. involving log of sum; an unknown form; etc.)
  - High dimensionality (e.g. more than a couple of parameters is often too many!)
  - Continuous dimensions instead of discrete
- Two general approaches:
  - Exact inference
  - Approximate Inference

The problem of inference

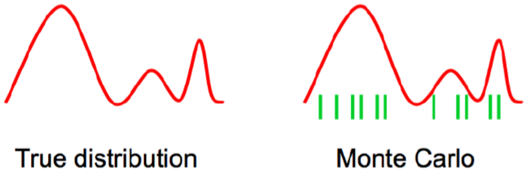
- ...your last exercise should show that we need efficient inference methods
  - Complex distribution (e.g. involving log of sum; an unknown form; etc.)
  - High dimensionality (e.g. more than a couple of parameters is often too many!)
  - Continuous dimensions instead of discrete
- Two general approaches:
  - Exact inference
  - Approximate Inference
- Before we get practical (i.e. STAN), we need to understand a bit how inference can be done
  - Important to manipulate STAN and understand its output
- STAN uses Approximate Inference (we'll talk about it today)
- In a later class, we'll get more detailed (in both Exact and Approx.).

Approximate Inference

- Stochastic
- Variational

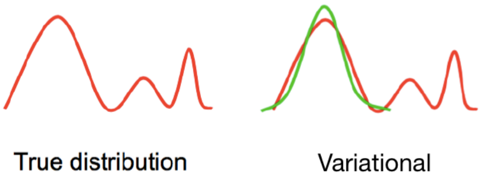
Approximate Inference

- Stochastic
    - We sample from the distribution
    - Markov Chain Monte Carlo (MCMC)
- Variational



Approximate Inference

- Stochastic
- Variational
  - We look for a *simpler but similar* distribution
  - Becomes an optimization problem (of minimizing the difference between *true* and *approximate* distribution)



Approximate Inference

- Stochastic
- Variational

- STAN uses
  - MCMC
  - Automatic Differentiation Variational Inference (ADVI) - a combination of variational and stochastic...

Intuition on Markov Chain Monte Carlo (MCMC)

- Major challenge: how do we choose the sample points?
- Don't forget that "one sample" means:
  - Assignment of a value to each variable of the joint distribution
  - Calculation of probability of this vector (just use formula)

Intuition on Markov Chain Monte Carlo (MCMC)

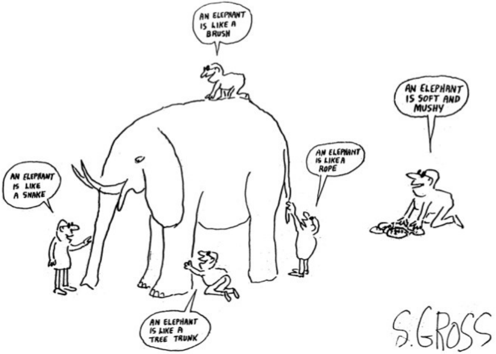
- Major challenge: how do we choose the sample points?
- Don't forget that "one sample" means:
  - Assignment of a value to each variable of the joint distribution
  - Calculation of probability of this vector (just use formula)
- Don't forget that we're either looking for:
  - the Maximum a Posteriori (the point/vector with highest probability)
  - the whole posterior distribution

Intuition on Markov Chain Monte Carlo (MCMC)

- Major challenge: how do we choose the sample points?
- Don't forget that "one sample" means:
  - Assignment of a value to each variable of the joint distribution
  - Calculation of probability of this vector (just use formula)
- Don't forget that we're either looking for:
  - the Maximum a Posteriori (the point/vector with highest probability)
  - the whole posterior distribution
- With a good number of points, we can:
  - Obtain approximate statistics for the distribution
  - Obtain estimates for individual parameters

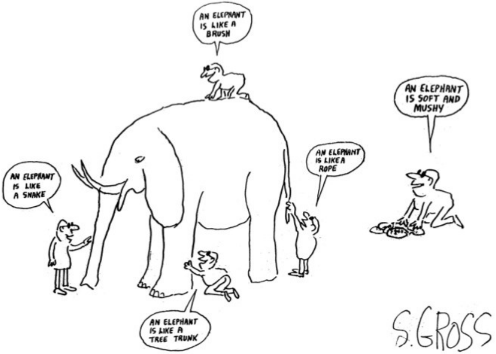
Intuition on MCMC

- Major challenge: how do we choose the sample points?



Intuition on MCMC

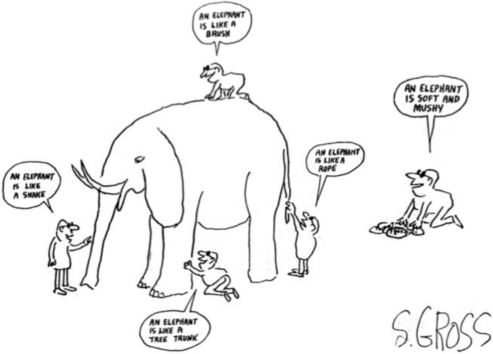
- Major challenge: how do we choose the sample points?



- Option 1: Just uniformly.

Intuition on MCMC

- Major challenge: how do we choose the sample points?



- Option 1: Just uniformly.
- Option 2: Using the true distribution (cleverly... ;-))

Intuition on MCMC

Better with an example: Gibbs sampling

- Our mixture model exercise. We want to estimate values for parameters π, $\boldsymbol{\mu} = [\mu_1, \mu_2]^T$ having the expression for $p(\pi, \boldsymbol{\mu}|\sigma, \sigma_\mu, \alpha)$
- Gibbs sampling for our Gaussian Mixture exercise[2].
  1. Initialize π, $\boldsymbol{\mu}$ with random values (from their priors). Let's call them $\pi^{(0)}$, $\boldsymbol{\mu}^{(0)}$
  2. For $t = 1...T$, do
     1. Choose new value $\pi^t \sim p(\pi|\boldsymbol{\mu}^{t-1})$.
     2. Choose new value $\boldsymbol{\mu}^t \sim p(\boldsymbol{\mu}|\pi^t)$

[2] we're dropping the full notation to better give the intuition. Notice all variables other than π and $\boldsymbol{\mu}$ are fixed anyway. In other words, we're estimating $p(\pi, \boldsymbol{\mu})$, but we mean $p(\pi, \boldsymbol{\mu}|\sigma, \sigma_\mu, \alpha)$.

Intuition on MCMC

Better with an example: Gibbs sampling

- Our mixture model exercise. We want to estimate values for parameters π, $\boldsymbol{\mu} = [\mu_1, \mu_2]^T$ having the expression for $p(\pi, \boldsymbol{\mu}|\sigma, \sigma_{\mu}, \alpha)$
- Gibbs sampling for our Gaussian Mixture exercise[2].
  1. Initialize π, $\boldsymbol{\mu}$ with random values (from their priors). Let's call them $\pi^{(0)}$, $\boldsymbol{\mu}^{(0)}$
  2. For $t = 1...T$, do
     1. Choose new value $\pi^t \sim p(\pi|\boldsymbol{\mu}^{t-1})$.
     2. Choose new value $\boldsymbol{\mu}^t \sim p(\boldsymbol{\mu}|\pi^t)$
- With T sufficiently large, we get enough points to estimate what we want! :-)

[2] we're dropping the full notation to better give the intuition. Notice all variables other than π and $\boldsymbol{\mu}$ are fixed anyway. In other words, we're estimating $p(\pi, \boldsymbol{\mu})$, but we mean $p(\pi, \boldsymbol{\mu}|\sigma, \sigma_{\mu}, \alpha)$.

Intuition on MCMC

Better with an example: Gibbs sampling

- We want to approximate $p(\mathbf{x})$, where $\mathbf{x} = [x_1, x_2, ..., x_k]$ and $q(\mathbf{x})$ is a prior distribution for $\mathbf{x}$
- Generic Gibbs sampling algorithm:
  1. Initialize $\mathbf{x} \sim q(\mathbf{x})$
  2. For $t = 1...T$, do
     1. $x_1^t \sim p(x_1 | x_2^{t-1}, x_3^{t-1}, ..., x_k^{t-1})$
     2. $x_2^t \sim p(x_2 | x_1^t, \ , x_3^{t-1}, ..., x_k^{t-1})$

        $\vdots$
     3. $x_k^t \sim p(x_k | x_1^t, x_2^t, x_3^t, ..., x_{k-1}^t)$

Intuition on MCMC

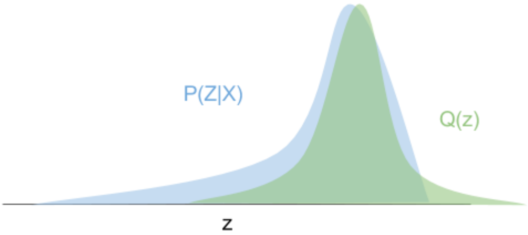
Better with an example: Gibbs sampling

- We want to approximate $p(\mathbf{x})$, where $\mathbf{x} = [x_1, x_2, ..., x_k]$ and $q(\mathbf{x})$ is a prior distribution for $\mathbf{x}$
- Generic Gibbs sampling algorithm:
  1. Initialize $\mathbf{x} \sim q(\mathbf{x})$
  2. For $t = 1...T$, do
     1. $x_1^t \sim p(x_1 | x_2^{t-1}, x_3^{t-1}, ..., x_k^{t-1})$
     2. $x_2^t \sim p(x_2 | x_1^t, \ , x_3^{t-1}, ..., x_k^{t-1})$

        $\vdots$
     3. $x_k^t \sim p(x_k | x_1^t, x_2^t, x_3^t, ..., x_{k-1}^t)$
- In general MCMC algorithms all have this flavor
- STAN uses Hamiltonian Monte Carlo
- We'll get back to this later... ;-)

Intuition on Variational Inference

- Key idea: approximate intractable distribution with a simpler, tractable one.



- We use a method to compare the two distributions, called Kullback-Leibler (KL) divergence
- We turn into an optimization problem, of minimizing KL divergence
- We later use the simpler distribution, to make our inference in the model

Conclusions

- PGMs are extremely flexible. They can combine:
  - Discrete and continuous variables
  - Parametric and non-parametric models
  - Informative and non-informative priors
  - Online learning with conjugate priors
  - Partial and complete data

Conclusions

- PGMs are extremely flexible. They can combine:
  - Discrete and continuous variables
  - Parametric and non-parametric models
  - Informative and non-informative priors
  - Online learning with conjugate priors
  - Partial and complete data
- Think in a generative way helps design a model

Conclusions

- PGMs are extremely flexible. They can combine:
    - Discrete and continuous variables
    - Parametric and non-parametric models
    - Informative and non-informative priors
    - Online learning with conjugate priors
    - Partial and complete data
- Think in a generative way helps design a model
- The more complex the model is, the harder inference may be
- Markov Chain Monte Carlo and Variational Inference (exact inference later...)

References

- (Srihari, 2017) MCMC and Gibbs Sampling. Introduction to Machine Learning Course. Sargur Srihari. Department of Computer Science and Engineering, University at Buffalo. http://www.cedar.buffalo.edu/ srihari/CSE574/Chap11/Ch11.3-MCMCSampling.pdf
- (Salakhutdinov, R., 2011) Approximate Inference. 9.520: Statistical Learning Theory and Applications, Spring 2011. MIT. http://www.mit.edu/ 9.520/spring11/slides/class19_approxinf.pdf
- (Jiang, 2016) "A Beginner's Guide to Variational Methods: Mean-Field Approximation". Eric Jiang. Blog post. https://blog.evjang.com/2016/08/variational-bayes.html
- (Koller and Friedman, 2009) Koller, D., and Friedman, N. Probabilistic graphical models: principles and techniques. MIT press. (2009).