CPSC 540: Machine Learning

MCMC and Non-Parametric Bayes

Mark Schmidt

University of British Columbia

Winter 2016

Admin

- I went through project proposals:
  - Some of you got a message on Piazza.
  - No news is good news.
- A5 coming tomorrow.
- Project submission details coming next week.

Overview of Bayesian Inference Tasks

- In Bayesian approach, we typically work with the posterior

$$p(\theta|x) = \frac{1}{Z}p(x|\theta)p(\theta) = \frac{1}{Z}\tilde{p}(\theta),$$

where Z makes the distribution sum/integrate to 1.

Overview of Bayesian Inference Tasks

- In Bayesian approach, we typically work with the posterior

$$p(\theta|x) = \frac{1}{Z}p(x|\theta)p(\theta) = \frac{1}{Z}\tilde{p}(\theta),$$

  where Z makes the distribution sum/integrate to 1.
- Typically, we need to compute expectation of some f with respect to posterior,

$$E[f(\theta)] = \int_\theta f(\theta)p(\theta|x)d\theta.$$

Overview of Bayesian Inference Tasks

- In Bayesian approach, we typically work with the posterior

$$p(\theta|x) = \frac{1}{Z}p(x|\theta)p(\theta) = \frac{1}{Z}\tilde{p}(\theta),$$

  where Z makes the distribution sum/integrate to 1.
- Typically, we need to compute expectation of some f with respect to posterior,

$$E[f(\theta)] = \int_\theta f(\theta)p(\theta|x)d\theta.$$

- Examples:
  - If $f(\theta) = p(\tilde{x}|\theta)$, we get posterior predictive.
  - If $f(\theta) = 1$ and we use $\tilde{p}(\theta)$, we get marginal likelihood Z.
  - If $f(\theta) = \mathbb{I}(\theta \in S)$ we get probability of S (e.g., marginals or conditionals).

Last Time: Conjugate Prior and Monte Carlo Methods

- Last time we saw two ways to deal with this:
  1. Conjugate priors:
     - Apply when $p(x|\theta)$ is in the exponential family.
     - Set $p(\theta)$ to a conjugate prior, and posterior will have the same form.
     - Integrals will often have closed-form solutions, but restricted class of models.

Last Time: Conjugate Prior and Monte Carlo Methods

- Last time we saw two ways to deal with this:
  1. Conjugate priors:
     - Apply when $p(x|\theta)$ is in the exponential family.
     - Set $p(\theta)$ to a conjugate prior, and posterior will have the same form.
     - Integrals will often have closed-form solutions, but restricted class of models.
  2. Monte Carlo methods: sample θ^i from $p(\theta|x)$ and use:

$$\mathbb{E}[f(\theta)] = \int f(\theta)p(\theta)d\theta \approx \frac{1}{n}\sum_{i=1}^{n} f(\theta^i).$$

Last Time: Conjugate Prior and Monte Carlo Methods

- Last time we saw two ways to deal with this:
  1. Conjugate priors:
     - Apply when $p(x|\theta)$ is in the exponential family.
     - Set $p(\theta)$ to a conjugate prior, and posterior will have the same form.
     - Integrals will often have closed-form solutions, but restricted class of models.
  2. Monte Carlo methods: sample θ^i from $p(\theta|x)$ and use:

$$\mathbb{E}[f(\theta)] = \int f(\theta)p(\theta)d\theta \approx \frac{1}{n}\sum_{i=1}^{n} f(\theta^i).$$

- We discussed basic Monte Carlo methods:
  - Inverse CDF, ancestral sampling, rejection sampling, importance sampling.
  - Work well in low dimensions or for posteriors with analytic properties.

Limitations of Simple Monte Carlo Methods

- These methods tend not to work in complex situations:
  - Inverse CDF may not be avaiable.
  - Conditional needed for ancestral sampling may be hard to compute.
  - Rejection sampling tends to reject almost all samples.
  - Importance sampling tends gives almost zero weight to all samples.
- We want an algorithm that gets better over time.

Limitations of Simple Monte Carlo Methods

- These methods tend not to work in complex situations:
  - Inverse CDF may not be avaiable.
  - Conditional needed for ancestral sampling may be hard to compute.
  - Rejection sampling tends to reject almost all samples.
  - Importance sampling tends gives almost zero weight to all samples.
- We want an algorithm that gets better over time.
- Two main strategies:
  - Sequential Monte Carlo:
    - Importance sampling where proposal q_t changes over time from simple to posterior.
    - "Particle Filter Explained without Equations": `https://www.youtube.com/watch?v=aUkBa1zMKv4`

Limitations of Simple Monte Carlo Methods

- These methods tend not to work in complex situations:
  - Inverse CDF may not be avaiable.
  - Conditional needed for ancestral sampling may be hard to compute.
  - Rejection sampling tends to reject almost all samples.
  - Importance sampling tends gives almost zero weight to all samples.
- We want an algorithm that gets better over time.
- Two main strategies:
  - Sequential Monte Carlo:
    - Importance sampling where proposal q_t changes over time from simple to posterior.
    - "Particle Filter Explained without Equations":
      `https://www.youtube.com/watch?v=aUkBa1zMKv4`
  - Markov chain Monte Carlo (MCMC).
    - Design Markov chain whose stationary distribution is the posterior.

Motivating Example: Sampling from a UGM

- High-dimensional integration problems arise in other settings:
  - Bayesian graphical models and Bayesian neural networks.
  - Deep belief networks, Boltzmann machines.

Motivating Example: Sampling from a UGM

- High-dimensional integration problems arise in other settings:
  - Bayesian graphical models and Bayesian neural networks.
  - Deep belief networks, Boltzmann machines.
- Recall the definition of a discrete paiwise undirected graphical model (UGM):

$$p(x) = \frac{\prod_{j=1}^{d} \phi_j(x_j) \prod_{(i,j)\in E} \phi_{ij}(x_i, x_j)}{Z} = \frac{\tilde{p}(x)}{Z}.$$

Motivating Example: Sampling from a UGM

- High-dimensional integration problems arise in other settings:
  - Bayesian graphical models and Bayesian neural networks.
  - Deep belief networks, Boltzmann machines.
- Recall the definition of a discrete paiwise undirected graphical model (UGM):

$$p(x) = \frac{\prod_{j=1}^{d} \phi_j(x_j) \prod_{(i,j) \in E} \phi_{ij}(x_i, x_j)}{Z} = \frac{\tilde{p}(x)}{Z}.$$

- In this model:
  - Compute $\tilde{p}(x)$ is easy.
  - Computing Z is #P-hard.
  - Generating a sample is NP-hard (at least).

Motivating Example: Sampling from a UGM

- High-dimensional integration problems arise in other settings:
  - Bayesian graphical models and Bayesian neural networks.
  - Deep belief networks, Boltzmann machines.
- Recall the definition of a discrete paiwise undirected graphical model (UGM):

$$p(x) = \frac{\prod_{j=1}^{d} \phi_j(x_j) \prod_{(i,j)\in E} \phi_{ij}(x_i, x_j)}{Z} = \frac{\tilde{p}(x)}{Z}.$$

- In this model:
  - Compute $\tilde{p}(x)$ is easy.
  - Computing Z is #P-hard.
  - Generating a sample is NP-hard (at least).
- With rejection sampling, probability of acceptance might be arbitrarily small.
- But there is a simple MCMC method...

Gibbs Sampling for Discrete UGMs

- A Gibbs sampling algorithm for pairwise UGMs:
  - Start with some configuration x^0, then repeat the following:
    1. Choose a variable j uniformly at random.
    2. Set $x_{-j}^{t+1} = x_{-j}^t$, and sample x_j^t from its conditional,

$$x_j^{t+1} \sim p(x_j | x_{-j}^t) = p(x_j | x_{\mathsf{MB}(j)}^t).$$

Gibbs Sampling for Discrete UGMs

- A Gibbs sampling algorithm for pairwise UGMs:
  - Start with some configuration x^0, then repeat the following:
    1. Choose a variable j uniformly at random.
    2. Set $x_{-j}^{t+1} = x_{-j}^t$, and sample x_j^t from its conditional,

$$x_j^{t+1} \sim p(x_j | x_{-j}^t) = p(x_j | x_{\mathsf{MB}(j)}^t).$$

  - Analogy: sampling version of coordinate descent:
    - Transformed d-dimensional sampling into 1-dimensional sampling.

Gibbs Sampling for Discrete UGMs

- A Gibbs sampling algorithm for pairwise UGMs:
  - Start with some configuration x^0, then repeat the following:
    1. Choose a variable j uniformly at random.
    2. Set $x_{-j}^{t+1} = x_{-j}^t$, and sample x_j^t from its conditional,

$$x_j^{t+1} \sim p(x_j | x_{-j}^t) = p(x_j | x_{\mathsf{MB}(j)}^t).$$

  - Analogy: sampling version of coordinate descent:
    - Transformed d-dimensional sampling into 1-dimensional sampling.
- These iterations are very cheap:
  - Need to know $\tilde{p}(x^t)$ for each value of x_j^t.
  - Then sample from a single discrete random variable.
- Does this work? How long does this take?

Gibbs Sampling in Action

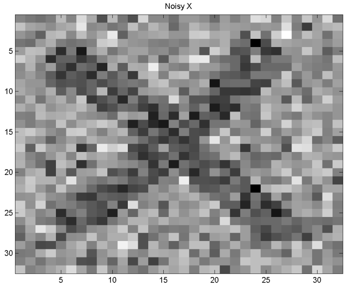
- Start with some initial value: $x^0 = \begin{bmatrix} 2 & 2 & 3 & 1 \end{bmatrix}$.
- Select random j: $j = 3$.
- Sample variable j: $x^2 = \begin{bmatrix} 2 & 2 & 1 & 1 \end{bmatrix}$.
- Select random j: $j = 1$.
- Sample variable j: $x^3 = \begin{bmatrix} 3 & 2 & 1 & 1 \end{bmatrix}$.
- Select random j: $j = 2$.
- Sample variable j: $x^4 = \begin{bmatrix} 3 & 2 & 1 & 1 \end{bmatrix}$.

$\vdots$

- Use all these samples to make approximation of $p(x)$.

Gibbs Sampling in Action: UGMs

Consider using a UGM for image denoising:



We have

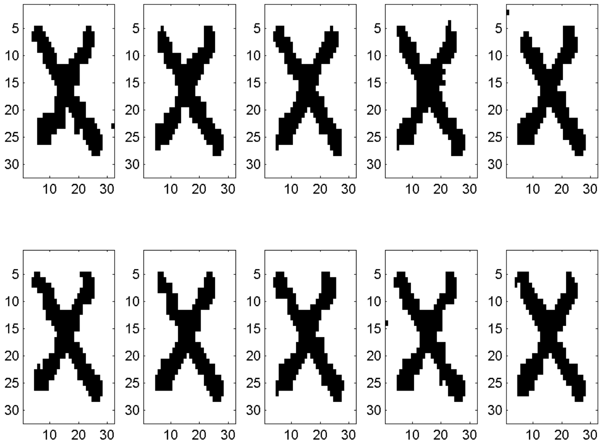
- Unary potentials ϕ_j for each position.
- Pairwise potentials ϕ_{ij} for neighbours on grid.
- Parameters are trained as CRF (next time).

Goal is to produce a noise-free image.

Gibbs Sampling in Action: UGMs

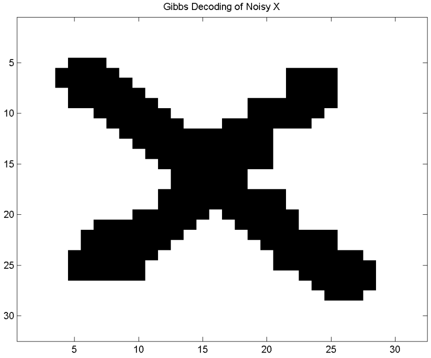
Gibbs samples after every $100d$ iterations:

Samples from Gibbs sampler



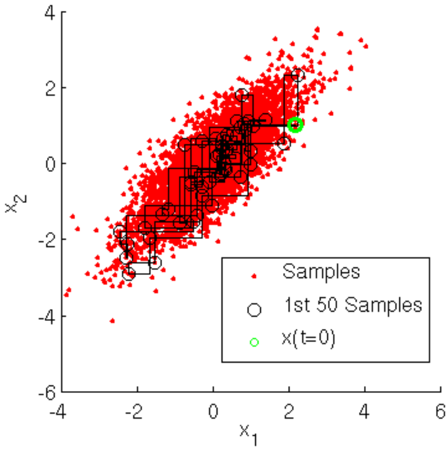
Gibbs Sampling in Action: UGMs

Mean image and marginal decoding:



Gibbs Sampling in Action: Multivariate Gaussian

- Gibbs sampling works for general distributions.
    - E.g., sampling from multivariate Gaussian by univariate Gaussian sampling.



Outline

Homoegenous Markov Chains and Invariant Distribution

- Given initial distribution $p(x^0)$ Markov chain assumes that

$$p(x^t|x^{1:t-1}) = p(x^t|x^{t-1}),$$

which we call the Markov property.

Homoegenous Markov Chains and Invariant Distribution

- Given initial distribution $p(x^0)$ Markov chain assumes that

$$p(x^t|x^{1:t-1}) = p(x^t|x^{t-1}),$$

  which we call the Markov property.

- Important special case is homogenous Markov chains, where

$$p(x^t = s|x^{t-1} = s') = p(x^{t-1} = s|x^{t-2} = s'),$$

  for all s, s', and t (transition probabilities don't change over time).

Homoegenous Markov Chains and Invariant Distribution

- Given initial distribution $p(x^0)$ Markov chain assumes that

$$p(x^t|x^{1:t-1}) = p(x^t|x^{t-1}),$$

  which we call the Markov property.
- Important special case is homogenous Markov chains, where

$$p(x^t = s|x^{t-1} = s') = p(x^{t-1} = s|x^{t-2} = s'),$$

  for all s, s', and t (transition probabilities don't change over time).
- Under weak conditions, homogenous chains converge to an invariant distribution,

$$p(s) = \sum_{s'} p(x^t = s|x^{t-1} = s')p(s').$$

  E.g., $p(x^t|x^{t-1}) > 0$ is sufficient, or weaker condition of "irreducible and aperiodic".

Markov Chain Monte Carlo

- Markov chain Monte Carlo (MCMC): given target p, design transitions such that

$$\frac{1}{n}\sum_{t=1}^{n} f(x^t) \to \int_x f(x)p(x)dx \quad \text{and/or} \quad x^n \sim p,$$

as $n \to \infty$.

- We are generating dependent samples that still solve the integral.

Markov Chain Monte Carlo

- Markov chain Monte Carlo (MCMC): given target p, design transitions such that

$$\frac{1}{n}\sum_{t=1}^{n} f(x^t) \to \int_x f(x)p(x)dx \quad \text{and/or} \quad x^n \sim p,$$

  as $n \to \infty$.
- We are generating dependent samples that still solve the integral.
- There are many transitions that will yield posterior as invariant distribution.
  - Typically easy to design sampler, but hard to characterize rate of convergence.

Markov Chain Monte Carlo

- Markov chain Monte Carlo (MCMC): given target p, design transitions such that

$$\frac{1}{n}\sum_{t=1}^{n} f(x^t) \to \int_x f(x)p(x)dx \quad \text{and/or} \quad x^n \sim p,$$

  as $n \to \infty$.
- We are generating dependent samples that still solve the integral.
- There are many transitions that will yield posterior as invariant distribution.
  - Typically easy to design sampler, but hard to characterize rate of convergence.
- Gibbs sampling satisfies the above under very weak conditions.

Markov Chain Monte Carlo

- Markov chain Monte Carlo (MCMC): given target p, design transitions such that

$$\frac{1}{n}\sum_{t=1}^{n} f(x^t) \to \int_x f(x)p(x)dx \quad \text{and/or} \quad x^n \sim p,$$

  as $n \to \infty$.
- We are generating dependent samples that still solve the integral.
- There are many transitions that will yield posterior as invariant distribution.
  - Typically easy to design sampler, but hard to characterize rate of convergence.
- Gibbs sampling satisfies the above under very weak conditions.
- Typically, we don't take all samples:
  - Burn in: throw away the initial samples when we haven't converged to stationary.
  - Thinning: only keep every k samples, since they will be highly correlated.

Markov Chain Monte Carlo

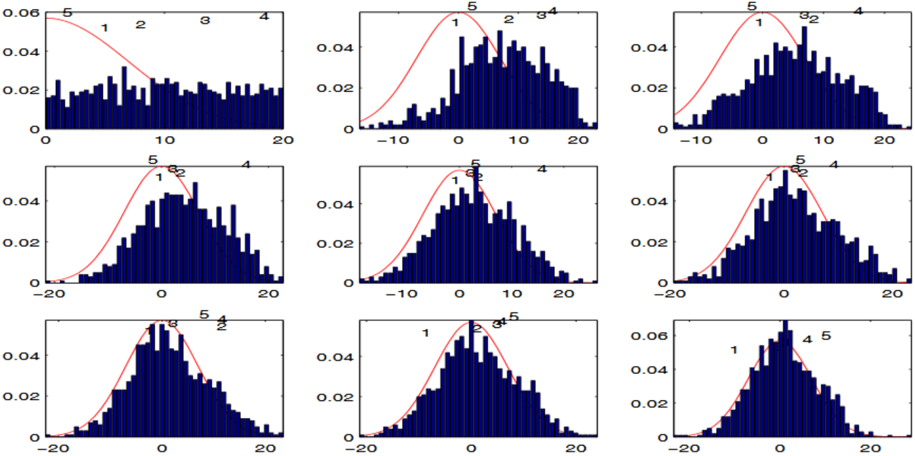
- Markov chain Monte Carlo (MCMC): given target p, design transitions such that

$$\frac{1}{n}\sum_{t=1}^{n} f(x^t) \rightarrow \int_x f(x)p(x)dx \quad \text{and/or} \quad x^n \sim p,$$

  as $n \rightarrow \infty$.
- We are generating dependent samples that still solve the integral.
- There are many transitions that will yield posterior as invariant distribution.
  - Typically easy to design sampler, but hard to characterize rate of convergence.
- Gibbs sampling satisfies the above under very weak conditions.
- Typically, we don't take all samples:
  - Burn in: throw away the initial samples when we haven't converged to stationary.
  - Thinning: only keep every k samples, since they will be highly correlated.
- It can very hard to diagnose if we reached invariant distribution.
  - Recent work showed that this is P-space hard (much worse than NP-hard).

Markov Chain Monte Carlo

From top left to bottom right: histograms of 1000 independent Markov chains with a normal distribution as target distribution.



Gibbs Sampilng: Variations

- Block Gibbs sampling samples multiple variables:
  - Sample a number of variables $k > 1$ jointly.
  - Sample a tree-structured subgraph of a UGM.

Gibbs Sampilng: Variations

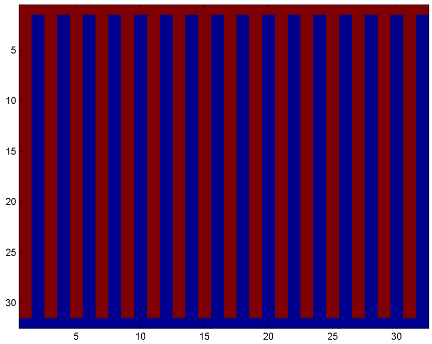
- Block Gibbs sampling samples multiple variables:
  - Sample a number of variables $k > 1$ jointly.
  - Sample a tree-structured subgraph of a UGM.
- Auxiliary-variable sampling: Introduce variables to sample bigger blocks:
  - E.g., introduce z variables in mixture models.
  - Also used in Bayesian logistic regression.

Gibbs Sampilng: Variations

- Block Gibbs sampling samples multiple variables:
  - Sample a number of variables $k > 1$ jointly.
  - Sample a tree-structured subgraph of a UGM.
- Auxiliary-variable sampling: Introduce variables to sample bigger blocks:
  - E.g., introduce z variables in mixture models.
  - Also used in Bayesian logistic regression.
- Collapsed or Rao-Blackwellized: integrate out variables that are not of interest.
  - Provably decrease variance of sampler.
  - E.g., integrate out hidden states in Bayesian hidden Markov model.

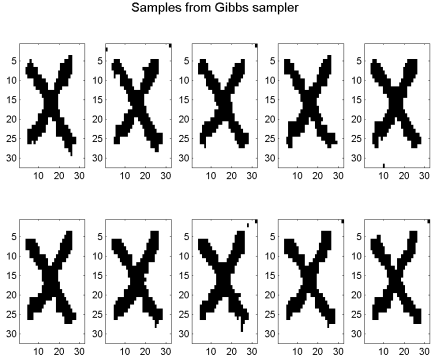
Block Gibbs Sampling in Action

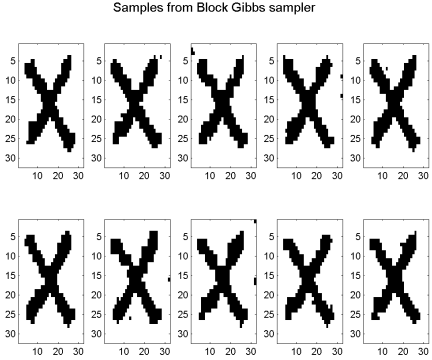
For denoising task, we could use two tree-structured blocks:



Block Gibbs Sampling in Action

Gibbs vs. tree-structured block-Gibbs samples:





Limitations of Gibbs Sampling

- Gibbs sampling is nice because it has no parameters:
  - You just need to decide on the blocks and auxiliary variables.

Limitations of Gibbs Sampling

- Gibbs sampling is nice because it has no parameters:
  - You just need to decide on the blocks and auxiliary variables.
- But it isn't always ideal:
  - Samples can be very correlated: slow progress.
  - Conditional may not have a nice form:
    - If Markov blanket is not conjugate, need rejection/importance sampling.

Limitations of Gibbs Sampling

- Gibbs sampling is nice because it has no parameters:
  - You just need to decide on the blocks and auxiliary variables.
- But it isn't always ideal:
  - Samples can be very correlated: slow progress.
  - Conditional may not have a nice form:
    - If Markov blanket is not conjugate, need rejection/importance sampling.
- Generalization that can address these is Metropolis-Hastings:
  - Oldest algorithm among the "Best of the 20th Century".

Outline

Metropolis Algorithms

- The Metropolis algorithm for sampling from a continuous $\tilde{p}(x)$:
  - Start from some x^0 and on iteration t:
    1. Add zero-mean Gaussian noise to x^t to generate $\tilde{x}^t$.
    2. Generate u from a $\mathcal{U}(0,1)$.

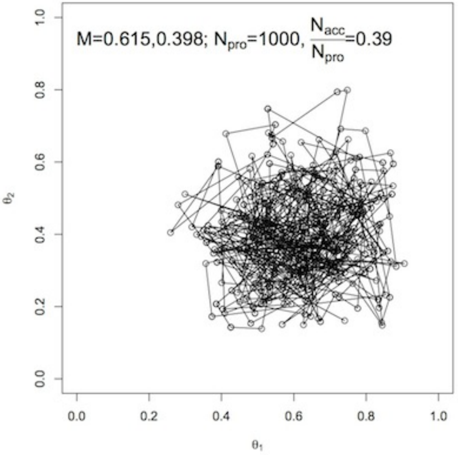
Metropolis Algorithms

- The Metropolis algorithm for sampling from a continuous $\tilde{p}(x)$:
  - Start from some x^0 and on iteration t:
    1. Add zero-mean Gaussian noise to x^t to generate $\tilde{x}^t$.
    2. Generate u from a $\mathcal{U}(0,1)$.
    3. Accept the sample and set $x^{t+1} = \tilde{x}^t$ if

$$u \leq \frac{\tilde{p}(\tilde{x}^t)}{\tilde{p}(x^t)},$$

    and otherwise reject the sample and set $x^{t+1} = x^t$.
  - A random walk, but sometimes rejecting steps that decrease probability:
    - Another valid MCMC algorithm, although convergence may again be slow.

Metropolis Algorithm in Action



http://www.columbia.edu/~cjd11/charles_dimaggio/DIRE/styled-4/styled-11/code-5/

Metropolis Algorithm Analysis

- Markov chain with transitions $p_{ss'} = p(x^t = s' | x^{t-1} = s)$ is reversible if there exists p such that

$$p(s)p_{ss'} = p(s')p_{s's},$$

which is called detailed balance.

Metropolis Algorithm Analysis

- Markov chain with transitions $p_{ss'} = p(x^t = s' | x^{t-1} = s)$ is reversible if there exists p such that
$$p(s)p_{ss'} = p(s')p_{s's},$$
which is called detailed balance.
- Assuming we reach stationary, detailed balance is sufficent for p to be the stationary distribution,

$$\sum_s p(s)p_{ss'} = \sum_s p(s')p_{s's}$$
$$\sum_s p(s)p_{ss'} = p(s')\underbrace{\sum_s p_{ss'}}_{=1}$$
$$\sum_s p(s)p_{ss'} = p(s') \qquad \text{(stationary condition)}$$

Metropolis Algorithm Analysis

- Metropolis algorithm has $p_{ss'} > 0$ and satisfies detailed balance,

$$p(s)p_{ss'} = p(s')p_{s's}.$$

- We can show this by defining transition kernel

$$T_{ss'} = \min\left\{1, \frac{\tilde{p}(s')}{\tilde{p}(s)}\right\},$$

Metropolis Algorithm Analysis

- Metropolis algorithm has $p_{ss'} > 0$ and satisfies detailed balance,
$$p(s)p_{ss'} = p(s')p_{s's}.$$
- We can show this by defining transition kernel
$$T_{ss'} = \min\left\{1, \frac{\tilde{p}(s')}{\tilde{p}(s)}\right\},$$
and observing that
$$\begin{aligned} p(s)T_{ss'} &= p(s)\min\left\{1, \frac{\tilde{p}(s')}{\tilde{p}(s)}\right\} = p(s)\min\left\{1, \frac{\frac{1}{Z}\tilde{p}(s')}{\frac{1}{Z}\tilde{p}(s)}\right\} \\ &= p(s)\min\left\{1, \frac{p(s')}{p(s)}\right\} = \min\left\{p(s), p(s')\right\} \\ &= p(s')\min\left\{1, \frac{p(s)}{p(s')}\right\} = p(s')T_{s's}. \end{aligned}$$

Metropolis-Hastings

- Instead of Gaussian noise, consider a general proposal distribution q:
  - Value $q(\tilde{x}^t|x^t)$ is probability of proposing $\tilde{x}^t$.

Metropolis-Hastings

- Instead of Gaussian noise, consider a general proposal distribution q:
  - Value $q(\tilde{x}^t | x^t)$ is probability of proposing $\tilde{x}^t$.
- Metropolis-Hastings accepts proposal if

$$u \leq \frac{\tilde{p}(\tilde{x}^t) q(x^t | \tilde{x}^t)}{\tilde{p}(x^t) q(\tilde{x}^t | x^t)},$$

where extra terms ensure detailed balance for asymmetric q:

- E.g., if you are more likely to propose to go from x^t to $\tilde{x}^t$ than the reverse.

Metropolis-Hastings

- Instead of Gaussian noise, consider a general proposal distribution q:
  - Value $q(\tilde{x}^t|x^t)$ is probability of proposing $\tilde{x}^t$.
- Metropolis-Hastings accepts proposal if

$$u \leq \frac{\tilde{p}(\tilde{x}^t)q(x^t|\tilde{x}^t)}{\tilde{p}(x^t)q(\tilde{x}^t|x^t)},$$

  where extra terms ensure detailed balance for asymmetric q:
  - E.g., if you are more likely to propose to go from x^t to $\tilde{x}^t$ than the reverse.
- This again works under very weak conditions, such as $q(\tilde{x}^t|x^t) > 0$.
- Gibbs sampling is a special case, but we have a lot of flexibility:
  - You can make performance much better/worse with an appropriate q.

Metropolis-Hastings

- Simple choices for proposal distribution q:
  - Metropolis originally used random walks: $x^t = x^{t-1} + \epsilon$ for $\epsilon \sim \mathcal{N}(0, \Sigma)$.
  - Hastings originally used independent proposal: $q(x^t | x^{t-1}) = q(x^t)$.

Metropolis-Hastings

- Simple choices for proposal distribution q:
  - Metropolis originally used random walks: $x^t = x^{t-1} + \epsilon$ for $\epsilon \sim \mathcal{N}(0, \Sigma)$.
  - Hastings originally used independent proposal: $q(x^t | x^{t-1}) = q(x^t)$.
  - Gibbs sampling updates single variable based on conditional:
    - In this case the acceptance rate is 1 so we never reject.

Metropolis-Hastings

- Simple choices for proposal distribution q:
  - Metropolis originally used random walks: $x^t = x^{t-1} + \epsilon$ for $\epsilon \sim \mathcal{N}(0, \Sigma)$.
  - Hastings originally used independent proposal: $q(x^t | x^{t-1}) = q(x^t)$.
  - Gibbs sampling updates single variable based on conditional:
    - In this case the acceptance rate is 1 so we never reject.
  - Mixture model for q: e.g., between big and small moves.

Metropolis-Hastings

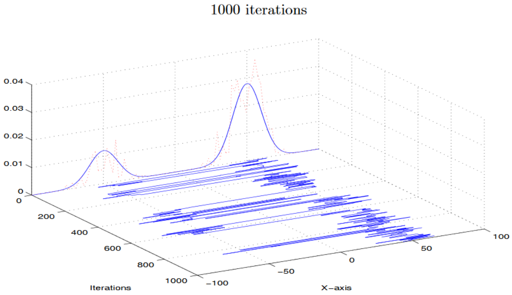
- Simple choices for proposal distribution q:
  - Metropolis originally used random walks: $x^t = x^{t-1} + \epsilon$ for $\epsilon \sim \mathcal{N}(0, \Sigma)$.
  - Hastings originally used independent proposal: $q(x^t | x^{t-1}) = q(x^t)$.
  - Gibbs sampling updates single variable based on conditional:
    - In this case the acceptance rate is 1 so we never reject.
  - Mixture model for q: e.g., between big and small moves.
  - "Adaptive MCMC": tries to update q as we go: needs to be done carefully.
  - "Particle MCMC": use particle filter to make proposal.

Metropolis-Hastings

- Simple choices for proposal distribution q:
  - Metropolis originally used random walks: $x^t = x^{t-1} + \epsilon$ for $\epsilon \sim \mathcal{N}(0, \Sigma)$.
  - Hastings originally used independent proposal: $q(x^t | x^{t-1}) = q(x^t)$.
  - Gibbs sampling updates single variable based on conditional:
    - In this case the acceptance rate is 1 so we never reject.
  - Mixture model for q: e.g., between big and small moves.
  - "Adaptive MCMC": tries to update q as we go: needs to be done carefully.
  - "Particle MCMC": use particle filter to make proposal.
- Unlike rejection sampling, we don't want acceptance rate as high as possible:
  - High acceptance rate may mean we're not moving very much.
  - Low acceptance rate definitely means we're not moving very much.
  - Designing q is an "art".

Metropolis-Hastings

Metropolis-Hastings for sampling from mixture of Gaussians:



http://www.cs.ubc.ca/~arnaud/stat535/slides10.pdf

- High acceptance rate could mean we are staying in one mode.
- We may to proposal to be mixture between random walk and "mode jumping".

Outline

Stochastic Processes and Non-Parametric Bayes

- A stochastic process is an infinite collection of random variables $\{x^i\}$.
- Non-parametric Bayesian methods use priors defined on stochastic processes:
  - Allows extremely-flexible prior, and posterior complexity grows with data size.
  - Typically set up so that samples from posterior are finite-sized.
- The two most common priors are Gaussian processes and Dirichlet processes:
  - Gaussian processes define prior on space of functions (universal approximators).
  - Dirichlet processes define prior on space of probabilities (without fixing dimension).

Gaussian Processes

- Recall that we can partition a multivariate Gaussian:

$$\mu = \left[\mu_x, \mu_y\right], \quad \Sigma = \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix},$$

and marginal distribution wrt x variables is just a $\mathcal{N}(\mu_x, \Sigma_x x)$ Gaussian.

Gaussian Processes

- Recall that we can partition a multivariate Gaussian:

$$\mu = \left[\mu_x, \mu_y\right], \quad \Sigma = \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix},$$

  and marginal distribution wrt x variables is just a $\mathcal{N}(\mu_x, \Sigma_x x)$ Gaussian.
- Generalization of this to infinite variables is Gaussian processes (GPs):
  - Infinite collection of random variables.
  - Any finite set from collection follows a Gaussian distribution.

Gaussian Processes

- Recall that we can partition a multivariate Gaussian:

$$\mu = \left[\mu_x, \mu_y\right], \quad \Sigma = \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix},$$

  and marginal distribution wrt x variables is just a $\mathcal{N}(\mu_x, \Sigma_x x)$ Gaussian.
- Generalization of this to infinite variables is Gaussian processes (GPs):
  - Infinite collection of random variables.
  - Any finite set from collection follows a Gaussian distribution.
- GPs are specified by a mean function m and covariance function k:
  - If

$$m(x) = \mathbb{E}[f(x)], \quad k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))^T],$$

    then we say that

$$f(x) \sim \mathsf{GP}(m(x), k(x, x')).$$

Regression Models as Gaussian Processes

- For example, predictions made by linear regression with Gaussian prior

$$f(x) = \phi(x)^T w, \quad w \sim \mathcal{N}(0, \Sigma),$$

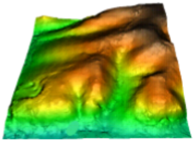
are a Gaussian process with mean function

$$\mathbb{E}[f(x)] = \mathbb{E}[\phi(x)^T w] = \phi(x)^T \mathbb{E}[w] = 0.$$

and covariance function

$$\mathbb{E}[f(x)f(x)^T] = \phi(x)^T \mathbb{E}[ww^T]\phi(x') = \phi(x)\Sigma\phi(x').$$

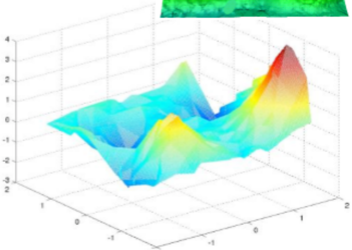
Gaussian Processes

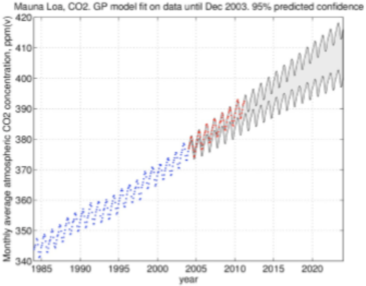


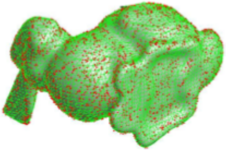
To date kriging has been used in a variety of disciplines, including the following:

- Environmental science[5]
- Hydrogeology[6][7][8]
- Mining[9][10]
- Natural resources[11][12]
- Remote sensing[13]
- Real estate appraisal[14][15]

and many others.

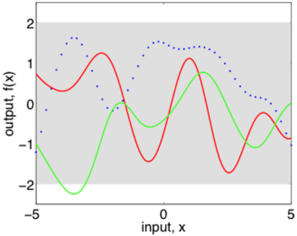




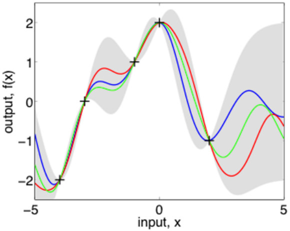


Gaussian Process Model Selection

- We can view a Gaussian process as a prior distribution over smooth functions.



(a), prior

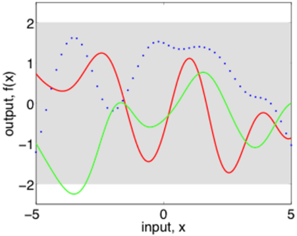


(b), posterior

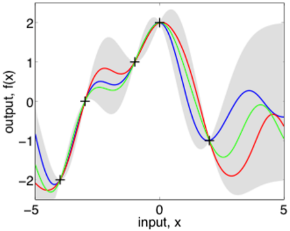
- Most common choice of covariance is RBF.

Gaussian Process Model Selection

- We can view a Gaussian process as a prior distribution over smooth functions.



(a), prior

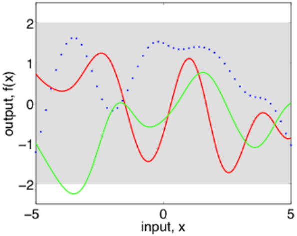


(b), posterior

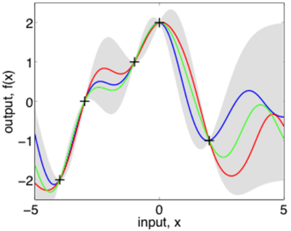
- Most common choice of covariance is RBF.
- Is this the same as using kernels?

Gaussian Process Model Selection

- We can view a Gaussian process as a prior distribution over smooth functions.



(a), prior



(b), posterior

- Most common choice of covariance is RBF.
- Is this the same as using kernels?
  - Yes, this is Bayesian linear regression plus the kernel trick.

Gaussian Process Model Selection

- So why do we care?
  - We can get estimate of uncertainty in the prediction.
  - We can use marginal likelihood to learn the kernel/covariance.
- Non-hierarchical approach:
  - Write kernel in terms of parameters, optimize parameters to learn kernel.

Gaussian Process Model Selection

- So why do we care?
  - We can get estimate of uncertainty in the prediction.
  - We can use marginal likelihood to learn the kernel/covariance.
- Non-hierarchical approach:
  - Write kernel in terms of parameters, optimize parameters to learn kernel.
- Hierarchical approach: put a hyper-prior of types of kernels.
- Can be viewed as an automatic statistician:
  `http://www.automaticstatistician.com/examples/`

Dirichlet Process

- Recall the finite mixture model:

$$p(x|\theta) = \sum_{c=1}^{k} \pi_c p(x|\theta_c).$$

- Non-parametric Bayesian methods allow us to consider infinite mixture model,

$$p(x|\theta) = \sum_{c=1}^{\infty} \pi_c p(x|\theta_c).$$

- Common choice for prior on π values is Dirichlet process:
  - Also called "Chinese restaurant process" and "stick-breaking process".
  - For finite datasets, only a fixed number of clusters have $\pi_c \neq 0$.
  - But don't need to pick number of clusters, grows with data size.

Dirichlet Process

- Gibbs sampling in Dirichlet process mixture model in action:
  `https://www.youtube.com/watch?v=0Vh7qZY9sPs`

Dirichlet Process

- Gibbs sampling in Dirichlet process mixture model in action: `https://www.youtube.com/watch?v=0Vh7qZY9sPs`
- We could alternately put a prior on k:
  - "Reversible-jump" MCMC can be used to sample from models of different sizes.
- There a variety of interesting extensions:
  - Beta process.
  - Hierarchical Dirichlet process,.
  - Polya trees.
  - Infinite hidden Markov models.

Summary

- Markov chain Monte Carlo generates a sequence of *dependent samples*:
  - But asymptotically these samples come from the posterior.

Summary

- Markov chain Monte Carlo generates a sequence of *dependent samples*:
  - But asymptotically these samples come from the posterior.
- Gibbs sampling is special of repeatedly sampling one variable at time.
  - Works poorly, but effective extensions like block/collapsed Gibbs.

Summary

- Markov chain Monte Carlo generates a sequence of *dependent samples*:
  - But asymptotically these samples come from the posterior.
- Gibbs sampling is special of repeatedly sampling one variable at time.
  - Works poorly, but effective extensions like block/collapsed Gibbs.
- Metropolis-Hastings is generalization allowing arbtirary "proposals".

Summary

- Markov chain Monte Carlo generates a sequence of *dependent samples*:
  - But asymptotically these samples come from the posterior.
- Gibbs sampling is special of repeatedly sampling one variable at time.
  - Works poorly, but effective extensions like block/collapsed Gibbs.
- Metropolis-Hastings is generalization allowing arbtirary "proposals".
- Non-Parametric Bayesian methods use flexible infinite-dimensional priors:
  - Allows model complexity to grow with data size.

- Next time: most cited ML paper in the 00s and variational inference.