Course Outline

- Regression Techniques
  - Linear Regression
  - Regularization (Ridge, Lasso)
  - Kernels (Kernel Ridge Regression)
- Deep Reinforcement Learning
- Probabilistic Graphical Models
  - Bayesian Networks
  - Markov Random Fields
  - Inference (exact & approximate)
- Deep Generative Models
  - Generative Adversarial Networks
  - Variational Autoencoders

Topics of This Lecture

- Approximate Inference
  - Variational methods
  - Sampling approaches
- Sampling approaches
  - Sampling from a distribution
  - Ancestral Sampling
  - Rejection Sampling
  - Importance Sampling
- Markov Chain Monte Carlo
  - Markov Chains
  - Metropolis Algorithm
  - Metropolis-Hastings Algorithm
  - Gibbs Sampling

Approximate Inference

- Exact Bayesian inference is often intractable.
  - Often infeasible to evaluate the posterior distribution or to compute expectations w.r.t. the distribution.
  - E.g. because the dimensionality of the latent space is too high.
  - Or because the posterior distribution has a too complex form.
- Problems with continuous variables
  - Required integrations may not have closed-form solutions.
- Problems with discrete variables
  - Marginalization involves summing over all possible configurations of the hidden variables.
  - There may be exponentially many such states.

⇒ We need to resort to approximation schemes.

Two Classes of Approximation Schemes

- Deterministic approximations (Variational methods)
  - Based on analytical approximations to the posterior distribution
    - E.g. by assuming that it factorizes in a certain form
    - Or that it has a certain parametric form (e.g., a Gaussian).
  - Can never generate exact results, but are often scalable to large applications.
- Stochastic approximations (Sampling methods)
  - Given infinite computational resources, they can generate exact results.
  - Approximation arises from the use of a finite amount of processor time.
  - Enable the use of Bayesian techniques across many domains.
  - But: computationally demanding, often limited to small-scale problems.
In general, assume we are given the pdf \( p(x) \) and the corresponding cumulative distribution:
\[
F(x) = \int_{-\infty}^{x} p(z)dz
\]

- To draw samples from this pdf, we can invert the cumulative distribution function:
\[
u \sim Uniform(0, 1) \Rightarrow F^{-1}(u) \sim p(x)
\]

Example 1: Sampling from Exponential Distrib.

- **Exponential Distribution**

\[
p(y) = \lambda \exp(-\lambda y)
\]

where \( 0 \leq y < \infty \).

- **Transformation sampling**

  - Indefinite Integral

  \[
h(y) = 1 - \exp(-\lambda y)
\]

  - Inverse function

\[
y = h(y)^{-1} = -\lambda^{-1} \ln(1-z)
\]

for a uniformly distributed input variable \( z \).
• Cauchy Distribution
  \[ p(y) = \frac{1}{\pi (1 + y^2)} \]

• Transformation sampling
  Inverse of integral can be expressed as a \( \tan \) function.
  \[ y = h(y)^{-1} = \tan(z) \]
  for a uniformly distributed input variable \( z \).

Example 2: Sampling from Cauchy Distrib.

Note: Efficient Sampling from a Gaussian

• Problem with transformation method
  – Integral over Gaussian cannot be expressed in analytical form.
  – Standard transformation approach is very inefficient.

• More efficient: Box-Muller Algorithm
  – Generate pairs of uniformly distributed random numbers \( z_1, z_2 \in (-1,1) \).
  – Discard each pair unless it satisfies \( r^2 = z_1^2 + z_2^2 \leq 1 \).
  – This leads to a uniform distribution of points inside the unit circle with \( p(z_1, z_2) = 1/\pi \).

Box-Muller Algorithm (cont’d)

• Box-Muller Algorithm (cont’d)
  \[ r^2 = z_1^2 + z_2^2 \]
  – For each pair \( z_1, z_2 \), evaluate
    \[ y_1 = z_1 \left( -\frac{2 \ln r^2}{r^2} \right)^{1/2} \]
    \[ y_2 = z_2 \left( -\frac{2 \ln r^2}{r^2} \right)^{1/2} \]
  – Then the joint distribution of \( y_1 \) and \( y_2 \) is given by
    \[ p(y_1, y_2) = \frac{\partial (z_1, z_2)}{\partial (y_1, y_2)} \]
    \[ = \left[ \frac{1}{\sqrt{2\pi}} \exp(-y_1^2/2) \right] \left[ \frac{1}{\sqrt{2\pi}} \exp(-y_2^2/2) \right] \]
  \( \Rightarrow y_1 \) and \( y_2 \) are independent and each has a Gaussian distribution
  with mean \( \mu \) and variance \( \sigma^2 \).
  – If \( y \sim N(0,1) \), then \( \sigma y + \mu \sim N(\mu, \sigma^2) \).

Box-Muller Algorithm (cont’d)

• Multivariate extension
  – If \( x \) is a vector valued random variable whose components
    are independent and Gaussian distributed with \( N(0,1) \),
  – Then \( y = \mu + L z \) will have mean \( \mu \) and covariance \( \Sigma \).
  – Where \( \Sigma = L L^T \) is the Cholesky decomposition of \( \Sigma \).

General Advice

• Use library functions whenever possible
  – Many efficient algorithms available for
    known univariate distributions
    (and some other special cases)
  – This book (free online) explains how some of them work
    – http://www.nrbook.com/devroye/

Ancestral Sampling

• Generalization of this idea to directed graphical models.
  – Joint probability factorizes into conditional probabilities:
    \[ p(x) = \prod_{k=1}^{K} p(x_k | p_{a_k}) \]

• Ancestral sampling
  – Assume the variables are ordered such that there are no links from
    any node to a lower-numbered node.
  – Start with lowest-numbered node and draw a sample from its distribution.
    \[ \hat{x}_1 \sim p(x_1) \]
  – Cycle through each of the nodes in order and draw samples from the conditional distribution
    (where the parent variable is set to its sampled value):
    \[ \hat{x}_n \sim p(x_n | p_{a_n}) \]
Approximate Inference I

Obviously we need some simpler distribution than the one in question.

\[ \mathcal{I} \]

Schiele dimensions are generated from the distribution \( \mathcal{I} \) from the transformation method. The remaining pairs \((u, z_i)\) have uniform distribution over the curve \( \int p(z) \).

Discussion

- Transformation method
  - Limited applicability, as we need to invert the indefinite integral of the required distribution \( p(x) \).
  - This will only be feasible for a limited number of simple distributions.
- More general
  - Rejection Sampling
  - Importance Sampling

Rejection Sampling

- Assumptions
  - Sampling directly from \( p(x) \) is difficult.
  - But we can easily evaluate \( p(x) \) up to some normalization factor \( Z_p \):
    \[ p(x) = \frac{1}{Z_p} f(x) \]
- Idea
  - We need some simpler distribution \( q(x) \) (called proposal distribution) from which we can draw samples.
  - Choose a constant \( k \) such that:
    \[ \forall x : kq(x) \geq p(x) \]

Rejection Sampling – Discussion

- Limitation: high-dimensional spaces
  - For rejection sampling to be of practical value, we require that \( kq(x) \) be close to the required distribution, so that the rate of rejection is minimal.
- Artificial example
  - Assume that \( p(x) \) is Gaussian with covariance matrix \( \sigma_x^2 I \)
  - Assume that \( q(x) \) is Gaussian with covariance matrix \( \sigma_q^2 I \)
  - Obviously: \( \sigma_x^2 \geq \sigma_q^2 \)
  - In \( D \) dimensions: \( k = (\sigma_q / \sigma_x)^D \).
    - Assume \( \sigma_x \) is just 1% larger than \( \sigma_q \).
    - \( D = 1000 \Rightarrow k = 1.01^{1000} \approx 20,000 \)
  - And \( p(\text{accept}) < \frac{1}{10000} \)

\[ \Rightarrow \text{Impractical to find good proposal distributions for high dimensions!} \]

Example: Sampling from a Gamma Distribution.

- Gamma distribution
  \[ \Gamma(a,b) = \frac{1}{\Gamma(a)} b^a \text{e}^{-b} \exp(-bx) \quad a > 1 \]
- Rejection sampling approach
  - For \( a > 1 \), Gamma distribution has a bell-shaped form.
  - Suitable proposal distribution is Cauchy (for which we can use the transformation method).
  - Generalize Cauchy slightly to ensure it is nowhere smaller than Gamma: \( y = b \tan(\theta + c) \) for uniform \( \theta \).
  - This gives random numbers distributed according to \( q(z) \) with optimal rejection rate for
    \[ c = \sigma - 1 \]
    \[ b^2 = 2a - 1 \]
Evaluating Expectations

- **Motivation**
  - Often, our goal is not sampling from \( p(x) \) by itself, but to evaluate expectations of the form
  \[
  \mathbb{E}[f] = \int f(x)p(x)dx
  \]
- **Simplistic strategy: Grid sampling**
  - Discretize \( x \)-space into a uniform grid.
  - Evaluate the integrand as a sum of the form
  \[
  \mathbb{E}[f] \approx \frac{1}{L} \sum_{l=1}^{L} f(x^{(l)})
  \]
  - Problem: number of terms grows exponentially with number of dimensions!

Importance Sampling

- **Typical setting:**
  - \( p(x) \) can only be evaluated up to an unknown normalization constant
    \( p(x) = \tilde{p}(x)/Z_p \)
  - \( q(x) \) can also be treated in a similar fashion.
    \( q(x) = \tilde{q}(x)/Z_q \)
  - Then
    \[
    \mathbb{E}[f] = \int f(x)p(x)dx = \frac{Z_q}{Z_p} \int f(x)\frac{\tilde{p}(x)}{\tilde{q}(x)}q(x)dx
    \]
    \[
    \approx \frac{Z_q}{Z_p} \sum_{l=1}^{L} \tilde{r}_l f(x^{(l)})
    \]
    - with:
    \[
    \tilde{r}_l = \frac{\tilde{p}(x^{(l)})}{\tilde{q}(x^{(l)})}
    \]

Importance Sampling – Discussion

- **Observations**
  - Success of importance sampling depends crucially on how well the sampling distribution \( q(x) \) matches the desired distribution \( p(x) \).
  - Often, \( p(x)/f(x) \) is strongly varying and has a significant proportion of its mass concentrated over small regions of \( x \)-space.
  - Weights \( \tilde{r}_l \) may be dominated by a few weights having large values.
  - Practical issue: if none of the samples falls in the regions where \( p(x)/f(x) \) is large...
    - The results may be ambiguous in error.
  - And there will be no diagnostic indication (no large variance in \( \tilde{r}_l \)).
  - Key requirement for sampling distribution \( q(x) \):
    - Should not be small or zero in regions where \( p(x) \) is significant!

Importance Sampling

- **Idea**
  - Use a proposal distribution \( q(x) \) from which it is easy to draw samples.
  - Express expectations in the form of a finite sum over samples \( \{x^{(l)}\} \) drawn from \( q(x) \).
    \[
    \mathbb{E}[f] = \int f(x)p(x)dx = \int f(x)\frac{\tilde{p}(x)}{\tilde{q}(x)}q(x)dx
    \]
    \[
    \approx \frac{1}{L} \sum_{l=1}^{L} \frac{\tilde{p}(x^{(l)})}{\tilde{q}(x^{(l)})} f(x^{(l)})
    \]
    - with importance weights
    \[
    \tilde{r}_l = \frac{\tilde{p}(x^{(l)})}{\tilde{q}(x^{(l)})}
    \]
  - Result
    - The resulting \( L \) samples \( x^{(1)}, \ldots, x^{(L)} \) are only approximately distributed according to \( p(x) \), but the distribution becomes correct in the limit \( L \to \infty \).

Sampling-Importance-Resampling

- **Two stages**
  - Draw \( L \) samples \( x^{(1)}, \ldots, x^{(L)} \) from \( q(x) \).
  - Construct weights using importance weighting
    \[
    w_l = \frac{\tilde{r}_l}{\sum_{m} \tilde{r}_m} = \frac{\tilde{p}(x^{(l)})}{\sum_{m} \tilde{p}(x^{(m)})}
    \]
    and draw a second set of samples \( x^{(1)}', \ldots, x^{(L)}' \) with probabilities given by the weights \( w^{(1)}', \ldots, w^{(L)}' \).

- **Result**
  - The resulting \( L \) samples are only approximately distributed according to \( p(x) \), but the distribution becomes correct in the limit \( L \to \infty \).
However, for many problems of practical interest, it is difficult or impossible to find $q(x)$ with the necessary properties. The proposal distribution depends on the current state: $q(z^t|z^{t-1})$. This is in contrast to rejection sampling, where rejected samples are simply discarded.

**Topics of This Lecture**

- Approximate Inference
  - Variational methods
  - Sampling approaches
- Sampling approaches
  - Sampling from a distribution
  - Ancestral Sampling
  - Rejection Sampling
  - Importance Sampling
- Markov Chain Monte Carlo
  - Markov Chains
  - Metropolis Algorithm
  - Metropolis-Hastings Algorithm
  - Gibbs Sampling

**Curve of Dimensionality**

- **Problem**
  - Rejection & Importance Sampling both scale badly with high dimensionality.
  - Example:
    $$p(x) \sim N(0, I), \quad q(x) \sim N(0, \sigma^2 I)$$
- **Rejection Sampling**
  - Requires $\sigma \geq 1$. Fraction of proposals accepted: $\sigma^{-1}$.
- **Importance Sampling**
  - Variance of importance weights:
    $$\left(\frac{\sigma^2}{2 - 1/\sigma^2}\right)^{1/2} - 1$$
  - Infinite / undefined variance if $\sigma \leq 1/\sqrt{2}$

**Independent Sampling vs. Markov Chains**

- **So far**
  - We’ve considered two methods, Rejection Sampling and Importance Sampling, which were both based on independent samples from $q(x)$.
  - However, for many problems of practical interest, it is difficult or impossible to find $q(x)$ with the necessary properties.
  - **Different approach**
    - We abandon the idea of independent sampling.
    - Instead, rely on a Markov Chain to generate dependent samples from the target distribution.
    - **Independence** would be a nice thing, but it is not necessary for the Monte Carlo estimate to be valid.

**MCMC – Markov Chain Monte Carlo**

- **Overview**
  - Allows to sample from a large class of distributions.
  - Scales well with the dimensionality of the sample space.
- **Idea**
  - We maintain a record of the current state $z^{t-1}$
  - The proposal distribution depends on the current state: $q(z^t|z^{t-1})$
  - The sequence of samples forms a Markov chain $z^{(1)}, z^{(2)}, \ldots$
- **Setting**
  - We can evaluate $p(x)$ (up to some normalizing factor $Z_0$):
    $$p(x) = \frac{p(x)}{Z_0}$$
    - At each time step, we generate a candidate sample from the proposal distribution and accept the sample according to a criterion.

**MCMC – Metropolis Algorithm**

- **Metropolis algorithm** ([Metropolis et al., 1953])
  - Proposal distribution is symmetric: $q(z_{t+1}|z_t) = q(z_t|z_{t+1})$
  - The new candidate sample $z'$ is accepted with probability
    $$A(z^t, z'^{(t)}) = \min\left(1, \frac{p(z')}{p(z^t)}\right)$$
- **Implementation**
  - Choose random number $u$ uniformly from unit interval $(0, 1)$.
  - Accept sample if $A(z^t, z'^{(t)}) > u$
- **Note**
  - New candidate samples always accepted if $p(z') \geq p(z^{(t)})$.
  - I.e. when new sample has higher probability than the previous one.
  - The algorithm sometimes accepts a state with lower probability.

- **Two cases**
  - If new sample is accepted: $z^{(t+1)} = z'$
  - Otherwise: $z^{(t+1)} = z^{(t)}$
  - This is in contrast to rejection sampling, where rejected samples are simply discarded.
  - ⇒ Leads to multiple copies of the same sample!
Approximate Inference

Many samples with very low weights...

1. Perturb parameters: e.g.,

- Sequence $x^{(t)}$, $x^{(t+1)}$, ... is not a set of independent samples from $p(x)$, as successive samples are highly correlated.
- We can obtain (largely) independent samples by just retaining every $M$th sample.

2. Example: Sampling from a Gaussian

- Proposal: Gaussian with $\sigma = 0.2$.
- Green: accepted samples
- Red: rejected samples

3. Importances Sampling weights

$$p(x) \propto \frac{1}{\prod_{i=1}^{M} p(x)}$$

4. Properties

- $p(x)$ is invariant, because

$$p(x) \propto \frac{1}{\prod_{i=1}^{M} p(x)}$$

5. Detailed balance

- Sufficient (but not necessary) condition to ensure that a distribution is invariant:

$$p(x) \propto \frac{1}{\prod_{i=1}^{M} p(x)}$$

- A Markov chain which respects detailed balance is reversible.

6. Question

- How can we show that $x^\tau$ tends to $p(x)$ as $\tau \to \infty$?

7. Markov chains

- First-order Markov chain:

$$p(x^{(m+1)}|x^{(m)}, \ldots, x^{(1)}) = p(x^{(m+1)}|x^{(m)})$$

8. Marginal probability

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

- A Markov chain is called homogeneous if the transition probabilities $p(x^{(m+1)}|x^{(m)})$ are the same for all $m$.

9. Detailed balance means

- If we pick a state from the target distribution $p(x)$ and make a transition under $T$ to another state, it is just as likely that we will pick $x_j$ and go from $x_k$ to $x_j$ than that we will pick $x_j$ and go from $x_k$ to $x_j$.

- It can easily be seen that a transition probability that satisfies detailed balance w.r.t. a particular distribution will leave that distribution invariant, because

$$\sum_{x'} p'(x') T(x', x) = \sum_{x'} p'(x) T(x', x)$$

10. $p(x) \sum_{x'} p(x'|x) = p(x)$

Line Fitting Example

11. Importance Sampling weights

- Many samples with very low weights...

- $w = 0.0548$ $w = 1.60e+08$ $w = 9.65e-06$ $w = 0.371$ $w = 0.103$

- $w = 1.01e-06$ $w = 0.111$ $w = 1.92e-09$ $w = 0.0126$ $w = 1.1e-51$

Markov Chains – Properties

- A distribution is said to be invariant (or stationary) w.r.t. a Markov chain if each step in the chain leaves that distribution invariant.

- Transition probabilities:

$$T(x^{(m)}, x^{(m+1)}) = p(x^{(m+1)}|x^{(m)})$$

- Distribution $p(x)$ is invariant if:

$$p(x) = \sum_{x'} T(x', x) p(x')$$

- Detailed balance

- Sufficient (but not necessary) condition to ensure that a distribution is invariant:

$$p(x) T(x', x) = p(x') T(x', x)$$

- A Markov chain which respects detailed balance is reversible.
Approximate Inference I

**Approximate Inference I**

- **Remark**
  - Our goal is to use Markov chains to sample from a given distribution.
  - We can achieve this if we set up a Markov chain such that the desired distribution is invariant.
  - However, must also require that for $m \to \infty$, the distribution $p(x^{(m)})$ converges to the required invariant distribution $p^*(x)$ irrespective of the choice of initial distribution $p(x^{(0)})$.
  - This property is called ergodicity and the invariant distribution is called the equilibrium distribution.
  - It can be shown that this is the case for a homogeneous Markov chain, subject only to weak restrictions on the invariant distribution and the transition probabilities.

**Ergodicity in Markov Chains**

- **Remark**
  - In practice, we often construct the transition probabilities from a set of 'base' transitions $B_1,...,B_K$.
  - This can be achieved through a mixture distribution
  
  $T'(x',x) = \sum_{k=1}^{K} \alpha_k B_k(x',x)$
  
  with mixing coefficients $\alpha_k \geq 0$ and $\sum_k \alpha_k = 1$.

- **Properties**
  - If the distribution is invariant w.r.t. each of the base transitions, then it will also be invariant w.r.t. $T'(x',x)$.
  - If each of the base transitions satisfies detailed balance, then the mixture transition $T$ will also satisfy detailed balance.
  - Common example: each base transition changes only a subset of the variables.

**Mixture Transition Distributions**

- **Properties**
  - In practice, we often construct the transition probabilities from a set of 'base' transitions $B_1,...,B_K$.
  - This can be achieved through a mixture distribution
  
  $T'(x',x) = \sum_{k=1}^{K} \alpha_k B_k(x',x)$
  
  where $k$ labels the members of the set of possible transitions considered.

- **Note**
  - Evaluation of acceptance criterion does not require normalizing constant $Z_T$.
  - When the proposal distributions are symmetric, Metropolis-Hastings reduces to the standard Metropolis algorithm.

**MCMC – Metropolis-Hastings Algorithm**

- **Metropolis-Hastings Algorithm**
  - Generalization: Proposal distribution not required to be symmetric.
  - The new candidate sample $x'$ is accepted with probability
  
  $A(x',x) = \min \left\{ 1, \frac{p(x') q_k(x|x')}{p(x) q_k(x'|x)} \right\}$
  
  where $k$ labels the members of the set of possible transitions considered.

- **Random Walks**
  - Consider a state space consisting of the integers $z \in \mathbb{Z}$ with initial state $z(1) = 0$ and transition probabilities
  
  $p(z^{(r+1)} = z(1)) = 0.5$
  
  $p(z^{(r+1)} = z(r) + 1) = 0.25$
  
  $p(z^{(r+1)} = z(r) - 1) = 0.25$

- **Analysis**
  - Expected state at time $\tau$:
    
    $E[z^{(\tau)}] = 0$
    
    $E[(z^{(\tau)})^2] = \tau/2$
  
  - After $\tau$ steps, the random walk has only traversed a distance that is on average proportional to $\sqrt{\tau}$.

- **Remark**
  - Central goal in MCMC is to avoid random walk behavior!

**MCMC – Metropolis-Hastings Algorithm**

- **Properties**
  - We can show detailed balance:
  
  $A(x',x) = \min \left\{ 1, \frac{p(x') q_k(x|x')}{p(x) q_k(x'|x)} \right\}$
  
  $\hat{p}(x) q_k(x'|x) A_k(x',x) = \min \left\{ \hat{p}(x) q_k(x|x'), \hat{p}(x') q_k(x'|x) \right\}$
  
  $\hat{p}(x) = T'(x',x)$
  
  Note: This is wrong in the Bishop book!

**Random Walks**

- **Example: Random Walk behavior**
  - Consider a state space consisting of the integers $z \in \mathbb{Z}$ with initial state $z(1) = 0$ and transition probabilities
  
  $p(z^{(r+1)} = z(1)) = 0.5$
  
  $p(z^{(r+1)} = z(r) + 1) = 0.25$
  
  $p(z^{(r+1)} = z(r) - 1) = 0.25$

- **Analysis**
  - Expected state at time $\tau$:
    
    $E[z^{(\tau)}] = 0$
    
    $E[(z^{(\tau)})^2] = \tau/2$
  
  - After $\tau$ steps, the random walk has only traversed a distance that is on average proportional to $\sqrt{\tau}$.

- **Schematic illustration**
  - For continuous state spaces, a common choice of proposal distribution is a Gaussian centered on the current state.

- **Remark**
  - Large variance: rejection rate will be high for complex problems.
  - The scale $\rho$ of the proposal distribution should be as large as possible without incurring high rejection rates.
  - $\rho$ should be of the same order as the smallest length scale $\sigma_{min}$.

- **Properties**
  - Large variance: rejection rate will be high for complex problems.
  - The scale $\rho$ of the proposal distribution should be as large as possible without incurring high rejection rates.
  - $\rho$ should be of the same order as the smallest length scale $\sigma_{min}$.

- **MCMC – Metropolis-Hastings Algorithm**
  - Central goal in MCMC is to avoid random walk behavior!
Gibbs Sampling

- **Approach**
  - MCMC algorithm that is simple and widely applicable.
  - May be seen as a special case of Metropolis-Hastings.

- **Idea**
  - Sample variable-wise: replace \( z_i \) by a value drawn from the distribution \( p(z_i | z_j) \).
  - This means we update one coordinate at a time.
  - Repeat procedure either by cycling through all variables or by choosing the next variable.

- **Properties**
  - The factor that determines the acceptance probability in the Metropolis-Hastings is determined by
    - \( A(z', z) = \frac{p(z')q_i(z'|z)}{p(z)q_i(z|z')} \)
    - \( q_i(z'|z) = p(z'|z) = p(z_i|z_j)p(z_j|z) \)
    - i.e. we get an algorithm which always accepts!
  - If you can compute (and sample from) the conditionals, you can apply Gibbs sampling.
  - The algorithm is completely parameter free.
  - Can also be applied to subsets of variables.

- **Example**
  - 20 iterations of Gibbs sampling on a bivariate Gaussian.

- **Discussion**
  - Gibbs sampling benefits from few free choices and convenient features of conditional distributions:
    - Conditionals with a few discrete settings can be explicitly normalized:
      - \( p(z_i | x_{fi}) = \frac{p(z_i, x_{fi})}{\sum_{x_{fi}} p(z_i, x_{fi})} \)
    - Continuous conditionals are often only univariate.
    - In case of graphical models, the conditional distributions depend only on the variables in the corresponding Markov blankets.

- **How Should We Run MCMC?**
  - Arbitrary initialization means starting iterations are bad
    - Discard a "burn-in" period.
  - How do we know if we have run for long enough?
    - You don’t. That’s the problem.
  - The samples are not independent
    - Solution 1: Keep only every \( M \)th sample ("thinning").
    - Solution 2: Keep all samples and use the simple Monte Carlo estimator on MCMC samples
      - It is consistent and unbiased if the chain has "burned in".
    - Use thinning only if computing \( f(x) \) is expensive.
  - For opinion on thinning, multiple runs, burn in, etc.
Summary: Approximate Inference

- Exact Bayesian Inference often intractable.
- Rejection and Importance Sampling
  - Generate independent samples.
  - Impractical in high-dimensional state spaces.
- Markov Chain Monte Carlo (MCMC)
  - Simple & effective (even though typically computationally expensive).
  - Scales well with the dimensionality of the state space.
  - Issues of convergence have to be considered carefully.
- Gibbs Sampling
  - Used extensively in practice.
  - Parameter free
  - Requires sampling conditional distributions.

References and Further Reading

- Sampling methods for approximate inference are described in detail in Chapter 11 of Bishop’s book.
- Another good introduction to Monte Carlo methods can be found in Chapter 29 of MacKay’s book (also available online: http://www.inference.phy.cam.ac.uk/mackay/itprnn/book.html)

Christopher M. Bishop
Pattern Recognition and Machine Learning
Springer, 2006

David MacKay
Information Theory, Inference, and Learning Algorithms
Cambridge University Press, 2003