CPSC 540: Machine Learning

Topic Models, Metropolis-Hastings

Mark Schmidt

University of British Columbia

Winter 2017

Admin

- Assignment 5:
  - Due Monday, 1 late day for Wednesday, 2 for following Monday.

- No office hours Tuesday this week, but I'll be there Friday.
- No office hours Friday next week, but I'll be there Tuesday.
- Project description posted on Piazza.
- Final is here on April 25th at 3:30.

- Bonus lecture on April 10th (same time and place) or just a long last lecture.

Outline

Motivation for Topic Models

We want a model of the "factors" making up a set of documents.

- In this context, latent-factor models are called topic models.

Suppose you have the following set of sentences:

- I like to eat broccoli and bananas.
- I ate a banana and spinach smoothie for breakfast.
- Chinchillas and kittens are cute.
- My sister adopted a kitten yesterday.
- Look at this cute hamster munching on a piece of broccoli.

What is latent Dirichlet allocation? It's a way of automatically discovering **topics** that these sentences contain. For example, given these sentences and asked for 2 topics, LDA might produce something like

- **Sentences 1 and 2**: 100% Topic A
- **Sentences 3 and 4**: 100% Topic B
- **Sentence 5**: 60% Topic A, 40% Topic B
- **Topic A**: 30% broccoli, 15% bananas, 10% breakfast, 10% munching, ... (at which point, you could interpret topic A to be about food)
- **Topic B**: 20% chinchillas, 20% kittens, 20% cute, 15% hamster, ... (at which point, you could interpret topic B to be about cute animals)

http://blog.echen.me/2011/08/22/introduction-to-latent-dirichlet-allocation

- "Topics" could be useful for things like searching for relevant documents.

Class Approach: Latent Semantic Indexing

- Classic methods are based on scores like TF-IDF:
  1. Term frequency: probability of a word occuring within a document.
     - E.g., 7% of words in document i are "the" and 2% of the words are "LeBron".
  2. Document frequency: probability of a word occuring across documents.
     - E.g., 100% of documents contain "the" and 0.01% have "LeBron".
  3. TF-IDF: measures like (term frequency)*log $1/$(document frequency).

- Latent semantic indexing (LSI) topic model:
  1. Summarize each document by its TF-IDF values.
  2. Run a latent-factor model like PCA or NMF on the matrix.
  3. Treat the latent factors as the "topics".

Modern Approach: Latent Dirichlet Allocation

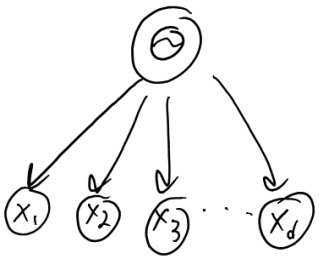
- LSI has largely been replace by latent Dirichlet allocation (LDA).
  - Hierarchical Bayesian model of all words in a document.
- The most cited ML paper from the last 15 years?
- LDA has several components, we'll build up to it by parts.
  - We'll assume all documents have d words and word order doesn't matter.

Model 1: Categorical Distribution of Words

- Base model: each word x_j comes from a categorical distribution.

$$p(x_j = \text{“the”}) = \theta_{\text{“the”}} \quad \text{where} \quad \theta_{\text{word}} \geq 0 \quad \text{and} \quad \sum_{\text{word}} \theta_{\text{word}} = 1.$$

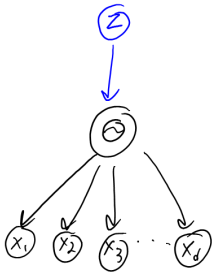
- So to generate a document with d words:
  - Sample d words from the categorical distribution.



- Drawback: misses that dcouments are about different “topics”.

Model 2: Mixture of Categorical Distributions

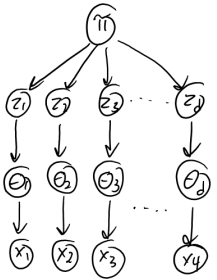
- To represent "topics", we'll use a mixture model.
  - Each mixture has its own categorical distribution over words.
  - E.g., the "basketball" mixture will have higher probability of "LeBron".
  - Can be fit with expectation maximization.
- So to generate a document with d words:
  - Sample a topic z from a categorical distribution.
  - Sample d word categorical distribution z.



- Drawback: misses that documents may be about more than one topics.

Model 3: Multi-Topic Mixture of Categorical

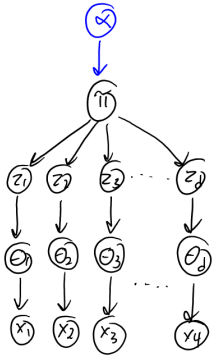
- Our third model introduces a new vector of "topic proportions" π.
  - Gives percentage of each topic that makes up the document.
    - E.g., 80% basketball and 20% politics.
  - Called probabilistic latent semantic indexing (PLSI).
- So to generate a document with d words given topic proportions π:
  - Sample d topics from π.
  - Sample a word from each sampled categorical distribution z.



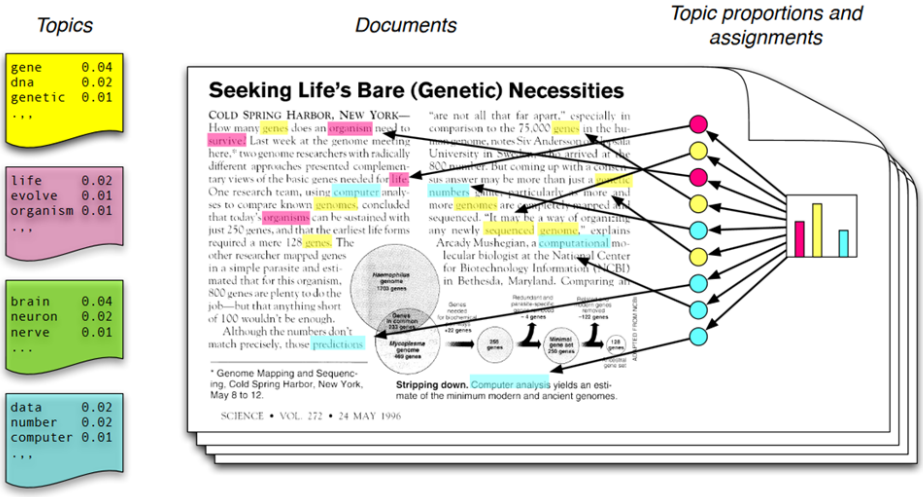
- Drawback: how do we compute π for a new document?.

Model 4: Latent Dirichlet Allocation

- Latent Dirichlet allocation (LDA) puts a prior on topic proportions.
  - Conjugate prior for categorical is Dirichlet distribution.
- So to generate a document with d words given Dirichlet prior:
  - Sample mixture proportions π from the Dirichlet prior.
  - Sample d topics from π.
  - Sample a word from each sampled categorical distribution z.

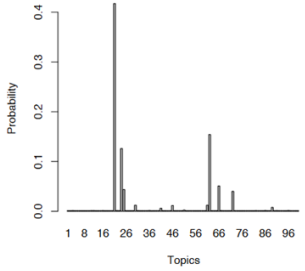


Latent Dirichlet Allocation Illustration



http://menome.com/wp/wp-content/uploads/2014/12/Blei2011.pdf

Latent Dirichlet Allocation Example



"Genetics"	"Evolution"	"Disease"	"Computers"
human	evolution	disease	computer
genome	evolutionary	host	models
dna	species	bacteria	information
genetic	organisms	diseases	data
genes	life	resistance	computers
sequence	origin	bacterial	system
gene	biology	new	network
molecular	groups	strains	systems
sequencing	phylogenetic	control	model
map	living	infectious	parallel
information	diversity	malaria	methods
genetics	group	parasite	networks
mapping	new	parasites	software
project	two	united	new
sequences	common	tuberculosis	simulations

Figure 2: **Real inference with LDA.** We fit a 100-topic LDA model to 17,000 articles from the journal *Science*. At left is the inferred topic proportions for the example article in Figure 1. At right are the top 15 most frequent words from the most frequent topics found in this article.

http://menome.com/wp/wp-content/uploads/2014/12/Blei2011.pdf

Latent Dirichlet Allocation Example

Figure 3: A topic model fit to the *Yale Law Journal*. Here there are twenty topics (the top eight are plotted). Each topic is illustrated with its top most frequent words. Each word's position along the x-axis denotes its specificity to the documents. For example "estate" in the first topic is more specific than "tax."

http://menome.com/wp/wp-content/uploads/2014/12/Blei2011.pdf

Latent Dirichlet Allocation Example

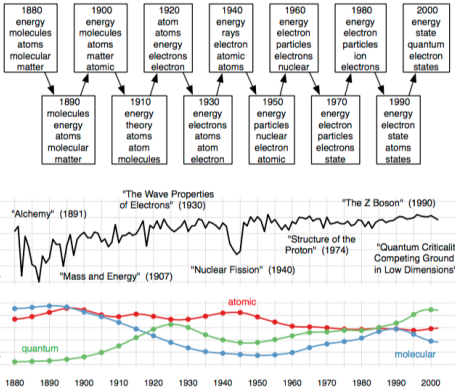
Non-Ailment Topics						
TV & Movies	**Games & Sports**	**School**	**Conversation**	**Family**	**Transportation**	**Music**
watch	killing	ugh	ill	mom	home	voice
watching	play	class	ok	shes	car	hear
tv	game	school	haha	dad	drive	feelin
killing	playing	read	ha	says	walk	lil
movie	win	test	fine	hes	bus	night
seen	boys	doing	yeah	sister	driving	bit
movies	games	finish	thanks	tell	trip	music
mr	fight	reading	hey	mum	ride	listening
watched	lost	teacher	thats	brother	leave	listen
hi	team	write	xd	thinks	house	sound

Ailments						
	Influenza-like Illness	**Insomnia & Sleep Issues**	**Diet & Exercise**	**Cancer & Serious Illness**	**Injuries & Pain**	**Dental Health**
General Words	better	night	body	cancer	hurts	dentist
	hope	bed	pounds	help	knee	appointment
	ill	body	gym	pray	ankle	doctors
	soon	ill	weight	awareness	hurt	tooth
	feel	tired	lost	diagnosed	neck	teeth
	feeling	work	workout	prayers	ouch	appt
	day	day	lose	died	leg	wisdom
	flu	hours	days	family	arm	eye
	thanks	asleep	legs	friend	fell	going
	xx	morning	week	shes	left	went
Symptoms	sick	sleep	sore	cancer	pain	infection
	sore	headache	throat	breast	sore	pain
	throat	fall	pain	lung	head	mouth
	fever	insomnia	aching	prostate	foot	ear
	cough	sleeping	stomach	sad	feet	sinus
Treatments	hospital	sleeping	exercise	surgery	massage	surgery
	surgery	pills	diet	hospital	brace	braces
	antibiotics	caffeine	dieting	treatment	physical	antibiotics
	fluids	pill	exercises	heart	therapy	eye
	paracetamol	tylenol	protein	transplant	crutches	hospital

http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0103408

Discussion of Topic Models

- There are *many* extensions of LDA:
  - We can put prior on the number of words (like Poisson).
  - Hierarchical topic models learn hierarchies of topics.
  - Can be combined with Markov models to capture word and/or topic dependences.



http://menome.com/wp/wp-content/uploads/2014/12/Blei2011.pdf

Discussion of Topic Models

- There are *many* extensions of LDA:
  - We can put prior on the number of words (like Poisson).
  - Hierarchical topic models learn hierarchies of topics.
  - Can be combined with Markov models to capture word and/or topic dependences.
  - Now being applied beyond text, like "cancer mutation signatures":
  - Recent work on representing considers"word2vec" representations (bonus slides).

http://journals.plos.org/plosgenetics/article?id=10.1371/journal.pgen.1005657

Discussion of Topic Models

- Topic models for analyzing musical keys:

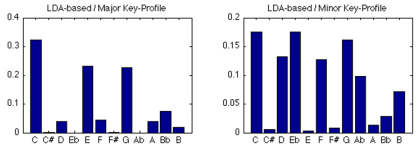


Figure 2: The C major and C minor key-profiles learned by our model, as encoded by the β matrix. Resulting key-profiles are obtained by transposition.

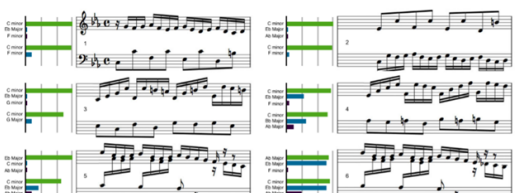


Figure 3: Key judgments for the first 6 measures of Bach's Prelude in C minor, WTC-II. Annotations for each measure show the top three keys (and relative strengths) chosen for each measure. The top set of three annotations are judgments from our LDA-based model; the bottom set of three are from human expert judgments [3].

http://cseweb.ucsd.edu/~dhu/docs/nips09_abstract.pdf

Outline

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).

Need for Approximate Integration

- Bayesian models allow things that aren't possible in other frameworks:
  - Optimize the regularizer (empirical Bayes).
  - Relax IID assumption (hierarchical Bayes).
  - Have clustering happen on multiple leves (topic models).
- But posterior often doesn't have a closed-form expression.
  - We don't just want to flip coins and multiply Gaussians.
- We once again need approximate inference:
  1. Variational methods.
  2. Monte Carlo methods.
- Classic ideas from statistical physics, that revolutionized Bayesian stats/ML.

Variational Inference vs. Monte Carlo

Two main strategies for approximate inference:

1. Variational methods:
   - Approximate p with "closest" distribution q from a tractable family,

$$p(x) \approx q(x).$$

   - Turns inference into optimization.
     - Called variational Bayes (some material in bonus slides).
2. Monte Carlo methods:
   - Approximate p with empirical distribution over samples,

$$p(x) \approx \frac{1}{n} \sum_{i=1}^{n} \mathcal{I}[x^i = x].$$

   - Turns inference into sampling.

Conjugate Graphical Models: Ancestral and Gibbs Sampling

- For conjugate DAGs, we can use ancestral sampling for unconditional sampling.
- Examples:
  - For LDA, sample π then sample the z_j then sample the x_j.
  - For HMMs, sample the hidden z_j then sample the x_j.
- We can also often use Gibbs sampling as an approximate sampler.
  - If neighbours are conjugate in UGMs.
  - To generate conditional samples in conjugate DAGs.
- However, without conjugacy our inverse transform trick doesn't work.
  - We can't even sample from the 1D conditionals with this method.

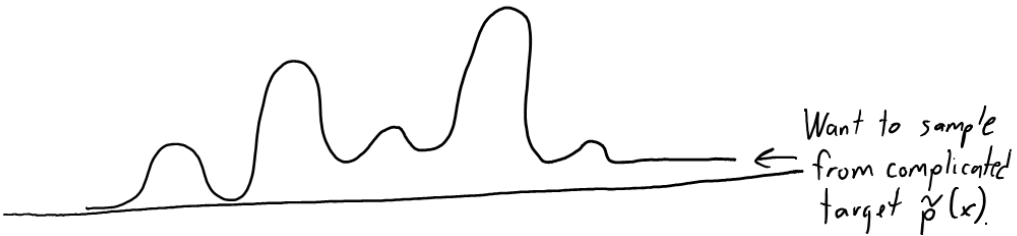
Beyond Inverse Transform and Conjugacy

- We want to use simple distributions to sample from complex distributions.
- Two common strategies are rejection sampling and importance sampling.

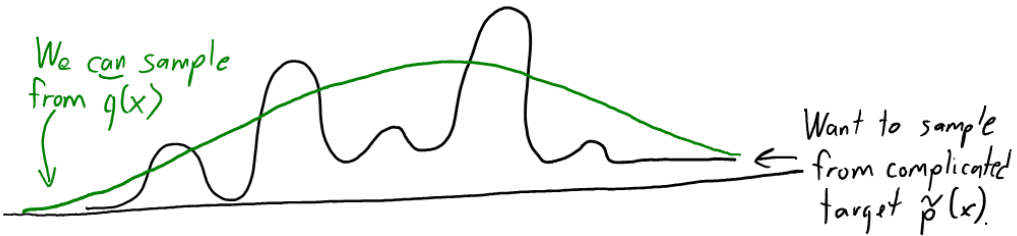
- We've previously seen rejection sampling to do conditional sampling:
  - Example: sampling from a Gaussian subject to $x \in [-1, 1]$.

  - Generate unconditional samples, throw out the ones that aren't in $[-1, 1]$.

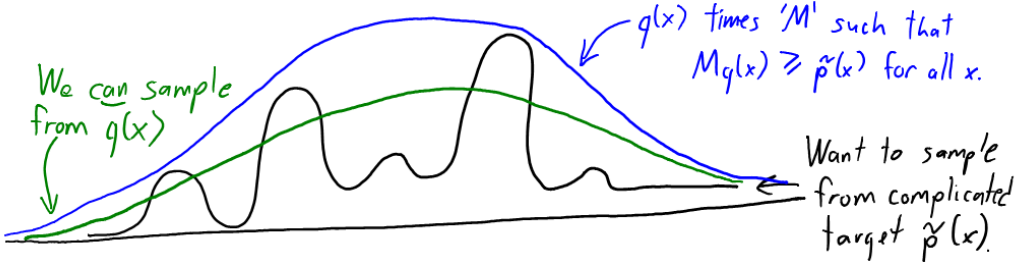
General Rejection Sampling Algorithm



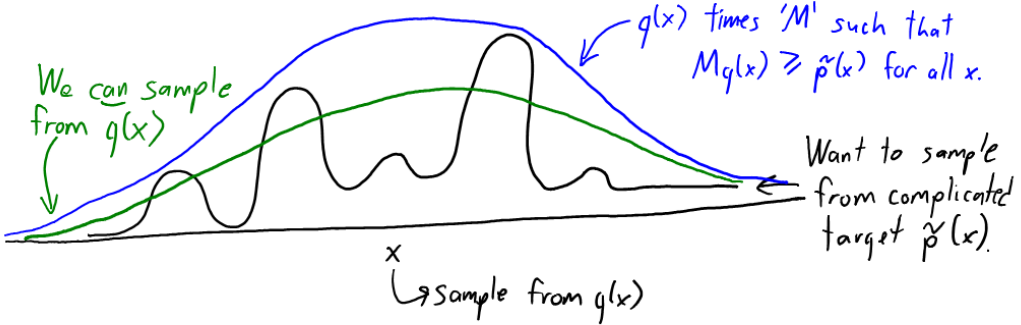
General Rejection Sampling Algorithm



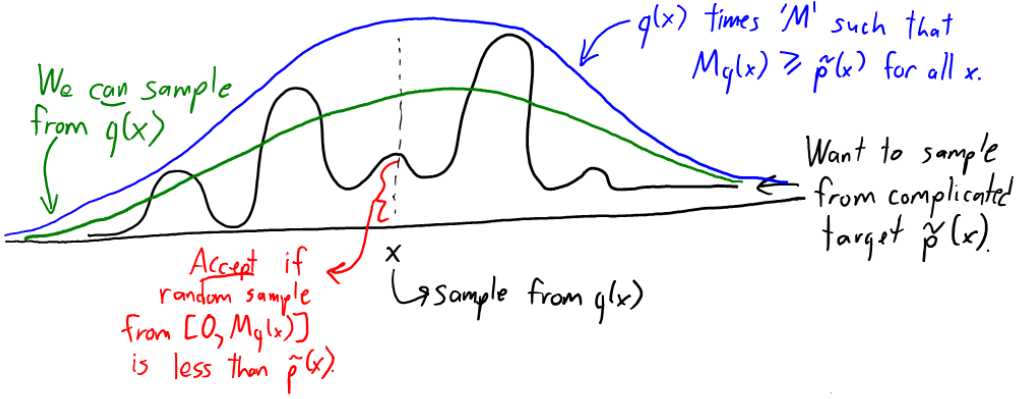
General Rejection Sampling Algorithm



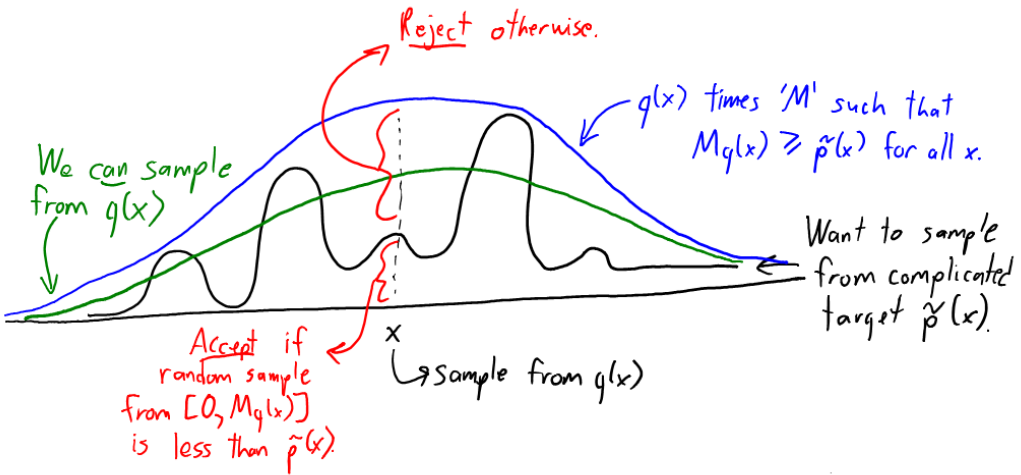
General Rejection Sampling Algorithm



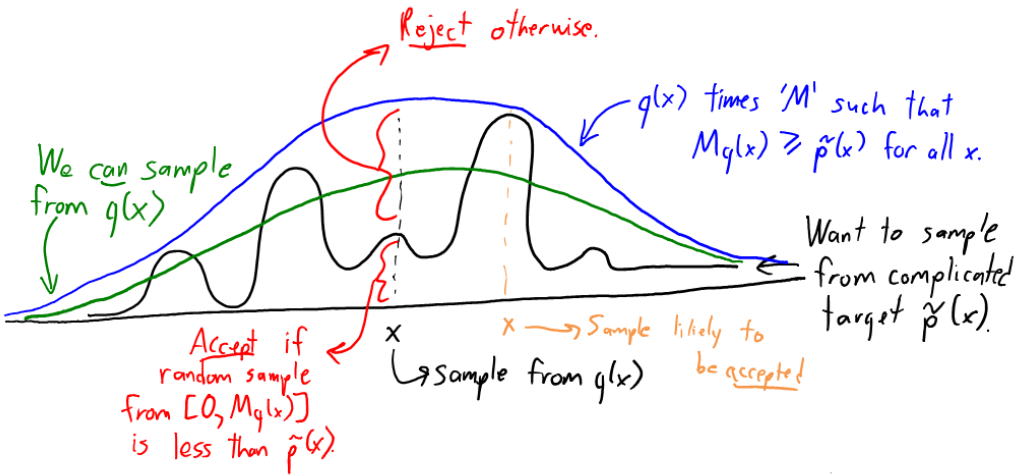
General Rejection Sampling Algorithm



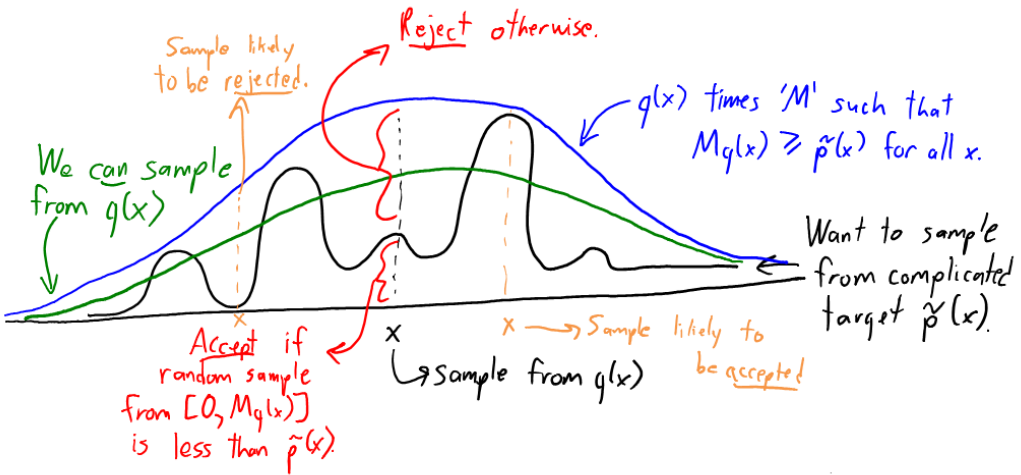
General Rejection Sampling Algorithm



General Rejection Sampling Algorithm



General Rejection Sampling Algorithm



General Rejection Sampling Algorithm

- Ingredients of a more general rejection sampling algorithm:
  1. Ability to evaluate unnormalized $\tilde{p}(x)$,

$$p(x) = \frac{\tilde{p}(x)}{Z}.$$

  2. A distribution q that is easy to sample from.
  3. An upper bound M on $\tilde{p}(x)/q(x)$.

- Rejection sampling algorithm:
  1. Sample x from $q(x)$.
  2. Sample u from $\mathcal{U}(0, 1)$.
  3. Keep the sample if $u \leq \frac{\tilde{p}(x)}{Mq(x)}$.
- The accepted samples will be from $p(x)$.

General Rejection Sampling Algorithm

- We can use general rejection sampling for:
  - Sample from Gaussian q to sample from student t.
  - Sample from prior to sample from posterior ($M = 1$),

$$p(\theta|x) = \underbrace{p(x|\theta)}_{\leq 1} p(\theta).$$

- Drawbacks:
  - You may reject a large number of samples.
    - Most samples are rejected for high-dimensional complex distributions.
  - You need to know M.

- Extension in 1D for convex $-\log p(x)$:
  - Adaptive rejection sampling refines piecewise-linear q after each rejection.

Importance Sampling

- Importance sampling is a variation that accepts all samples.
  - Key idea is similar to EM,

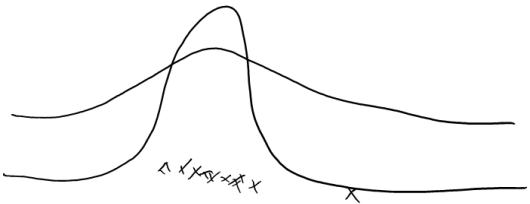
$$\begin{aligned}\mathbb{E}_p[f(x)] &= \sum_x p(x)f(x) \\ &= \sum_x q(x)\frac{p(x)f(x)}{q(x)} \\ &= \mathbb{E}_q\left[\frac{p(x)}{q(x)}f(x)\right],\end{aligned}$$

    and similarly for continuous distributions.

  - We can sample from q but reweight by $p(x)/q(x)$ to sample from p.
  - Only assumption is that q is non-zero when p is non-zero.
  - If you only know unnormalized $\tilde{p}(x)$, a variant gives approximation of Z.

Importance Sampling

- As with rejection sampling, only efficient if q is close to p.
- Otherwise, weights will be huge for a small number of samples.
  - Even though unbiased, variance will be huge.



- In high-dimensions, this doesn't tend to work well.

Outline

Limitations of Simple Monte Carlo Methods

- The basic ingredients of our previous sampling methods:
  - Inverse CDF, rejection sampling, importance sampling.
  - Sampling in higher-dimensions: ancestral sampling, Gibbs sampling.
- These work well in low dimensions or for posteriors with analytic properties.

- But we want to solve high-dimensional integration problems in other settings:
  - Deep belief networks, Boltzmann machines.
  - Bayesian graphical models and Bayesian neural networks.

- Our previous methods tend not to work in complex situations:
  - Inverse CDF may not be available.
  - Conditionals needed for ancestral/Gibbs sampling may be hard to compute.
  - Rejection sampling tends to reject almost all samples.
  - Importance sampling tends gives almost zero weight to all samples.

Dependent-Sample Monte Carlo Methods

- We want an algorithm that gets better over time.
- Two main strategies for generating dependent samples:
  - 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`
    - AKA sequential importance sampling, annealed importance sampling, particle filter.
  - Markov chain Monte Carlo (MCMC).
    - Design Markov chain whose stationary distribution is the posterior.
- These are the main tools to sample from high-dimensional distributions.

Markov Chain Monte Carlo

- We've previously discussed Markov chain Monte Carlo (MCMC).
  1. Based on generating samples from a Markov chain q.
  2. Designed so stationary distribution π of q is target distribution p.
- If we run the chain long enough, it gives us samples from p.
- Gibbs sampling is an example of an MCMC method.
  - Sample x_j conditioned on all other variables x_{-j}.

Limitations of Gibbs Sampling

- Gibbs sampling is nice because it has no parameters:
  - You just need to decide on the blocks and figure out the conditonals.
- 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 "10 Best of the 20th Century".

Metropolis Algorithm

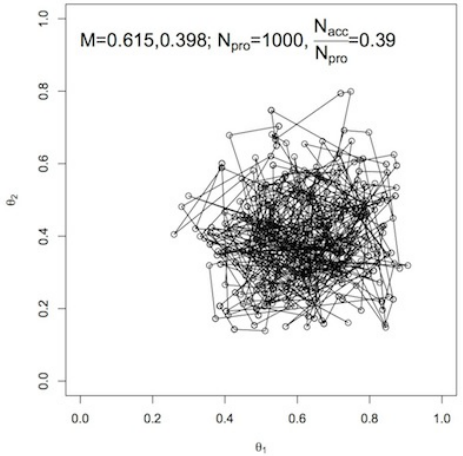
- The Metropolis algorithm for sampling from a continuous target $p(x)$:
  - Start from some x^0 and on iteration t:
    1. Add zero-mean Gaussian noise to x^t to generate $\tilde{x}^t$.
    2. Generate u from a $\mathcal{U}(0,1)$.
    3. Accept the sample and set $x^{t+1} = \tilde{x}^t$ if

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

       and otherwise reject the sample and set $x^{t+1} = x^t$.

  - A random walk, but sometimes rejecting steps that decrease probability:
    - A valid MCMC algorithm on continuous densities, but convergence may 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 $q_{ss'} = q(x^t = s' | x^{t-1} = s)$ is reversible if

$$\pi(s) q_{ss'} = \pi(s') q_{s's},$$

for some distribution π (called detailed balance).

- Assuming we reach stationary, reversibility implies π is stationary distribution,

$$\sum_s \pi(s) q_{ss'} = \sum_s \pi(s') q_{s's}$$

$$\sum_s \pi(s) q_{ss'} = \pi(s') \underbrace{\sum_s q_{ss'}}_{=1}$$

$$\sum_s \pi(s) q_{ss'} = \pi(s') \qquad \text{(stationary condition)}$$

- Metropolis is reversible (bonus slide) so has correct stationary distribution.

Metropolis-Hastings

- Metropolis-Hastings algorithms allows general proposal distribution q:
  - Value $q(\tilde{x}^t|x^t)$ is probability of proposing $\tilde{x}^t$.
  - Metropolis algorithm is special case where q is zero-mean Gaussian.

- It accepts a proposed $\tilde{x}^t$ if

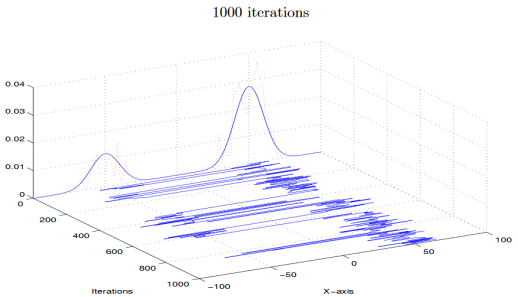
$$u \leq \frac{\tilde{p}(\tilde{x}^t)q(x^t|\tilde{x}^t)}{\tilde{p}(x^t)q(\tilde{x}^t|x^t)},$$

  where extra terms ensure reversibility 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 it's often not the best choice:
  - You can make performance much better/worse with an appropriate q.

Metropolis-Hastings

Metropolis-Hastings for sampling from mixture of Gaussians:



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

- With a random walk q we may get stuck in one mode.
- We could have proposal be mixture between random walk and "mode jumping".

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".

Advanced Monte Carlo Methods

- Some other more-powerful MCMC methods:
  - Block Gibbs sampling improves over single-variable Gibb sampling.
  - Collapsed Gibbs sampling (Rao-Blackwellization): integrate out variables that are not of interest.
    - E.g., integrate out hidden states in Bayesian hidden Markov model.
    - E.g., integrate over different components in topic models.
    - Provably decreases variance of sampler (if you can do it, you should do it).
  - Auxiliary-variable sampling: Introduce variables to sample bigger blocks:
    - E.g., introduce z variables in mixture models.
    - Also used in Bayesian logistic regression.

Advanced Monte Carlo Methods

- Population MCMC:
  - Run multiple MCMC methods, each having different "move" size.
  - Large moves do exploration and small moves refine good estimates.
- Combinations of variational inference and stochastic methods:
  - Variational MCMC: Metropolis-Hastings where variational q can make proposals.
  - Stochastic variational inference (SVI): variational methods using stochastic gradient.

Summary

- Latent Dirichlet allocation: factor/topic model for discrete data like text.
- Rejection sampling: generate exact samples from complicated distributions.
- Importance sampling: reweights samples from the wrong distribution.
- Markov chain Monte Carlo generates a sequence of *dependent samples*:
  - But asymptotically these samples come from the posterior.
- Metropolis-Hastings allowing arbitrary "proposals".
  - With good proposals works much better than Gibbs sampling.

- The remaining hottest topics in machine learning.

Bonus Slide: Word2Vec

- In natural language, we often represent words by an index.
  - E.g., "cat" is word 124056.
- But this may be innefficient:
  - Should "cat" and "kitten" share parameters in some way?
- We want a latent-factor representation of words.
  - Closeness in latent space should indicate similarity, distances could represent meaning?
- We could use PCA, LDA, and so on.
- But recent "word2vec" approach is getting a lot of popularity...

Bonus Slide: Word2Vec

- Two variations of word2vec:
  1. Try to predict word from surrounding words ("continuous bag of words").
  2. Try to predict surrounding words from word ("skip-gram").

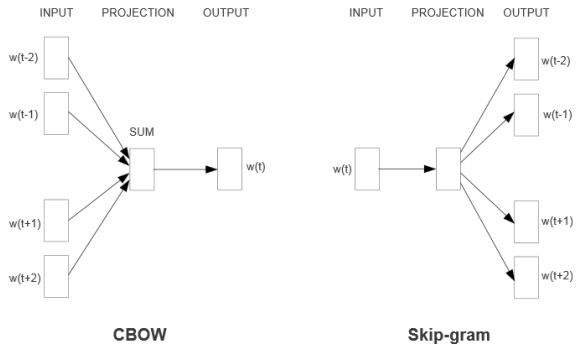


Figure 1: New model architectures. The CBOW architecture predicts the current word based on the context, and the Skip-gram predicts surrounding words given the current word.

https://arxiv.org/pdf/1301.3781.pdf

Bonus Slide: Word2Vec

- In both cases, each word i is represented by a vector z^i.
- We optimize likelihood of word vectors z^i under the model

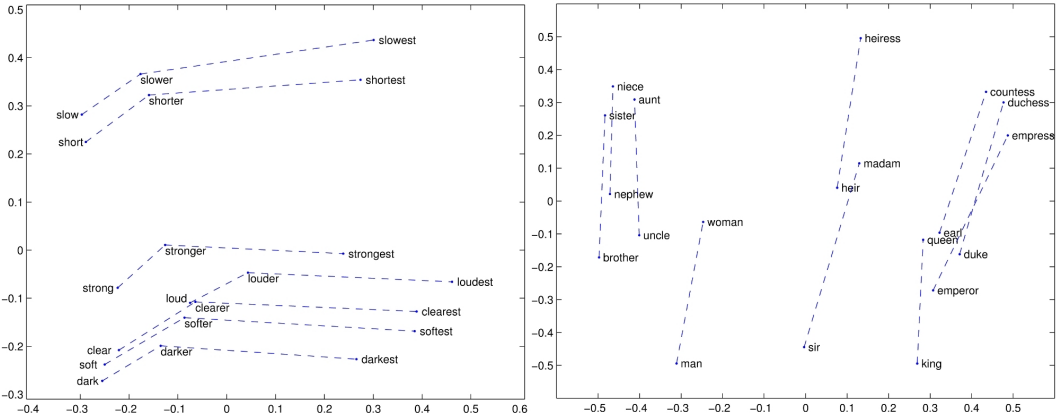
$$p(x^i|x^j) \propto \exp((z^i)^T z^j),$$

  and we usually assume everything is independent while training.

- The denominator sums over all words (CBOW) or combinations of words (skip-gram), so people have come up with tricks:
  - Hierarchical softmax.
  - Negative sampling.

Bonus Slide: Word2Vec

MDS visualization of a set of related words.



http://sebastianruder.com/secret-word2vec

Distances between vectors can represent semantic relationships.

Bonus Slide: Word2Vec

Table 8: *Examples of the word pair relationships, using the best word vectors from Table 4 (Skip-gram model trained on 783M words with 300 dimensionality).*

Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

Table 8 shows words that follow various relationships. We follow the approach described above: the relationship is defined by subtracting two word vectors, and the result is added to another word. Thus for example, *Paris - France + Italy = Rome*. As it can be seen, accuracy is quite good, although

https://arxiv.org/pdf/1301.3781.pdf

Subtracting word vectors to find related words.

Laplace Approximation

- Simple variational method is Laplace approximation:
  - Find 'x' that maximizes p(x):

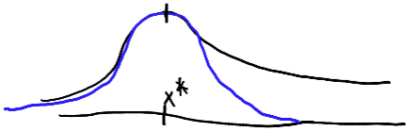
$$f(x) = -\log p(x) \qquad \min_x f(x)$$

  - Choose 'q' so that –log q(x) and –log p(x) have same Taylor expansion at x^*:

$$\text{We want } -\log q(x) = f(x^*) + \underbrace{\nabla f(x^*)}_{=0}^T (x - x^*) + \frac{1}{2}(x - x^*)^T \nabla^2 f(x^*)(x - x^*)$$

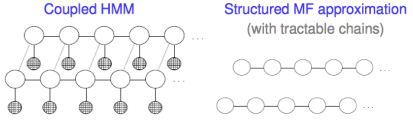
$$= f(x^*) + \frac{1}{2}(x - x^*)^T \nabla^2 f(x)(x - x^*)$$

$$\text{So } q \sim N(x^*, \nabla^2 f(x)^{-1})$$

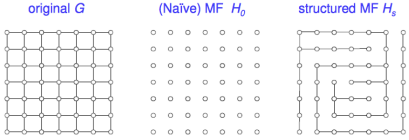


Bonus Slide: Structure Mean Field

- Mean field for Bayesian models has same update.
  - Common to use a Gaussian other conjugate model to approximate non-conjugate.
- Common variant is structured mean field: q function includes many original edges.



http://courses.cms.caltech.edu/cs155/slides/cs155-14-variational.pdf



http://courses.cms.caltech.edu/cs155/slides/cs155-14-variational.pdf

  - Original LDA article proposed a structured mean field approximation.
- Extension of loopy belief propagation for non-conjugate: expectation propagation.

Bonus Slide: Metropolis Algorithm Analysis

- Metropolis algorithm has $q_{ss'} > 0$ (sufficient to guarantee stationary distribution is unique and we reach it) and satisfies detailed balance with target distribution p,

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

- We can show this by defining transition probabilities

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

and observing that

$$\begin{aligned} p(s)q_{ss'} &= p(s)\min\left\{1, \frac{\tilde{p}(s')}{\tilde{p}(s)}\right\} = p(s)\min\left\{1, \frac{\frac{1}{Z}\tilde{p}(s')}{\frac{1}{Z}\tilde{p}(s)}\right\} \\ &= p(s)\min\left\{1, \frac{p(s')}{p(s)}\right\} = \min\left\{p(s), p(s')\right\} \\ &= p(s')\min\left\{1, \frac{p(s)}{p(s')}\right\} = p(s')q_{s's}. \end{aligned}$$