CPSC 440: Advanced Machine Learning

Topic Models and Variational Inference

Mark Schmidt

University of British Columbia

Winter 2022

Motivation for Topic Models

We want a model of the hidden "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.

Classic 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).
     - Seeing "LeBron" tells you a lot about document, seeing "the" tells you nothing.
- Many many many variations exist.
- TF-IDF features are very redundant.
  - Consider TF-IDF of "LeBron", "Durant", and "Giannis".
  - High values of these typically just indicate topic of "basketball".
  - Basically a weighted bag of words.
- We want to find latent factors ("topics") like "basketball".

Modern Approach: Latent Dirichlet Allocation

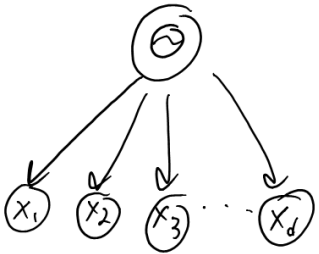
- Latent semantic indexing (LSI) topic model:
  1. Summarize each document by its TF-IDF values.
  2. Run a latent-factor model like PCA or NMF on the matrix.
  3. Treat the latent factors as the "topics".
- LSI has largely been replace by latent Dirichlet allocation (LDA).
  - Hierarchical Bayesian model of all words in a document.
    - Still ignores word order.
    - Tries to explain all words in terms of topics.
- The most cited ML paper in the 00s?
- 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 the same categorical distribution.

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

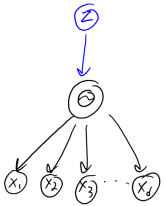
- So to generate a document with d words:
  - Sample d words from the categorical distribution.



- Drawback: misses that documents are about different “topics”.
  - We want the word distribution to depend on the “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".
- So to generate a document with d words:
  - Sample a topic z from a categorical distribution.
  - Sample d words from categorical distribution z.

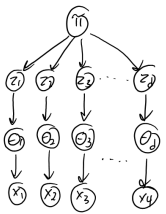


- Similar to a mixture of independent categorical distributions.
  - But we tie categorical distribution across the d variables, given cluster.
- Drawback: misses that documents may be about more than one topics.

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 z_j from categorical distribution π.
  - Sample a word for each z_j from corresponding categorical distribution.

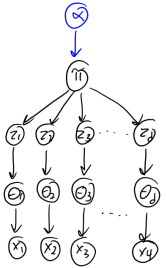


- Similar to HMM where each "time" has own cluster (but no Markov assumption).

Model 4: Latent Dirichlet Allocation

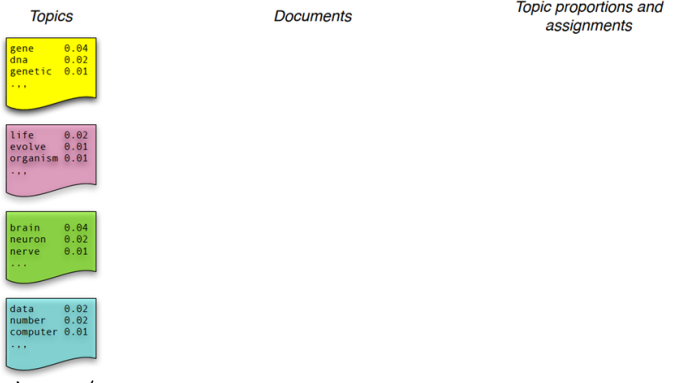
- Latent Dirichlet allocation (LDA) puts a prior on topic proportions.
  - Conjugate prior for categorical is Dirichlet distribution.

- So to generate a document with d words given Dirichlet prior:
  - Sample mixture proportions π from the Dirichlet prior.
  - Sample d topics z_j from categorical distribution π.
  - Sample a word for each z_j from corresponding categorical distribution.



- This is the generative model, typically used with MCMC or variational methods.

Latent Dirichlet Allocation (LDA)



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

Latent Dirichlet Allocation (LDA)

1. Sample topic proportions θ from Dirichlet.

Each topic is like a "principal component" or "latent factor"

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

Latent Dirichlet Allocation (LDA)

1. Sample topic proportions θ from Dirichlet.
2. Sample 'd' topics z_j from θ.

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

Latent Dirichlet Allocation (LDA)

1. Sample topic proportions θ from Dirichlet.
2. Sample 'd' topics z_j from θ.
3. For each z_j sample a word based on frequencies for topic.

Topics

gene	0.04
dna	0.02
genetic	0.01
...	

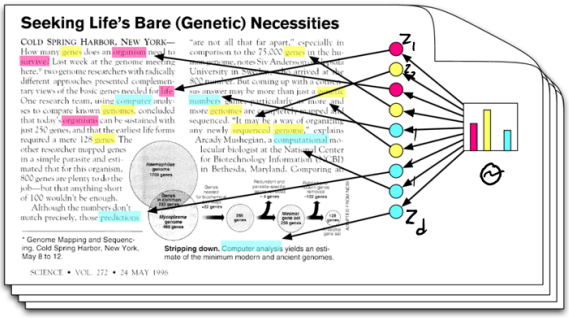
life	0.02
evolve	0.01
organism	0.01
...	

brain	0.04
neuron	0.02
nerve	0.01
...	

data	0.02
number	0.02
computer	0.01
...	

Documents

Topic proportions and assignments



Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 256 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predicted en-

"are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus may be more than just a genomic numbers game, particularly as more and more genomes are completed and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing ge-

* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

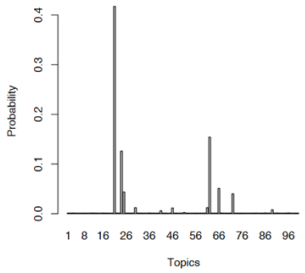
Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

SCIENCE • VOL. 272 • 24 MAY 1996

Each topic is like a "principal component" or "latent factor"

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

Health topics in social media:

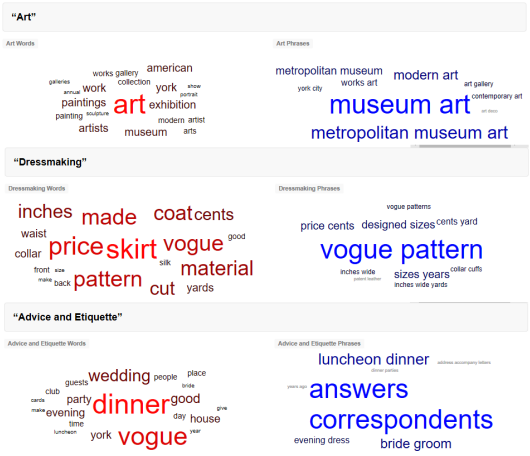
Non-Ailment Topics						
TV & Movies	**Games & Sports**	**School**	**Conversation**	**Family**	**Transportation**	**Music**
watch	killing	ugh	ill	mom	home	voice
watching	play	class	ok	shes	car	hear
tv	game	school	haha	dad	drive	feelin
killing	playing	read	ha	says	walk	lil
movie	win	test	fine	hes	bus	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

Latent Dirichlet Allocation Example

Three topics in 100 years of "Vogue" fashion magazine:



http://dh.library.yale.edu/projects/vogue/topics/

Discussion of Topic Models

- There are *many* extensions of LDA:
  - We can put prior on the number of words (like Poisson).
  - Correlated and hierarchical topic models learn dependencies between topics.

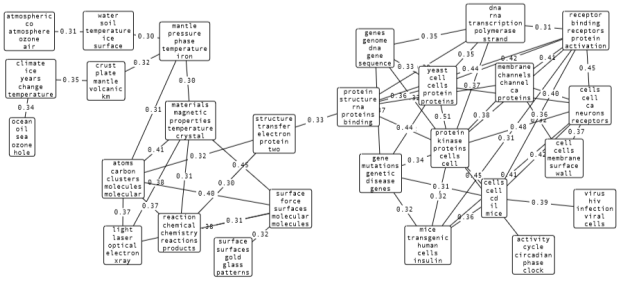
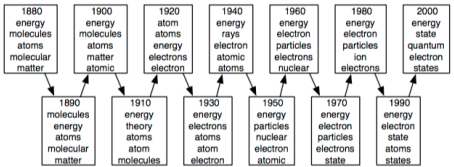


Figure 2: A portion of the topic graph learned from 15,744 OCR articles from *Science*. Each node represents a topic, and is labeled with the five most probable words from its distribution; edges are labeled with the correlation between topics.

http://people.ee.duke.edu/~lcarin/Blei2005CTM.pdf

Discussion of Topic Models

- There are *many* extensions of LDA:
  - We can put prior on the number of words (like Poisson).
  - Correlated and hierarchical topic models learn dependencies between topics.
  - Can be combined with Markov models to capture dependencies over time.



Discussion of Topic Models

- There are *many* extensions of LDA:
  - We can put prior on the number of words (like Poisson).
  - Correlated and hierarchical topic models learn dependencies between topics.
  - Can be combined with Markov models to capture dependencies over time.
  - Recent work on better word representations like"word2vec" (CPSC 340).
  - Now being applied beyond text, like "cancer mutation signatures":

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

Discussion of Topic Models

- Topic models for analyzing musical keys:

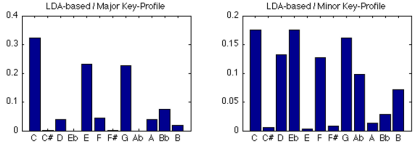


Figure 2: The C major and C minor key-profiles learned by our model, as encoded by the β matrix. Resulting key-profiles are obtained by transposition.

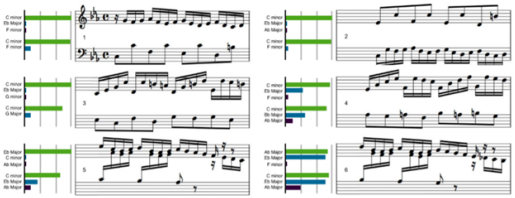


Figure 3: Key judgments for the first 6 measures of Bach's Prelude in C minor, WTC-II. Annotations for each measure show the top three keys (and relative strengths) chosen for each measure. The top set of three annotations are judgments from our LDA-based model; the bottom set of three are from human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

Monte Carlo Methods for Topic Models

- Nasty integrals in topic models:

Inference [edit]

See also: Dirichlet-multinomial distribution

Learning the various distributions (the set of topics, their associated word probabilities, the topic of each word, and the particular topic mixture of each document) is a problem of Bayesian inference. The original paper used a variational Bayes approximation of the posterior distribution;[1] alternative inference techniques use Gibbs sampling[6] and expectation propagation.[7]

Following is the derivation of the equations for collapsed Gibbs sampling, which means φs and θs will be integrated out. For simplicity, in this derivation the documents are all assumed to have the same length N. The derivation is equally valid if the document lengths vary.

According to the model, the total probability of the model is:

$$P(\boldsymbol{W}, \boldsymbol{Z}, \boldsymbol{\theta}, \boldsymbol{\varphi}; \alpha, \beta) = \prod_{i=1}^{K} P(\varphi_i; \beta) \prod_{j=1}^{M} P(\theta_j; \alpha) \prod_{t=1}^{N} P(Z_{j,t}|\theta_j) P(W_{j,t}|\varphi_{Z_{j,t}}),$$

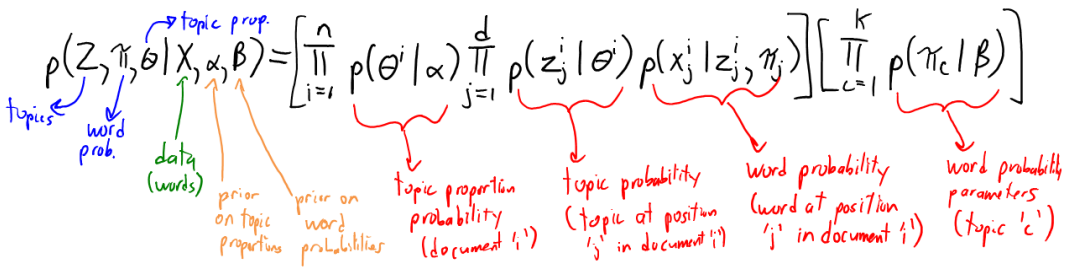
where the bold-font variables denote the vector version of the variables. First, $\boldsymbol{\varphi}$ and $\boldsymbol{\theta}$ need to be integrated out.

$$\begin{aligned} P(\boldsymbol{Z}, \boldsymbol{W}; \alpha, \beta) &= \int_{\boldsymbol{\theta}} \int_{\boldsymbol{\varphi}} P(\boldsymbol{W}, \boldsymbol{Z}, \boldsymbol{\theta}, \boldsymbol{\varphi}; \alpha, \beta)\, d\boldsymbol{\varphi}\, d\boldsymbol{\theta} \\ &= \int_{\boldsymbol{\varphi}} \prod_{i=1}^{K} P(\varphi_i; \beta) \prod_{j=1}^{M} \prod_{t=1}^{N} P(W_{j,t} \mid \varphi_{Z_{j,t}})\, d\boldsymbol{\varphi} \int_{\boldsymbol{\theta}} \prod_{j=1}^{M} P(\theta_j; \alpha) \prod_{t=1}^{N} P(Z_{j,t} \mid \theta_j)\, d\boldsymbol{\theta}. \end{aligned}$$

https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation

Monte Carlo Methods for Topic Models

- How do we actually *use* Monte Carlo for topic models?
- First we write out the posterior:



Monte Carlo Methods for Topic Models

- How do we actually *use* Monte Carlo for topic models?
- First we generate samples from the posterior:
  - With Gibbs sampling we alternate between:
    - Sampling topics given word probabilities and topic proportions.
    - Sampling topic proportions given topics and prior parameters α.
    - Sampling word probabilities given topics, words, and prior parameters β.
  - Have a burn-in period, use thinning, try to monitor convergence, and so on.
- Then we use posterior samples to do inference:
  - Distribution of topic proportions for sample i is frequency in samples.
  - To see if words come from same topic, check frequency in samples.

Outline

Need for Approximate Inference

- We have seen a variety of models where inference can be intractiable:
  - Bayesian logistic regression.
  - Markov chains with non-Gaussians continuous states.
  - Non-forest graphical models.
  - LDA topic modeling.
- Monte Carlo methods can solve these problems, but is so slow.
- Most common alternative is variational methods.

Monte Carlo vs. Variational Inference

Two main strategies for approximate inference:

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

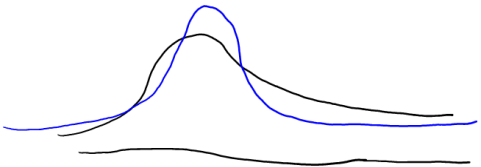
$$p(x) \approx q(x).$$

     - E.g., Gaussian, independent Bernoulli, or tree UGM.

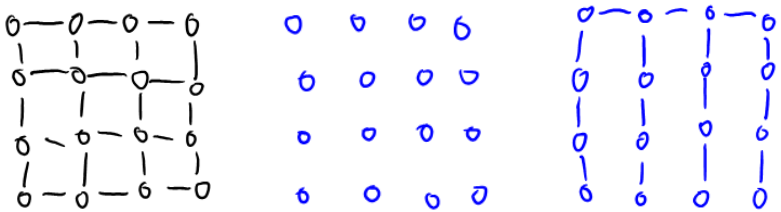
       (or mixtures of these simple distributions)
   - Turns inference into optimization.

Variational Inference Illustration

- Approximate non-Gaussian p by a Gaussian q:



- Approximate loopy UGM by independent distribution or tree-structed UGM:



- Variational methods try to find simple distribution q that is closest to target p.
  - This isn't consistent like MCMC, but can be very fast.

Kullback-Leibler (KL) Divergence

- How do we define "closeness" between a distribution p and q?
- A common measure is Kullback-Leibler (KL) divergence between p and q:

$$\mathrm{KL}(p \,||\, q) = \sum_x p(x) \log \frac{p(x)}{q(x)}.$$

  - Replace sum with integral for continuous families of q distributions.

- Also called information gain: "information lost when p is approximated by q".
  - If p and q are the same, we have $KL(p \,||\, q) = 0$ (no information lost).
  - Otherwise, $KL(p \,||\, q)$ grows as it becomes hard to predict p from q.
    - Note that KL is not commutative: we may have $KL(p \,||q) \neq KL(q \,||p)$.
- Unfortunately, this requires summing/integrating over p.
  - The problem we are trying to solve.

Minimizing Reverse KL Divergence

- Most variational methods minimize KL with arguments "reversed",

$$\mathrm{KL}(q \,||\, p) = \sum_x q(x) \log \frac{q(x)}{p(x)} = \sum_x q(x) \log \frac{q(x)}{\tilde{p}(x)} Z.$$

  which just swaps all p and q values in the definition.
  - Not intuitive: "how much information is lost when we approximate q by p".

- "Reverse" KL only needs unnormalized distribution $\tilde{p}$ and expectations over q.

$$\begin{aligned}\mathrm{KL}(q \,||\, p) &= \sum_x q(x) \log q(x) - \sum_x q(x) \log \tilde{p}(x) + \sum_x q(x) \log(Z) \\ &= \mathbb{E}_q[\log q(x)] - \mathbb{E}_q[\log \tilde{p}(x)] + \underbrace{\log(Z)}_{\text{const. in } q}.\end{aligned}$$

- By non-negativiy of KL this also gives a lower bound on $\log(Z)$ in terms of q.

$$\log(Z) \geq \mathbb{E}_q[\log \tilde{p}(x)] - \mathbb{E}_q[\log q(x)] \quad \text{("evidence lower bound" or ELBO).}$$

Coordinate Optimization: Mean Field Approximation

- Minimizing non-convex reverse KL is still difficulty due to $\mathbb{E}_q[\log \tilde{p}(x)]$ term.
  - But with appropriate q we can do coordinate optimization to decrease it.

- Consider minimizing reverse KL when q is a product of independent,

$$q(x) = \prod_{j=1}^{d} q_j(x_j),$$

where we choose q to be discrete or conjugate (usually Gaussian).

- If we fix q_{-j} and optimize the functional q_j we obtain (see Murphy's book)

$$q_j(x_j) \propto \exp\left(\mathbb{E}_{q_{-j}}[\log \tilde{p}(x)]\right),$$

which we can use to update q_j for a particular j.

Coordinate Optimization: Mean Field Approximation

- Each iteration we choose a j and set q based on mean (of neighbours),

$$q_j(x_j) \propto \exp\left(\mathbb{E}_{q_{-j}}[\log \tilde{p}(x)]\right).$$

- This improves the (non-convex) reverse KL on each iteration.

- Applying this update is called:
  - Mean field method (graphical models).
  - Variational Bayes (Bayesian inference).

3 Coordinate-Wise Algorithms

- Gibbs sampling is a coordinate-wise method for approximate sampling:
  - Choose a coordinate i to update.
  - Sample x_i keeping other variables fixed.
- ICM is a coordinate-wise method for approximate decoding (not covered):
  - Choose a coordinate i to update.
  - Maximize x_i keeping other variables fixed.
- Mean field is a coordinate-wise method for approximate marginalization:
  - Choose a coordinate i to update.
  - Update marginal $\underbrace{q_i(x_i)}_{\text{for all } x_i}$ keeping other variables fixed ($q_i(x_i)$ approximates $p_i(x_i)$).

3 Coordinate-Wise Algorithms

- Consider a pairwise UGM:

$$p(x_1, x_2, \ldots, x_d) \propto \left(\prod_{i=1}^{d} \phi_i(x_i) \right) \left(\prod_{(i,j) \in E} \phi_{ij}(x_i, x_j) \right),$$

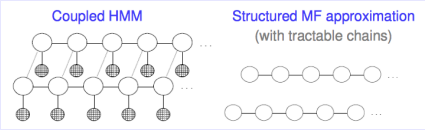
- ICM for updating a node i with 2 neighbours (j and k).
  1. Compute $M_i(x_i) = \phi_i(x_i)\phi_{ij}(x_i, x_j)\phi_{ik}(x_i, x_k)$ for all x_i.
  2. Set x_i to the largest value of $M_i(x_i)$.

- Gibbs for updating a node i with 2 neighbours (j and k).
  1. Compute $M_i(x_i) = \phi_i(x_i)\phi_{ij}(x_i, x_j)\phi_{ik}(x_i, x_k)$ for all x_i.
  2. Sample x_i proportional to $M_i(x_i)$.

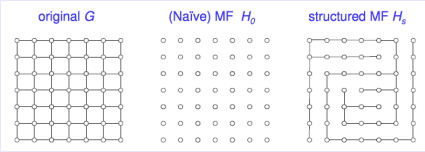
- Mean field for updating a node i with 2 neighbours (j and k).
  1. Compute $M_i(x_i) = \phi_i(x_i) \exp\left(\sum_{x_j} q_j(x_j) \log \phi_{ij}(x_i, x_j) + \sum_{x_k} q_k(x_k) \log \phi_{ik}(x_i, x_k) \right)$.
  2. Set $q_i(x_i)$ proportional to $M_i(x_i)$.

Structure Mean Field

- Common variant is structured mean field: q function includes some of the 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 paper proposed a structured mean field approximation.

Variational vs. Monte Carlo

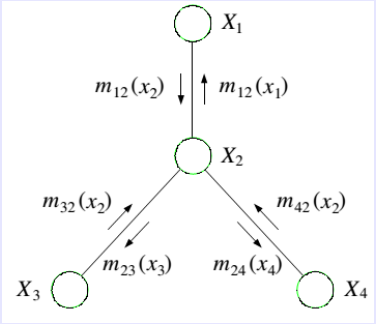
- Monte Carlo vs. variational methods:
  - Variational methods are typically more complicated.
  - Variational methods are not consistent.
    - q does not converge to p if we run the algorithm forever.
  - But variational methods often give better approximation for the same time.
    - Although MCMC is easier to parallelize.
  - Variational methods typically have similar cost to MAP.

- Combinations of variational inference and stochastic methods:
  - Stochastic variational inference (SVI): use SGD to speed up variational methods.
  - Variational MCMC: use Metropolis-Hastings where variational q can make proposals.

Previously: Belief Propagation

- Generalization of forward-backward to forests is belief propagation.

(undirected graphs with no loops, which must be pairwise)



https://www.quora.com/Probabilistic-graphical-models-what-are-the-relationships-between-sum-product-algorithm-belief-propagation-and-junction-tree-

- Defines "messages" that can be sent along each edge.

Loopy Belief Propagation

- In pairwise UGM, belief propagation "message" from parent p to child c is gven by

$$M_{pc}(x_c) \propto \sum_{x_p} \phi_i(x_p)\phi_{pc}(x_p, x_c)M_{jp}(x_p)M_{kp}(x_p),$$

  assuming that parent p has parents j and k.
  - We get marginals by multiplying all incoming messages with local potentials.

- Loopy belief propagation: a "hacker" approach to approximate marginals:
  - Choose an edge ic to update.
  - Update messages $M_{ic}(x_c)$ keeping all other messages fixed.
  - Repeat until "convergence".
    - We approximate marginals by multiplying all incoming messages with local potentials.

- Empirically much better than mean field, we've spent 20+ years figuring out why.

Discussion of Loopy Belief Propagation

- Loopy BP decoding is used for "error correction" in WiFi and Skype.
  - Called "turbo codes" in information theory.
- Loopy BP is not optimizing an objective function.
  - Convergence of loopy BP is hard to characterize: does not converge in general.
- If it converges, loopy BP finds fixed point of "Bethe free energy":
  - Instead of "Gibbs mean-field free-energy" for mean field, which lower bounds Z.
  - Bethe typically gives better approximation than mean field, but not a bound.
- There are convex variants that upper bound Z.
  - Tree-reweighted belief propagation.
  - Variations that are guaranteed to converge.
    - Convex variants are more consistent but often give worse approximations.
- Messages only have closed-form update for conjugate models.
  - Can approximate non-conjugate models using expectation propagation.

Convex Relaxations

- I've overviewed view of variational methods as minimzing non-convex reverse KL.
- Alternate view: write exact inference as constrained convex optimization.
  - Writing inference as maximizing entropy with constraints on marginals.
    - See bonus slides exponential family lecture.
  - Different methods correspond to different entropy/constraint approximations.
    - Mean field and loopy belief propation relax entropy and marginals in different ways.
    - Weirdly, these approximations are non-convex even though original problem is convex.
  - There are also convex relaxations that approximate with linear programs (or SDPs).
- For an overview of these ideas, see:
  `people.eecs.berkeley.edu/~wainwrig/Papers/WaiJor08_FTML.pdf`

Summary

- Topic models: latent-factor model of discrete data text.
  - The latent "factors" are called "topics".
- Latent Dirichlet allocation: hierarchical Bayesian topic model.
  - Represent words in documents as coming from different topics.
  - Each document has its own proportion for each topic.
- Variational methods approximate p with a simpler distribution q.
  - Mean field approximation minimizes reverse KL divergence with independent q.
  - Loopy belief propagation is a heuristic that often works well.
- Next lecture: VAEs and GANs.

Outline

Other Topics

- The VAE/GAN slides are here (lecture will be posted to Piazza):
  `https://www.cs.ubc.ca/~schmidtm/Courses/540-W20/L36.pdf`

- Some other topics we did not have time for:
  - Graph neural nets.
  - Deep sets.
  - Normalizing flows.
  - Particle filters.
  - PixelCNN.
  - Vision transfomers.

- And reinforcement learning (really good when you have a simulator).
  - Read Sutton ad Barto's "Introduction to Reinforcement Learning".
  - You can also take EECE 592 or Michiel van de Panne's graduate course.
    - Or maybe convince Jeff Clune to teach this? Or a new hire?

Other Topics

- Major topics we did not cover in 340 or 440:
  - Optimization methods (does SGD converge on neural networks with ReLU?).
    - Will give a grad course next year, or lecture series over the summer.
  - Online learning (data coming in over time).
  - Active learning (semi-supervised where you choose examples to label).
  - Causality (distinguishing cause from effect.).
  - Learning theory (VC dimension).
  - Probabilistic context-free grammars (recursive version of Markov chains).
  - Probabilistic programming ("object oriented" graphical models).
  - Sub-modularity (discrete version of convexity).
  - Spectral methods (consistent HMM parameter estimation).

- Long-term, we will probably split into multiple courses.

A Word of Caution

- ML world is really exciting right now, but proceed with caution:
  - Need rigorous testing, sanity checking, and considering misuse cases.
  - "Microsoft deletes 'teen girl' AI after it became a Hitler-loving sex robot within 24 hours":
    - `https://www.telegraph.co.uk/technology/2016/03/24/microsofts-teen-girl-ai-turns-into-a-hitler-loving-sex-robot-wit`
  - "Amazon AI Designed to Choose Phone Cases Terribly Malfunctions, Fills Store with 31,000+ Hilarious Products:
    - `https://www.boredpanda.com/funny-amazon-ai-designed-phone-cases-fail`
  - "Uber video shows the kind of crash self-driving cars are made to avoid":
    - `https://www.wired.com/story/uber-self-driving-crash-video-arizona`
  - "Failures of Gradient-Based Deep Learning":
    - `https://arxiv.org/abs/1703.07950`
  - Important to get a sense of what can and cannot be done (now and in near-future).
    - Many industry people have unrealistic expectations.

What is Next?

- "Calling Bullshit in the Age of Big Data":
  - https://www.youtube.com/playlist?list=PLPnZfvKID1Sje5jWxt-4CSZD7bUI4gSPS
  - There is a lot of bullshit in the machine learning world.
    - For example, cherry-picking of examples in papers and overfitting to test sets.
  - You should try to start recognizing obvious non-sense, and not accidently produce non-sense yourself!

- Material from all my courses is here:
  - https://www.cs.ubc.ca/~schmidtm/Courses/LecturesOnML
  - "100 Lectures on Machine Learning".
  - I will try to keep this up to date and keep extending it with new topics.
- Our (mostly-weekly) Machine Learning Reading Group (MLRG):
  - http://www.cs.ubc.ca/labs/lci/mlrg

- Thank you for your patience this term!
  - Combination of hybrid-online/newCourse/newOrganization/newSlides is not easy.
  - Good luck with the next steps!

Difficulty of Variational Formulation

- In exponential family bonus slides, we write inference as a convex optimization:

$$\log(Z) = \sup_{\mu \in \mathcal{M}} \{w^T \mu + H(p_\mu)\},$$

- Did this make anything easier?
  - Computing entropy $H(p_\mu)$ seems as hard as inference.
  - Characterizing marginal polytope $\mathcal{M}$ becomes hard with loops.
- Practical variational methods:
  - Work with approximation/bound on entropy H.
  - Work with approximation to marginal polytope $\mathcal{M}$.

Mean Field Approximation

- Mean field approximation assumes

$$\mu_{ij,st} = \mu_{i,s}\mu_{j,t},$$

for all edges, which means

$$p(x_i = s, x_j = t) = p(x_i = s)p(x_j = t),$$

and that variables are independent.

- Entropy is simple under mean field approximation:

$$\sum_X p(X) \log p(X) = \sum_i \sum_{x_i} p(x_i) \log p(x_i).$$

- Marginal polytope is also simple:

$$\mathcal{M}_F = \{\mu \mid \mu_{i,s} \geq 0,\ \sum_s \mu_{i,s} = 1,\ \mu_{ij,st} = \mu_{i,s}\mu_{j,t}\}.$$

Entropy of Mean Field Approximation

- Entropy form is from distributive law and probabilities sum to 1:

$$\begin{aligned}
\sum_X p(X) \log p(X) &= \sum_X p(X) \log(\prod_i p(x_i)) \\
&= \sum_X p(X) \sum_i \log(p(x_i)) \\
&= \sum_i \sum_X p(X) \log p(x_i) \\
&= \sum_i \sum_X \prod_j p(x_j) \log p(x_i) \\
&= \sum_i \sum_X p(x_i) \log p(x_i) \prod_{j \neq i} p(x_j) \\
&= \sum_i \sum_{x_i} p(x_i) \log p(x_i) \sum_{x_j \mid j \neq i} \prod_{j \neq i} p(x_j) \\
&= \sum_i \sum_{x_i} p(x_i) \log p(x_i).
\end{aligned}$$

Mean Field as Non-Convex Lower Bound

- Since $\mathcal{M}_F \subseteq \mathcal{M}$, yields a lower bound on $\log(Z)$:

$$\sup_{\mu \in \mathcal{M}_F} \{w^T \mu + H(p_\mu)\} \leq \sup_{\mu \in \mathcal{M}} \{w^T \mu + H(p_\mu)\} = \log(Z).$$

- Since $\mathcal{M}_F \subseteq \mathcal{M}$, it is an inner approximation:

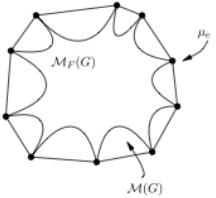
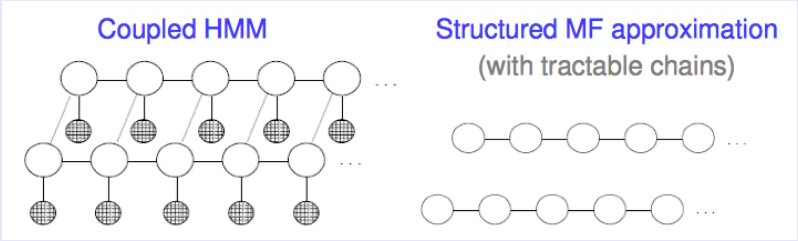


Fig. 5.3 Cartoon illustration of the set $\mathcal{M}_F(G)$ of mean parameters that arise from tractable distributions is a nonconvex inner bound on $\mathcal{M}(G)$. Illustrated here is the case of discrete random variables where $\mathcal{M}(G)$ is a polytope. The circles correspond to mean parameters that arise from delta distributions, and belong to both $\mathcal{M}(G)$ and $\mathcal{M}_F(G)$.

- Constraints $\mu_{ij,st} = \mu_{i,s}\mu_{j,t}$ make it non-convex.
- Mean field algorithm is coordinate descent on $w^T \mu + H(p_\mu)$ over $\mathcal{M}_F$.

Discussion of Mean Field and Structured MF

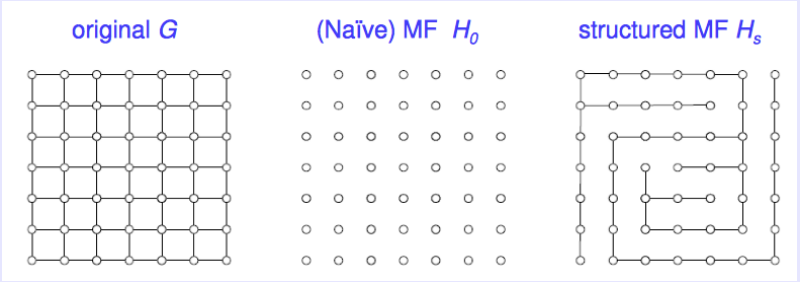
- Mean field is weird:
  - Non-convex approximation to a convex problem.
  - For learning, we want upper bounds on $\log(Z)$.
- Structured mean field:
  - Cost of computing entropy is similar to cost of inference.
  - Use a subgraph where we can perform exact inference.



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

Structured Mean Field with Tree

- More edges means better approximation of $\mathcal{M}$ and $H(p_\mu)$:



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

- Fixed points of loopy correspond to using "Bethe" approximation of entropy and "local polytope" approximation of "marginal polytope".

- You can design better variational methods by constructing better approximations.