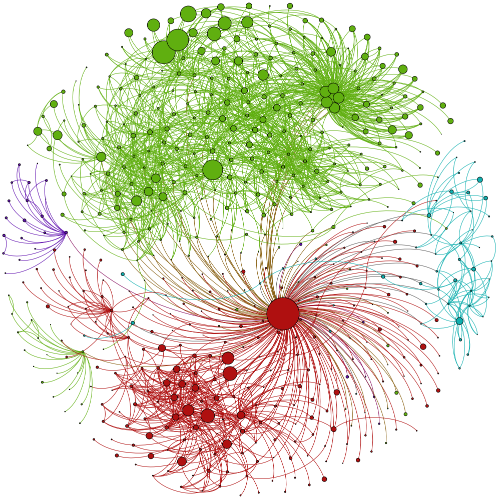
Variational Inference: Foundations and Modern Methods

David Blei, Rajesh Ranganath, Shakir Mohamed

NIPS 2016 Tutorial · December 5, 2016

Columbia University in the City of New York
Princeton University
DeepMind



Communities discovered in a 3.7M node network of U.S. Patents

[Gopalan and Blei, PNAS 2013]

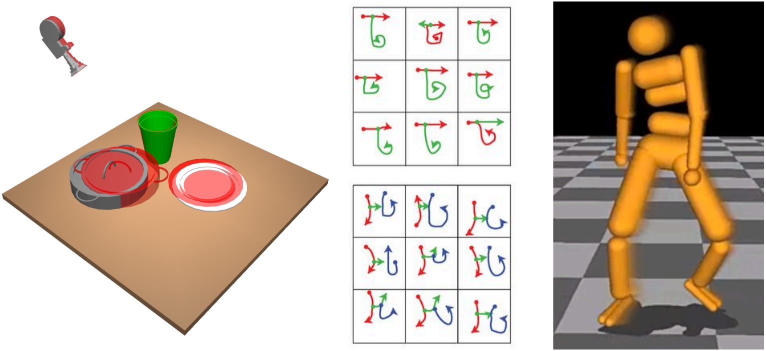
1	2	3	4	5
Game	Life	Film	Book	Wine
Season	Know	Movie	Life	Street
Team	School	Show	Books	Hotel
Coach	Street	Life	Novel	House
Play	Man	Television	Story	Room
Points	Family	Films	Man	Night
Games	Says	Director	Author	Place
Giants	House	Man	House	Restaurant
Second	Children	Story	War	Park
Players	Night	Says	Children	Garden

6	7	8	9	10
Bush	Building	Won	Yankees	Government
Campaign	Street	Team	Game	War
Clinton	Square	Second	Mets	Military
Republican	Housing	Race	Season	Officials
House	House	Round	Run	Iraq
Party	Buildings	Cup	League	Forces
Democratic	Development	Open	Baseball	Iraqi
Political	Space	Game	Team	Army
Democrats	Percent	Play	Games	Troops
Senator	Real	Win	Hit	Soldiers

11	12	13	14	15
Children	Stock	Church	Art	Police
School	Percent	War	Museum	Yesterday
Women	Companies	Women	Show	Man
Family	Fund	Life	Gallery	Officer
Parents	Market	Black	Works	Officers
Child	Bank	Political	Artists	Case
Life	Investors	Catholic	Street	Found
Says	Funds	Government	Artist	Charged
Help	Financial	Jewish	Paintings	Street
Mother	Business	Pope	Exhibition	Shot

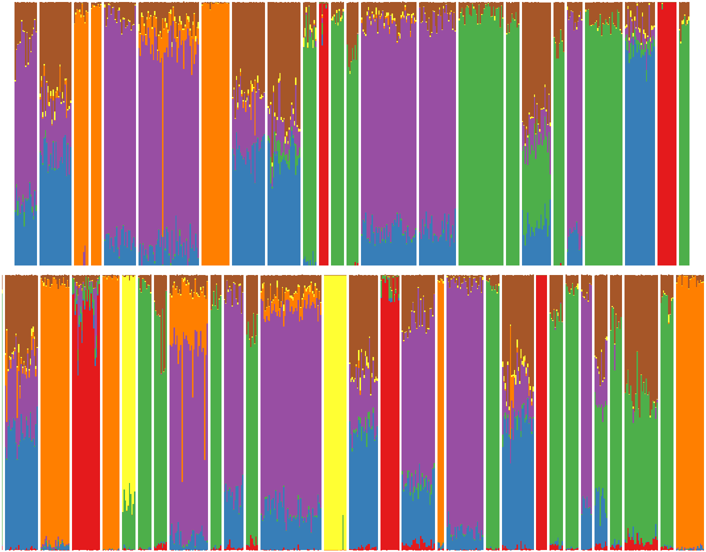
Topics found in 1.8M articles from the New York Times

[Hoffman, Blei, Wang, Paisley, JMLR 2013]



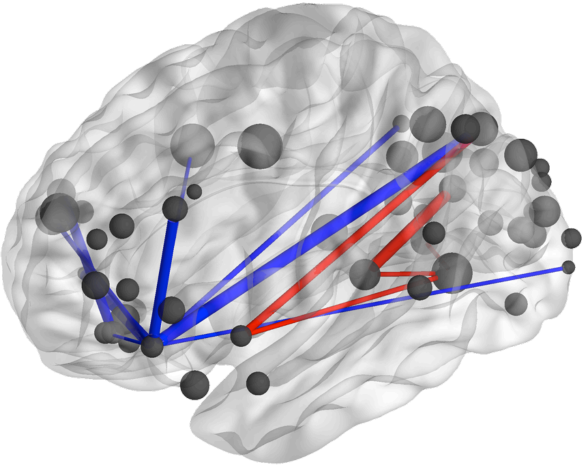
Scenes, concepts and control.

[Eslami et al., 2016, Lake et al. 2015]



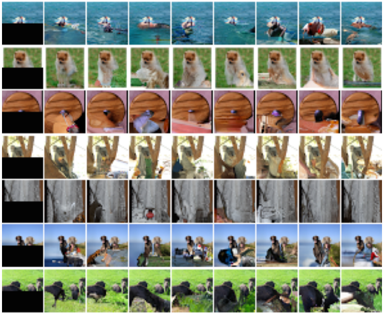
Population analysis of 2 billion genetic measurements

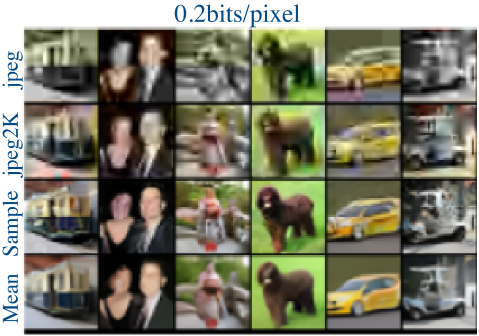
[Gopalan, Hao, Blei, Storey, Nature Genetics (in press)]



Neuroscience analysis of 220 million fMRI measurements

[Manning et al., PLOS ONE 2014]





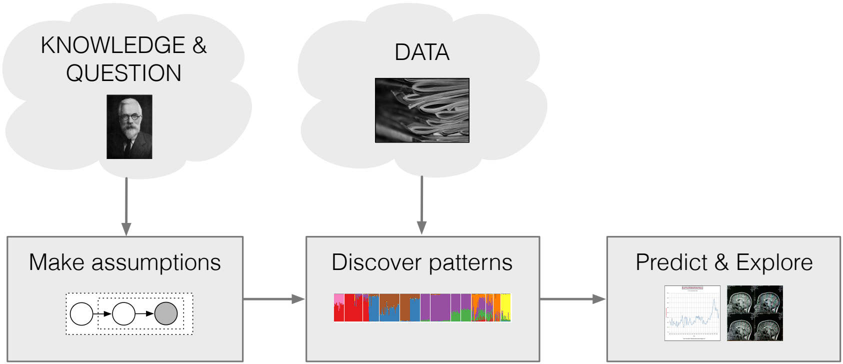
Compression and content generation.

[Van den Oord et al., 2016, Gregor et al., 2016]

Analysis of 1.7M taxi trajectories, in Stan

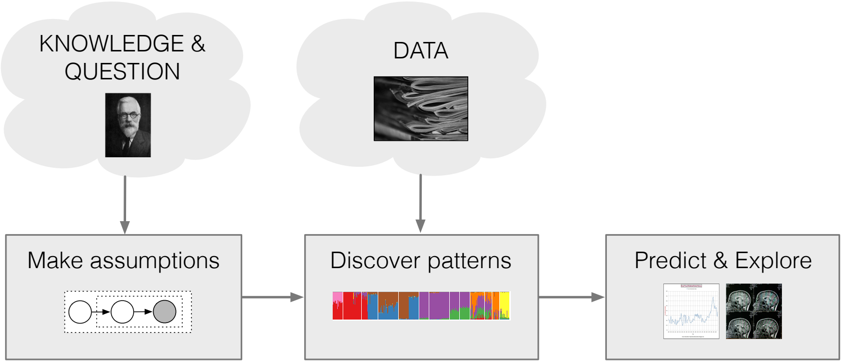
[Kucukelbir et al., 2016]

The probabilistic pipeline

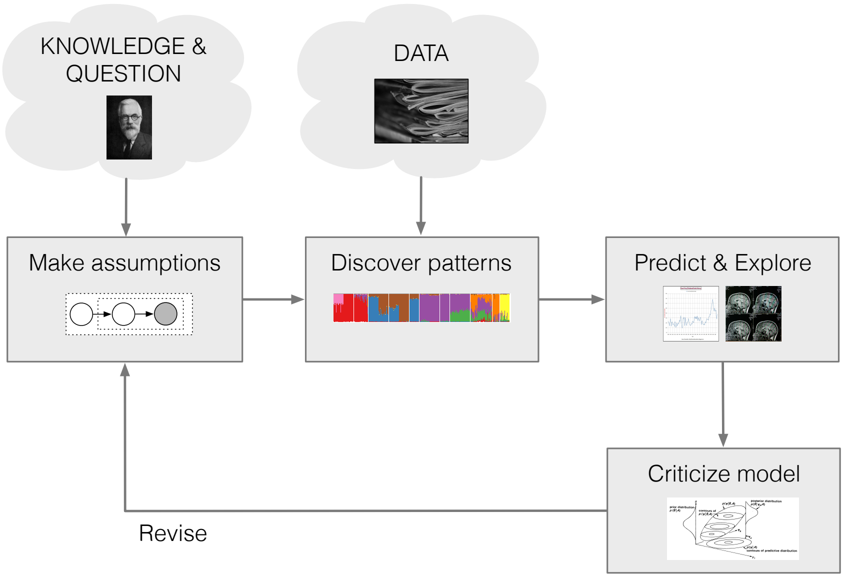


- Customized data analysis is important to many fields.
- Pipeline separates **assumptions**, **computation**, **application**
- Eases collaborative solutions to statistics problems

The probabilistic pipeline



- **Inference** is the key algorithmic problem.
- Answers the question: What does this model say about this data?
- Our goal: **General** and **scalable** approaches to inference



[Box, 1980; Rubin, 1984; Gelman et al., 1996; Blei, 2014]

PART I

Main ideas and historical context

Probabilistic Machine Learning

- A probabilistic model is a joint distribution of hidden variables $\mathbf{z}$ and observed variables $\mathbf{x}$,

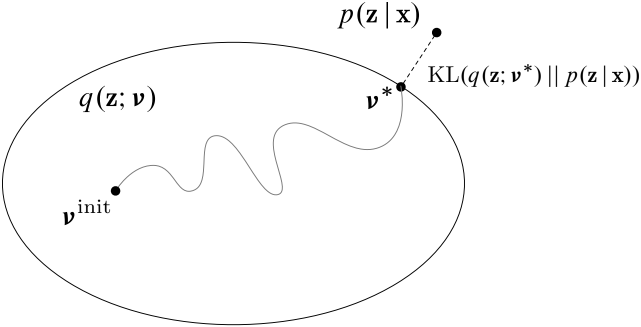
$$p(\mathbf{z}, \mathbf{x}).$$

- Inference about the unknowns is through the **posterior**, the conditional distribution of the hidden variables given the observations

$$p(\mathbf{z} \mid \mathbf{x}) = \frac{p(\mathbf{z}, \mathbf{x})}{p(\mathbf{x})}.$$

- For most interesting models, the denominator is not tractable. We appeal to **approximate posterior inference**.

Variational Inference

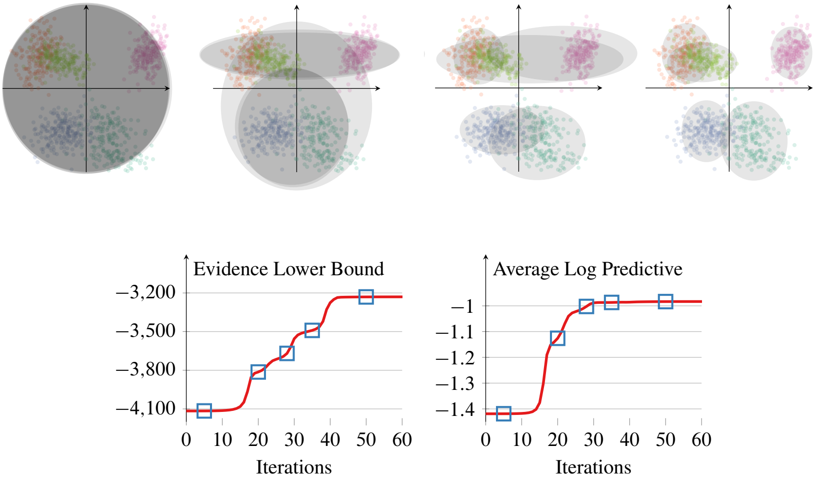


- VI turns **inference into optimization**.
- Posit a **variational family** of distributions over the latent variables,

$$q(\mathbf{z}; \boldsymbol{\nu})$$

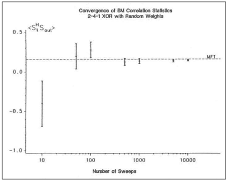
- Fit the **variational parameters** $\boldsymbol{\nu}$ to be close (in KL) to the exact posterior.
  (There are alternative divergences, which connect to algorithms like EP, BP, and others.)

Example: Mixture of Gaussians

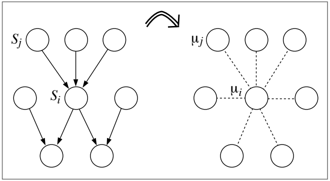


[images by Alp Kucukelbir]

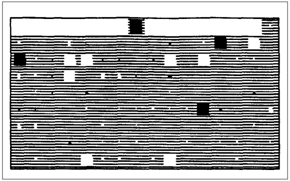
History



[Peterson and Anderson 1987]



[Jordan et al. 1999]

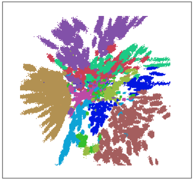


[Hinton and van Camp 1993]

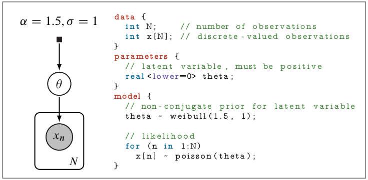
- Variational inference adapts **ideas from statistical physics** to probabilistic inference. Arguably, it began in the late eighties with Peterson and Anderson (1987), who used mean-field methods to fit a neural network.
- This idea was picked up by Jordan's lab in the early 1990s—Tommi Jaakkola, Lawrence Saul, Zoubin Gharamani—who **generalized it to many probabilistic models**. (A review paper is Jordan et al., 1999.)
- In parallel, Hinton and Van Camp (1993) also **developed mean-field for neural networks**. Neal and Hinton (1993) connected this idea to the EM algorithm, which lead to further variational methods for mixtures of experts (Waterhouse et al., 1996) and HMMs (MacKay, 1997).

Today

[Kingma and Welling 2013]



[Rezende et al. 2014]



[Kucukelbir et al. 2015]

- There is now a flurry of new work on variational inference, making it scalable, easier to derive, faster, more accurate, and applying it to more complicated models and applications.
- Modern VI touches many important areas: probabilistic programming, reinforcement learning, neural networks, convex optimization, Bayesian statistics, and myriad applications.
- Our goal today is to teach you the basics, explain some of the newer ideas, and to suggest open areas of new research.

Variational Inference:
Foundations and Modern Methods

Part II: Mean-field VI and stochastic VI

Jordan+, *Introduction to Variational Methods for Graphical Models*, 1999

Ghahramani and Beal, *Propagation Algorithms for Variational Bayesian Learning*, 2001

Hoffman+, *Stochastic Variational Inference*, 2013

Part III: Stochastic gradients of the ELBO

Kingma and Welling, *Auto-Encoding Variational Bayes*, 2014

Ranganath+, *Black Box Variational Inference*, 2014

Rezende+, *Stochastic Backpropagation and Approximate Inference in Deep Generative Models*, 2014

Part IV: Beyond the mean field

Agakov and Barber, *An Auxiliary Variational Method*, 2004

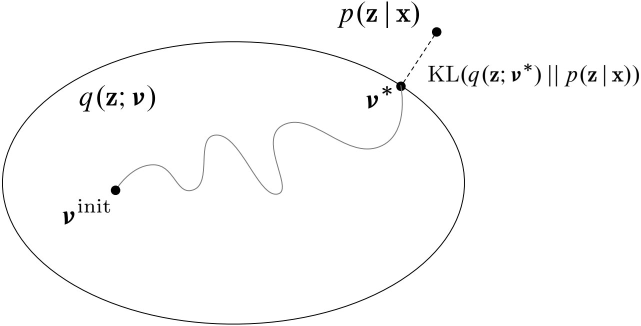
Gregor+, *DRAW: A recurrent neural network for image generation*, 2015

Rezende+, *Variational Inference with Normalizing Flows*, 2015

Ranganath+, *Hierarchical Variational Models*, 2015

Maaløe+, *Auxiliary Deep Generative Models*, 2016

Variational Inference: Foundations and Modern Methods



VI approximates difficult quantities from complex models.

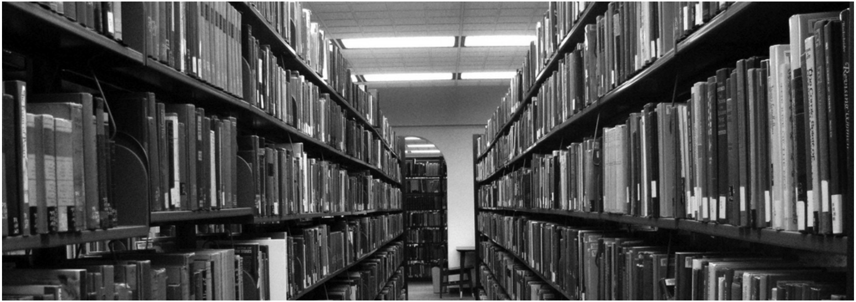
With **stochastic optimization** we can

- scale up VI to massive data
- enable VI on a wide class of difficult models
- enable VI with elaborate and flexible families of approximations

PART II

Mean-field variational inference and stochastic variational inference

Motivation: Topic Modeling



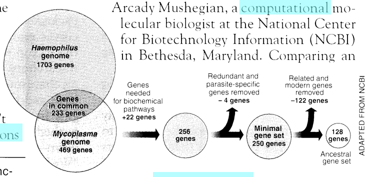
Topic models use posterior inference to discover the hidden thematic structure in a large collection of documents.

Example: Latent Dirichlet Allocation (LDA)

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 250 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 predictions "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 answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped 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 an

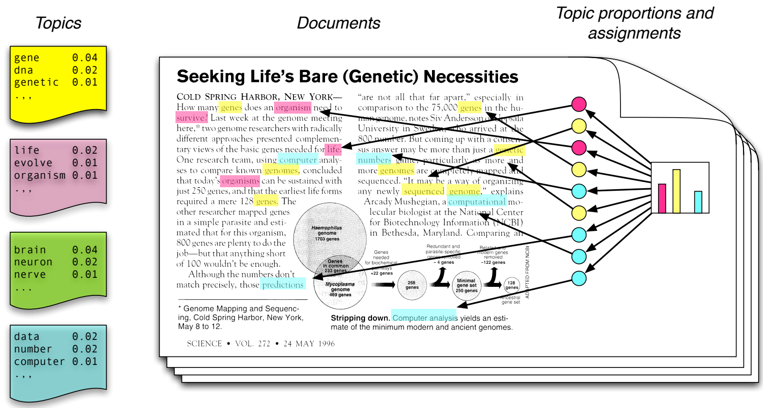


Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

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

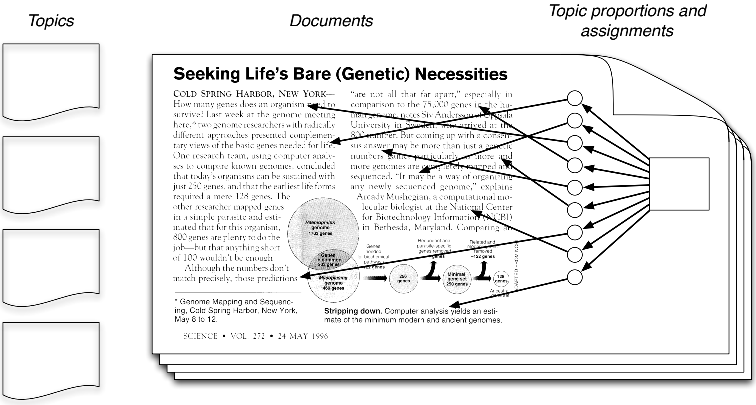
Documents exhibit multiple topics.

Example: Latent Dirichlet Allocation (LDA)



- Each **topic** is a distribution over words
- Each **document** is a mixture of corpus-wide topics
- Each **word** is drawn from one of those topics

Example: Latent Dirichlet Allocation (LDA)

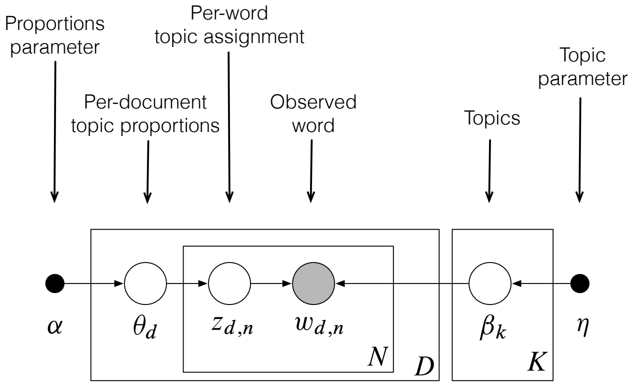


- But we only observe the documents; everything else is hidden.
- So we want to calculate the posterior

$$p(\text{topics, proportions, assignments} \mid \text{documents})$$

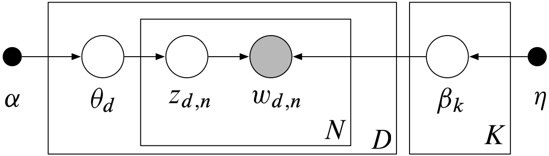
(Note: millions of documents; billions of latent variables)

LDA as a Graphical Model



- Encodes **assumptions** about data with a factorization of the joint
- Connects assumptions to **algorithms** for computing with data
- Defines the **posterior** (through the joint)

Posterior Inference



- The posterior of the latent variables given the documents is

$$p(\beta, \boldsymbol{\theta}, \mathbf{z} \,|\, \mathbf{w}) = \frac{p(\beta, \boldsymbol{\theta}, \mathbf{z}, \mathbf{w})}{\int_{\beta} \int_{\boldsymbol{\theta}} \sum_{\mathbf{z}} p(\beta, \boldsymbol{\theta}, \mathbf{z}, \mathbf{w})}.$$

- We can't compute the denominator, the marginal $p(\mathbf{w})$.
- We use approximate inference.

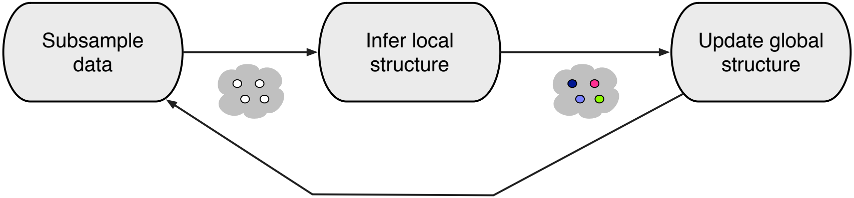
1	2	3	4	5
Game	Life	Film	Book	Wine
Season	Know	Movie	Life	Street
Team	School	Show	Books	Hotel
Coach	Street	Life	Novel	House
Play	Man	Television	Story	Room
Points	Family	Films	Man	Night
Games	Says	Director	Author	Place
Giants	House	Man	House	Restaurant
Second	Children	Story	War	Park
Players	Night	Says	Children	Garden

6	7	8	9	10
Bush	Building	Won	Yankees	Government
Campaign	Street	Team	Game	War
Clinton	Square	Second	Mets	Military
Republican	Housing	Race	Season	Officials
House	House	Round	Run	Iraq
Party	Buildings	Cup	League	Forces
Democratic	Development	Open	Baseball	Iraqi
Political	Space	Game	Team	Army
Democrats	Percent	Play	Games	Troops
Senator	Real	Win	Hit	Soldiers

11	12	13	14	15
Children	Stock	Church	Art	Police
School	Percent	War	Museum	Yesterday
Women	Companies	Women	Show	Man
Family	Fund	Life	Gallery	Officer
Parents	Market	Black	Works	Officers
Child	Bank	Political	Artists	Case
Life	Investors	Catholic	Street	Found
Says	Funds	Government	Artist	Charged
Help	Financial	Jewish	Paintings	Street
Mother	Business	Pope	Exhibition	Shot

Topics found in 1.8M articles from the New York Times

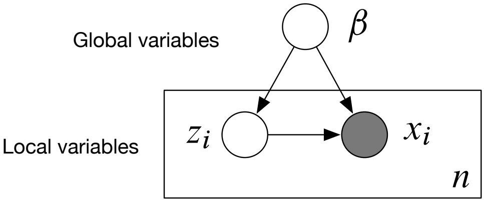
Mean-field VI and Stochastic VI



Road map:

- Define the generic class of conditionally conjugate models
- Derive classical mean-field VI
- Derive stochastic VI, which scales to massive data

A Generic Class of Models

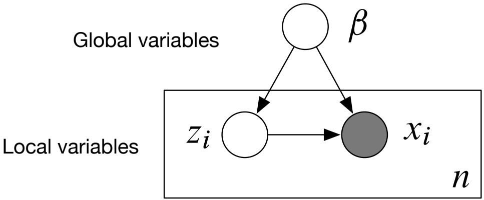


$$p(\beta, \mathbf{z}, \mathbf{x}) = p(\beta) \prod_{i=1}^{n} p(z_i, x_i \mid \beta)$$

- The observations are $\mathbf{x} = x_{1:n}$.
- The **local** variables are $\mathbf{z} = z_{1:n}$.
- The **global** variables are β.
- The ith data point x_i only depends on z_i and β.

Compute $p(\beta, \mathbf{z} \mid \mathbf{x})$.

A Generic Class of Models



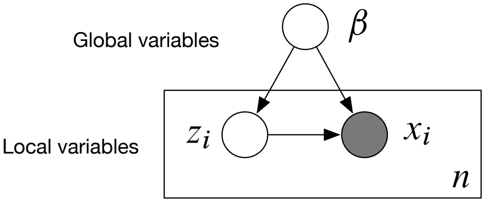
$$p(\beta, \mathbf{z}, \mathbf{x}) = p(\beta) \prod_{i=1}^{n} p(z_i, x_i \mid \beta)$$

- A **complete conditional** is the conditional of a latent variable given the observations and other latent variables.

- Assume each complete conditional is in the exponential family,

$$p(z_i \mid \beta, x_i) = h(z_i) \exp\{\eta_\ell(\beta, x_i)^\top z_i - a(\eta_\ell(\beta, x_i))\}$$
$$p(\beta \mid \mathbf{z}, \mathbf{x}) = h(\beta) \exp\{\eta_g(\mathbf{z}, \mathbf{x})^\top \beta - a(\eta_g(\mathbf{z}, \mathbf{x}))\}.$$

A Generic Class of Models



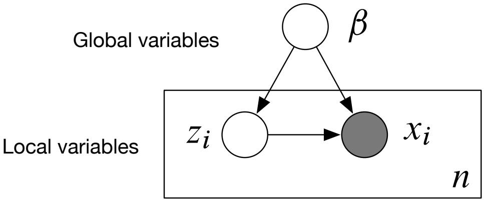
$$p(\beta, \mathbf{z}, \mathbf{x}) = p(\beta) \prod_{i=1}^{n} p(z_i, x_i \mid \beta)$$

- A **complete conditional** is the conditional of a latent variable given the observations and other latent variable.
- The global parameter comes from conjugacy [Bernardo and Smith, 1994]

$$\eta_g(\mathbf{z}, \mathbf{x}) = \alpha + \textstyle\sum_{i=1}^{n} t(z_i, x_i),$$

where α is a hyperparameter and $t(\cdot)$ are sufficient statistics for $[z_i, x_i]$.

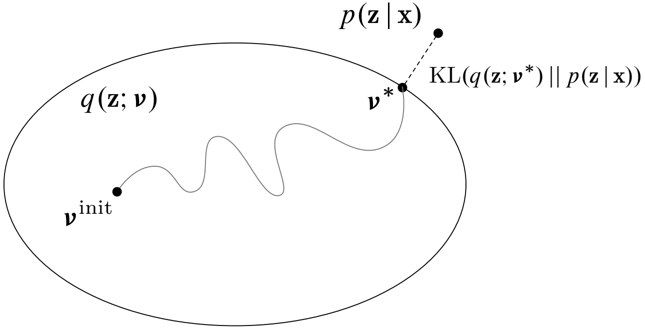
A Generic Class of Models



$$p(\beta, \mathbf{z}, \mathbf{x}) = p(\beta) \prod_{i=1}^{n} p(z_i, x_i \mid \beta)$$

- Bayesian mixture models
- Time series models (HMMs, linear dynamic systems)
- Factorial models
- Matrix factorization (factor analysis, PCA, CCA)
- Dirichlet process mixtures, HDPs
- Multilevel regression (linear, probit, Poisson)
- Stochastic block models
- Mixed-membership models (LDA and some variants)

Variational Inference



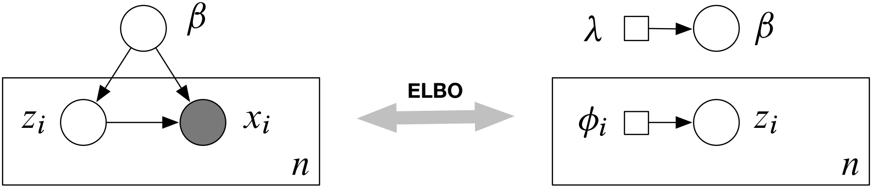
Minimize KL between $q(\beta, \mathbf{z}; \boldsymbol{\nu})$ and the posterior $p(\beta, \mathbf{z} \mid \mathbf{x})$.

The Evidence Lower Bound

$$\mathscr{L}(\boldsymbol{\nu}) = \mathbb{E}_q[\log p(\beta, \mathbf{z}, \mathbf{x})] - \mathbb{E}_q[\log q(\beta, \mathbf{z}; \boldsymbol{\nu})]$$

- KL is intractable; VI optimizes the **evidence lower bound** (ELBO) instead.
  - It is a lower bound on $\log p(\mathbf{x})$.
  - Maximizing the ELBO is equivalent to minimizing the KL.
- The ELBO trades off two terms.
  - The first term prefers $q(\cdot)$ to place its mass on the MAP estimate.
  - The second term encourages $q(\cdot)$ to be diffuse.
- Caveat: The ELBO is not convex.

Mean-field Variational Inference



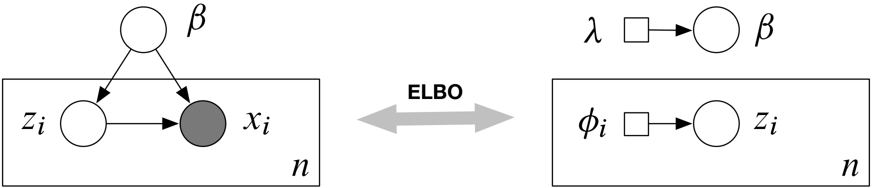
- We need to specify the form of $q(\beta, \mathbf{z})$.
- The **mean-field family** is fully factorized,

$$q(\beta, \mathbf{z}; \lambda, \boldsymbol{\phi}) = q(\beta; \lambda) \prod_{i=1}^{n} q(z_i; \phi_i).$$

- Each factor is the same family as the model's complete conditional,

$$p(\beta \,|\, \mathbf{z}, \mathbf{x}) = h(\beta) \exp\{\eta_g(\mathbf{z}, \mathbf{x})^\top \beta - a(\eta_g(\mathbf{z}, \mathbf{x}))\}$$
$$q(\beta; \lambda) = h(\beta) \exp\{\lambda^\top \beta - a(\lambda)\}.$$

Mean-field Variational Inference



- Optimize the ELBO,

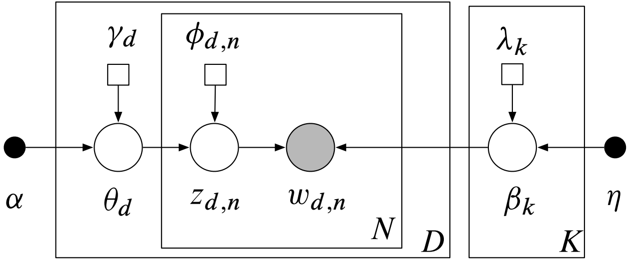
$$\mathscr{L}(\lambda, \boldsymbol{\phi}) = \mathbb{E}_q[\log p(\beta, \mathbf{z}, \mathbf{x})] - \mathbb{E}_q[\log q(\beta, \mathbf{z})].$$

- Traditional VI uses coordinate ascent [Ghahramani and Beal, 2001]

$$\lambda^* = \mathbb{E}_\phi\left[\eta_g(\mathbf{z}, \mathbf{x})\right];\ \phi_i^* = \mathbb{E}_\lambda[\eta_\ell(\beta, x_i)]$$

- Iteratively update each parameter, holding others fixed.
  - Notice the relationship to Gibbs sampling [Gelfand and Smith, 1990] .
  - Caveat: The ELBO is not convex.

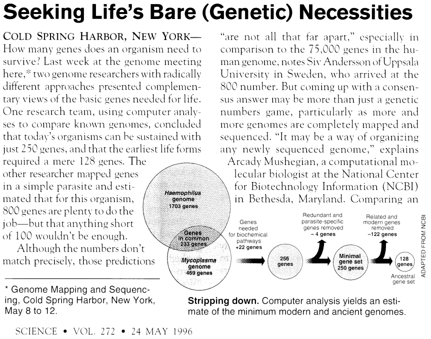
Mean-field Variational Inference for LDA



- The local variables are the per-document variables θ_d and $\mathbf{z}_d$.
- The global variables are the topics $\beta_1, \ldots, \beta_K$.
- The variational distribution is

$$q(\beta, \boldsymbol{\theta}, \mathbf{z}) = \prod_{k=1}^{K} q(\beta_k; \lambda_k) \prod_{d=1}^{D} q(\theta_d; \gamma_d) \prod_{n=1}^{N} q(z_{d,n}; \phi_{d,n})$$

Mean-field Variational Inference for LDA



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 250 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 predictions "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 answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped 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 an

Haemophilus genome 1703 genes

Genes in common 233 genes

Mycoplasma genome 469 genes

Genes needed for biochemical pathways +22 genes

256 genes

Redundant and parasite-specific genes removed − 4 genes

Minimal gene set 250 genes

Related and modern genes removed −122 genes

128 genes

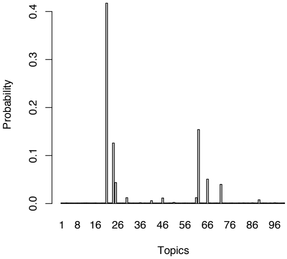
Ancestral gene set

ADAPTED FROM NCBI

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

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

SCIENCE • VOL. 272 • 24 MAY 1996



Mean-field Variational Inference for LDA

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

Classical Variational Inference

Input: data $\mathbf{x}$, model $p(\beta, \mathbf{z}, \mathbf{x})$.

Initialize λ randomly.

repeat

  for *each data point i* **do**

    Set local parameter $\phi_i \leftarrow \mathbb{E}_\lambda [\eta_\ell(\beta, x_i)]$.

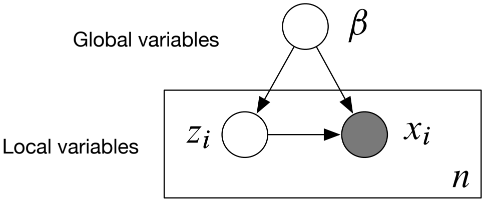
  end

  Set global parameter

$$\lambda \leftarrow \alpha + \textstyle\sum_{i=1}^{n} \mathbb{E}_{\phi_i} [t(Z_i, x_i)] .$$

until *the ELBO has converged*

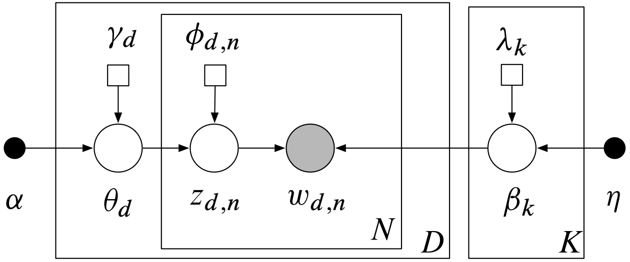
A Generic Class of Models



$$p(\beta, \mathbf{z}, \mathbf{x}) = p(\beta) \prod_{i=1}^{n} p(z_i, x_i \mid \beta)$$

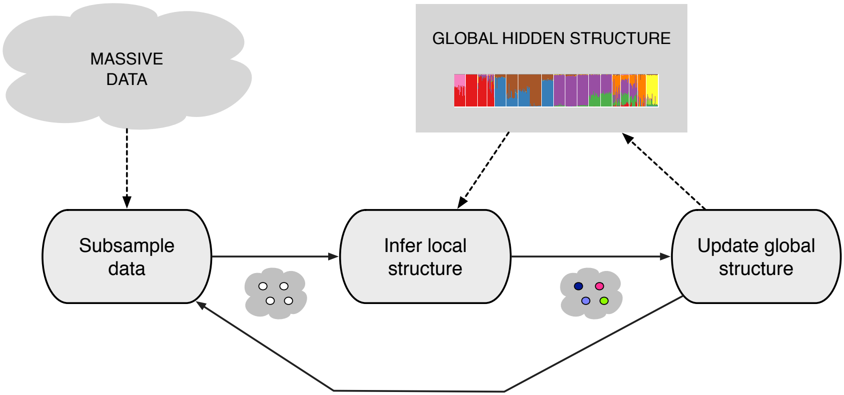
- Bayesian mixture models
- Time series models (HMMs, linear dynamic systems)
- Factorial models
- Matrix factorization (factor analysis, PCA, CCA)
- Dirichlet process mixtures, HDPs
- Multilevel regression (linear, probit, Poisson)
- Stochastic block models
- Mixed-membership models (LDA and some variants)

Stochastic Variational Inference



- Classical VI is inefficient:
  - Do some local computation *for each data point*.
  - Aggregate these computations to re-estimate global structure.
  - Repeat.
- This cannot handle massive data.
- **Stochastic variational inference** (SVI) scales VI to massive data.

Stochastic Variational Inference



Stochastic Optimization

A STOCHASTIC APPROXIMATION METHOD[1]

By Herbert Robbins and Sutton Monro

University of North Carolina

1. Summary. Let $M(x)$ denote the expected value at level x of the response to a certain experiment. $M(x)$ is assumed to be a monotone function of x but is unknown to the experimenter, and it is desired to find the solution $x = \theta$ of the equation $M(x) = \alpha$, where α is a given constant. We give a method for making successive experiments at levels $x_1, x_2, \cdots$ in such a way that x_n will tend to θ in probability.

- Replace the gradient with cheaper noisy estimates [Robbins and Monro, 1951]
- Guaranteed to converge to a local optimum [Bottou, 1996]
- Has enabled modern machine learning

Stochastic Optimization

A STOCHASTIC APPROXIMATION METHOD[1]

BY HERBERT ROBBINS AND SUTTON MONRO

University of North Carolina

1. Summary. Let $M(x)$ denote the expected value at level x of the response to a certain experiment. $M(x)$ is assumed to be a monotone function of x but is unknown to the experimenter, and it is desired to find the solution $x = \theta$ of the equation $M(x) = \alpha$, where α is a given constant. We give a method for making successive experiments at levels $x_1, x_2, \cdots$ in such a way that x_n will tend to θ in probability.

- With noisy gradients, update

$$\nu_{t+1} = \nu_t + \rho_t \hat{\nabla}_\nu \mathscr{L}(\nu_t)$$

- Requires unbiased gradients, $\mathbb{E}\left[\hat{\nabla}_\nu \mathscr{L}(\nu)\right] = \nabla_\nu \mathscr{L}(\nu)$
- Requires the step size sequence ρ_t follows the Robbins-Monro conditions

Stochastic Variational Inference

- The **natural gradient** of the ELBO [Amari, 1998; Sato, 2001]

$$\nabla_{\lambda}^{\text{nat}} \mathscr{L}(\lambda) = \left(\alpha + \sum_{i=1}^{n} \mathbb{E}_{\phi_i^*}[t(Z_i, x_i)]\right) - \lambda.$$

- Construct a **noisy natural gradient**,

$$j \sim \text{Uniform}(1, \ldots, n)$$
$$\hat{\nabla}_{\lambda}^{\text{nat}} \mathscr{L}(\lambda) = \alpha + n\mathbb{E}_{\phi_j^*}[t(Z_j, x_j)] - \lambda.$$

- This is a good noisy gradient.
  - Its expectation is the exact gradient (*unbiased*).
  - It only depends on optimized parameters of one data point (*cheap*).

Stochastic Variational Inference

Input: data $\mathbf{x}$, model $p(\beta, \mathbf{z}, \mathbf{x})$.

Initialize λ randomly. Set ρ_t appropriately.

repeat

Sample $j \sim \text{Unif}(1, \ldots, n)$.

Set local parameter $\phi \leftarrow \mathbb{E}_\lambda\left[\eta_\ell(\beta, x_j)\right]$.

Set intermediate global parameter

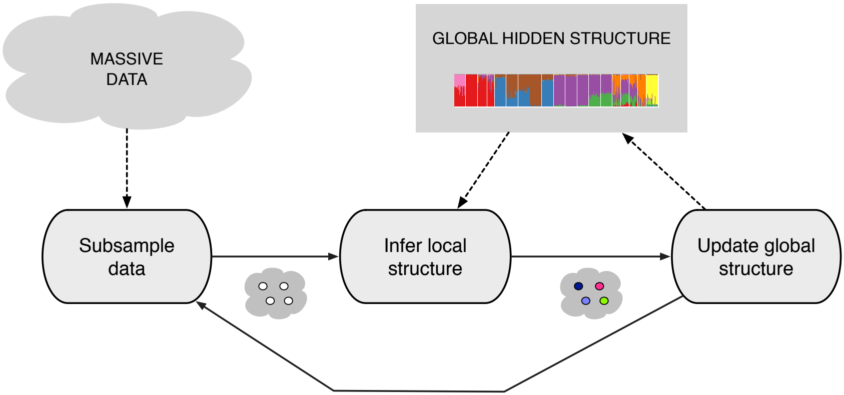
$$\hat{\lambda} = \alpha + n\mathbb{E}_\phi[t(Z_j, x_j)].$$

Set global parameter

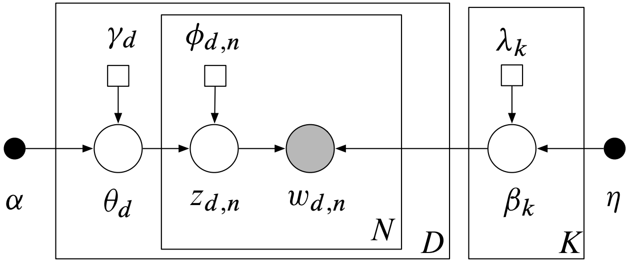
$$\lambda = (1 - \rho_t)\lambda + \rho_t\hat{\lambda}.$$

until *forever*

Stochastic Variational Inference

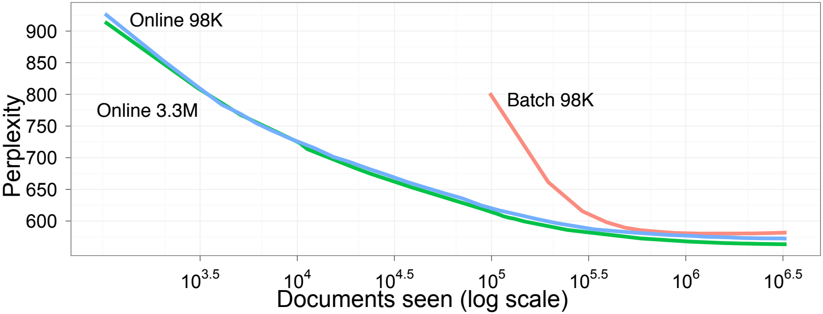


Stochastic Variational Inference in LDA



- Sample a document
- Estimate the local variational parameters using the current topics
- Form intermediate topics from those local parameters
- Update topics as a weighted average of intermediate and current topics

Stochastic Variational Inference in LDA



Documents analyzed	**2048**	**4096**	**8192**	**12288**	**16384**	**32768**	**49152**	**65536**
Top eight words	systems	systems	service	service	service	business	business	business
	road	health	systems	systems	companies	service	service	industry
	made	communication	health	companies	systems	companies	companies	service
	service	service	companies	business	business	industry	industry	companies
	announced	billion	market	company	company	company	services	services
	national	language	communication	billion	industry	management	company	company
	west	care	company	health	market	systems	management	management
	language	road	billion	industry	billion	services	public	public

[Hoffman et al., 2010]

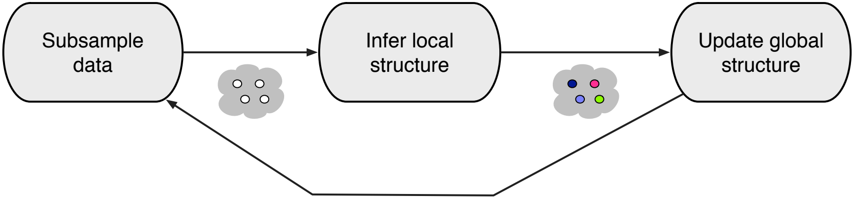
1	2	3	4	5
Game	Life	Film	Book	Wine
Season	Know	Movie	Life	Street
Team	School	Show	Books	Hotel
Coach	Street	Life	Novel	House
Play	Man	Television	Story	Room
Points	Family	Films	Man	Night
Games	Says	Director	Author	Place
Giants	House	Man	House	Restaurant
Second	Children	Story	War	Park
Players	Night	Says	Children	Garden

6	7	8	9	10
Bush	Building	Won	Yankees	Government
Campaign	Street	Team	Game	War
Clinton	Square	Second	Mets	Military
Republican	Housing	Race	Season	Officials
House	House	Round	Run	Iraq
Party	Buildings	Cup	League	Forces
Democratic	Development	Open	Baseball	Iraqi
Political	Space	Game	Team	Army
Democrats	Percent	Play	Games	Troops
Senator	Real	Win	Hit	Soldiers

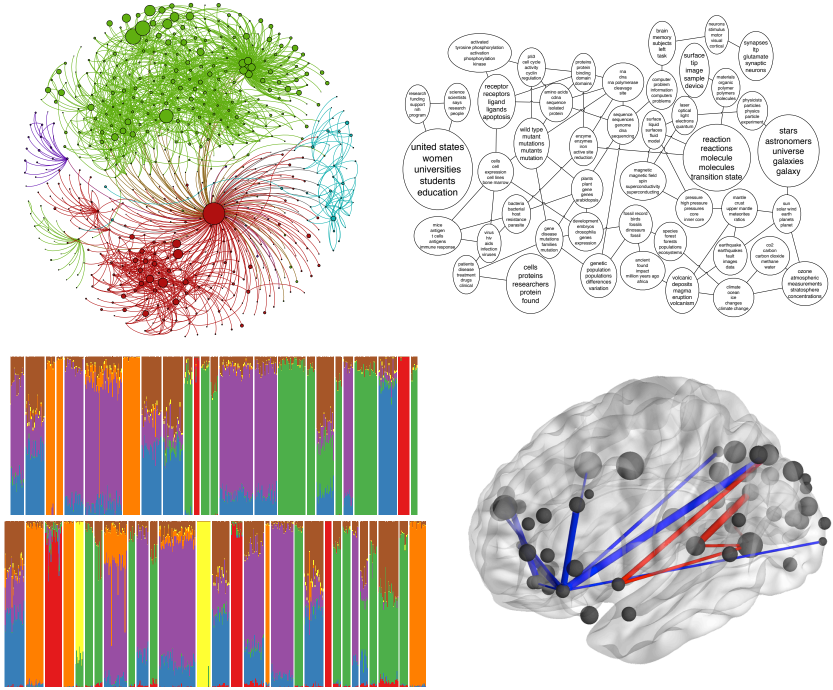
11	12	13	14	15
Children	Stock	Church	Art	Police
School	Percent	War	Museum	Yesterday
Women	Companies	Women	Show	Man
Family	Fund	Life	Gallery	Officer
Parents	Market	Black	Works	Officers
Child	Bank	Political	Artists	Case
Life	Investors	Catholic	Street	Found
Says	Funds	Government	Artist	Charged
Help	Financial	Jewish	Paintings	Street
Mother	Business	Pope	Exhibition	Shot

Topics using the HDP, found in 1.8M articles from the New York Times

SVI scales many models



- Bayesian mixture models
- Time series models
  (HMMs, linear dynamic systems)
- Factorial models
- Matrix factorization
  (factor analysis, PCA, CCA)
- Dirichlet process mixtures, HDPs
- Multilevel regression
  (linear, probit, Poisson)
- Stochastic block models
- Mixed-membership models
  (LDA and some variants)

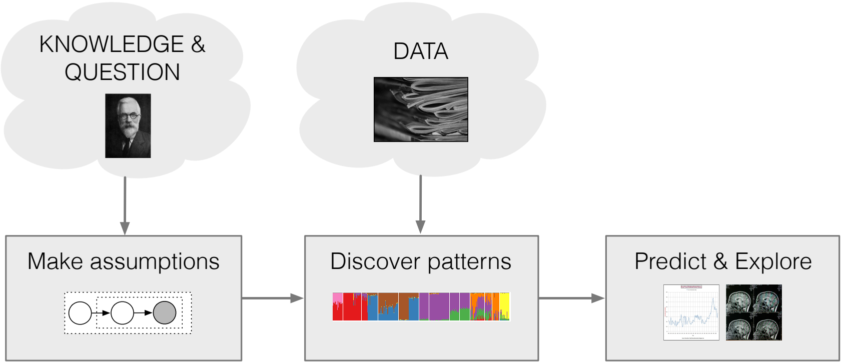


activated tyrosine phosphorylation activation phosphorylation kinase
p53 cell cycle activity cyclin regulation
proteins protein binding domain domains
rna dna rna polymerase cleavage site
brain memory subjects left task
neurons stimulus motor visual cortical
synapses ltp glutamate synaptic neurons
surface tip image sample device
computer problem information computers problems
materials organic polymer polymers molecules
research funding support nih program
science scientists says research people
receptor receptors ligand ligands apoptosis
amino acids cdna sequence isolated protein
sequence sequences genome dna sequencing
surface liquid surfaces fluid model
laser optical light electrons quantum
physicists particles physics particle experiment
united states women universities students education
wild type mutant mutations mutants mutation
enzyme enzymes iron active site reduction
reaction reactions molecule molecules transition state
stars astronomers universe galaxies galaxy
cells cell expression cell lines bone marrow
plants plant gene genes arabidopsis
magnetic magnetic field spin superconductivity superconducting
pressure high pressure pressures core inner core
mantle crust upper mantle meteorites ratios
sun solar wind earth planets planet
bacteria bacterial host resistance parasite
mice antigen t cells antigens immune response
virus hiv aids infection viruses
gene disease mutations families mutation
development embryos drosophila genes expression
fossil record birds fossils dinosaurs fossil
species forest forests populations ecosystems
earthquake earthquakes fault images data
co2 carbon carbon dioxide methane water
patients disease treatment drugs clinical
cells proteins researchers protein found
genetic population populations differences variation
ancient found impact million years ago africa
volcanic deposits magma eruption volcanism
climate ocean ice changes climate change
ozone atmospheric measurements stratosphere concentrations

PART III

Stochastic Gradients of the ELBO

Review: The Promise



- Realized for conditionally conjugate models

- What about the general case?

The Variational Inference Recipe

Start with a model:

$$p(\mathbf{z}, \mathbf{x})$$

The Variational Inference Recipe

Choose a variational approximation:

$$q(\mathbf{z}; \nu)$$

The Variational Inference Recipe

Write down the ELBO:

$$\mathcal{L}(\nu) = \mathbb{E}_{q(\mathbf{z};\,\nu)}[\log p(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z};\, \nu)]$$

The Variational Inference Recipe

Compute the expectation(integral):

$$\text{Example: } \mathscr{L}(\boldsymbol{\nu}) = x\boldsymbol{\nu}^2 + \log \boldsymbol{\nu}$$

The Variational Inference Recipe

Take derivatives:

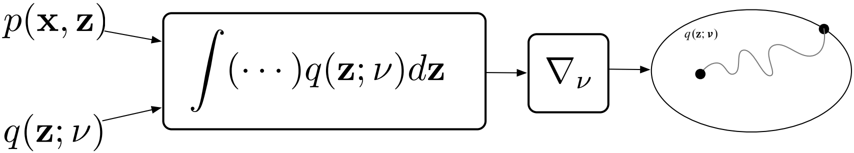
$$\text{Example: } \nabla_{\nu}\mathscr{L}(\nu) = 2x\boldsymbol{\nu} + \frac{1}{\boldsymbol{\nu}}$$

The Variational Inference Recipe

Optimize:

$$\nu_{t+1} = \nu_t + \rho_t \nabla_{\nu} \mathscr{L}$$

The Variational Inference Recipe



Example: Bayesian Logistic Regression

- Data pairs y_i, x_i
- x_i are covariates
- y_i are label
- z is the regression coefficient
- Generative process

$$p(z) \sim N(0, 1)$$
$$p(y_i \mid x_i, z) \sim \text{Bernoulli}(\sigma(zx_i))$$

VI for Bayesian Logistic Regression

Assume:

- We have one data point (y, x)
- x is a scalar
- The approximating family q is the normal; $\boldsymbol{\nu} = (\mu, \sigma^2)$

The ELBO is

$$\mathscr{L}(\mu, \sigma^2) = \mathbb{E}_q[\log p(z) + \log p(y \mid x, z) - \log q(z)]$$

VI for Bayesian Logistic Regression

$$\begin{aligned}
&\mathcal{L}(\mu, \sigma^2) \\
&= \quad \mathbb{E}_q[\log p(z) - \log q(z) + \log p(y \mid x, z)]
\end{aligned}$$

VI for Bayesian Logistic Regression

$$\begin{aligned}
&\mathscr{L}(\mu, \sigma^2) \\
&\quad = \quad \mathbb{E}_q[\log p(z) - \log q(z) + \log p(y \mid x, z)] \\
&\quad = \quad -\frac{1}{2}(\mu^2 + \sigma^2) + \frac{1}{2}\log \sigma^2 + \mathbb{E}_q[\log p(y \mid x, z)] + C
\end{aligned}$$

VI for Bayesian Logistic Regression

$$
\begin{aligned}
&\mathcal{L}(\mu, \sigma^2) \\
&\quad = \quad \mathbb{E}_q[\log p(z) - \log q(z) + \log p(y \mid x, z)] \\
&\quad = \quad -\frac{1}{2}(\mu^2 + \sigma^2) + \frac{1}{2}\log \sigma^2 + \mathbb{E}_q[\log p(y \mid x, z)] + C \\
&\quad = \quad -\frac{1}{2}(\mu^2 + \sigma^2) + \frac{1}{2}\log \sigma^2 + \mathbb{E}_q[yxz - \log(1 + \ exp(xz))]
\end{aligned}
$$

VI for Bayesian Logistic Regression

$$
\begin{aligned}
&\mathscr{L}(\mu, \sigma^2) \\
&\quad = \qquad \mathbb{E}_q[\log p(z) - \log q(z) + \log p(y \,|\, x, z)] \\
&\quad = \qquad -\frac{1}{2}(\mu^2 + \sigma^2) + \frac{1}{2}\log \sigma^2 + \mathbb{E}_q[\log p(y \,|\, x, z)] + C \\
&\quad = \qquad -\frac{1}{2}(\mu^2 + \sigma^2) + \frac{1}{2}\log \sigma^2 + \mathbb{E}_q[yxz - \log(1 + \ exp(xz))] \\
&\quad = \qquad -\frac{1}{2}(\mu^2 + \sigma^2) + \frac{1}{2}\log \sigma^2 + yx\mu - \mathbb{E}_q[\log(1 + \exp(xz))]
\end{aligned}
$$

VI for Bayesian Logistic Regression

$$
\begin{aligned}
&\mathscr{L}(\mu, \sigma^2) \\
&\quad = \quad \mathbb{E}_q[\log p(z) - \log q(z) + \log p(y \,|\, x, z)] \\
&\quad = \quad -\frac{1}{2}(\mu^2 + \sigma^2) + \frac{1}{2}\log \sigma^2 + \mathbb{E}_q[\log p(y \,|\, x, z)] + C \\
&\quad = \quad -\frac{1}{2}(\mu^2 + \sigma^2) + \frac{1}{2}\log \sigma^2 + \mathbb{E}_q[yxz - \log(1 + \ exp(xz))] \\
&\quad = \quad -\frac{1}{2}(\mu^2 + \sigma^2) + \frac{1}{2}\log \sigma^2 + yx\mu - \mathbb{E}_q[\log(1 + \exp(xz))]
\end{aligned}
$$

We are stuck.

1. We cannot analytically take that expectation.
2. The expectation hides the objectives dependence on the variational parameters. This makes it hard to directly optimize.

Options?

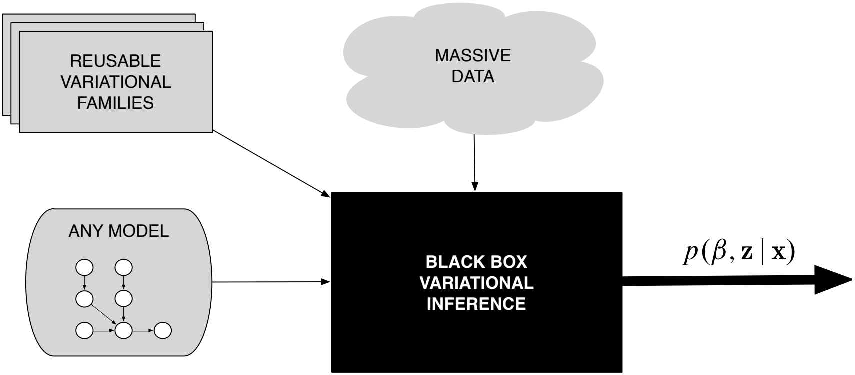
- Derive a model specific bound:
  [Jordan and Jaakola; 1996], [Braun and McAuliffe; 2008], others
- More general approximations that require model-specific analysis:
  [Wang and Blei; 2013], [Knowles and Minka; 2011]

Nonconjugate Models

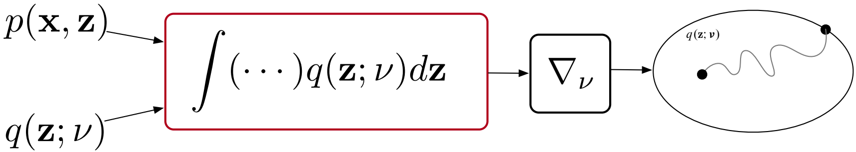
- Nonlinear Time series Models
- Deep Latent Gaussian Models
- Models with Attention (such as DRAW)
- Generalized Linear Models (Poisson Regression)
- Stochastic Volatility Models
- Discrete Choice Models
- Bayesian Neural Networks
- Deep Exponential Families (e.g. Sparse Gamma or Poisson)
- Correlated Topic Model (including nonparametric variants)
- Sigmoid Belief Network

We need a solution that does not entail model specific work

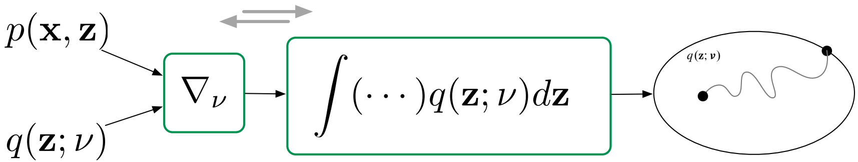
Black Box Variational Inference (BBVI)



The Problem in the Classical VI Recipe



The New VI Recipe



Use stochastic optimization!

Computing Gradients of Expectations

- Define

$$g(\mathbf{z}, \boldsymbol{\nu}) = \log p(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z}; \boldsymbol{\nu})$$

- What is $\nabla_{\boldsymbol{\nu}} \mathscr{L}$

$$\begin{aligned}
\nabla_{\boldsymbol{\nu}} \mathscr{L} &= \nabla_{\boldsymbol{\nu}} \int q(\mathbf{z}; \boldsymbol{\nu}) g(\mathbf{z}, \boldsymbol{\nu}) d\mathbf{z} \\
&= \int \nabla_{\boldsymbol{\nu}} q(\mathbf{z}; \boldsymbol{\nu}) g(\mathbf{z}, \boldsymbol{\nu}) + q(\mathbf{z}; \boldsymbol{\nu}) \nabla_{\boldsymbol{\nu}} g(\mathbf{z}, \boldsymbol{\nu}) d\mathbf{z} \\
&= \int q(\mathbf{z}; \boldsymbol{\nu}) \nabla_{\boldsymbol{\nu}} \log q(\mathbf{z}; \boldsymbol{\nu}) g(\mathbf{z}, \boldsymbol{\nu}) + q(\mathbf{z}; \boldsymbol{\nu}) \nabla_{\boldsymbol{\nu}} g(\mathbf{z}, \boldsymbol{\nu}) d\mathbf{z} \\
&= \mathbb{E}_{q(\mathbf{z}; \boldsymbol{\nu})} [\nabla_{\boldsymbol{\nu}} \log q(\mathbf{z}; \boldsymbol{\nu}) g(\mathbf{z}, \boldsymbol{\nu}) + \nabla_{\boldsymbol{\nu}} g(\mathbf{z}, \boldsymbol{\nu})]
\end{aligned}$$

Using $\nabla_{\boldsymbol{\nu}} \log q = \frac{\nabla_{\boldsymbol{\nu}} q}{q}$

Roadmap

- **Score Function Gradients**
- **Pathwise Gradients**
- **Amortized Inference**

Score Function Gradients of the ELBO

Score Function Estimator

Recall

$$\nabla_{\boldsymbol{\nu}} \mathscr{L} = \mathbb{E}_{q(\mathbf{z};\boldsymbol{\nu})}[\nabla_{\boldsymbol{\nu}} \log q(\mathbf{z};\boldsymbol{\nu}) g(\mathbf{z},\boldsymbol{\nu}) + \nabla_{\boldsymbol{\nu}} g(z,\boldsymbol{\nu})]$$

Simplify:

$$\mathbb{E}_q[\nabla_{\boldsymbol{\nu}} g(\mathbf{z},\boldsymbol{\nu})] = \mathbb{E}_q[\nabla_{\boldsymbol{\nu}} \log q(\mathbf{z};\boldsymbol{\nu})] = 0$$

Gives the gradient:

$$\nabla_{\boldsymbol{\nu}} \mathscr{L} = \mathbb{E}_{q(\mathbf{z};\boldsymbol{\nu})}[\nabla_{\boldsymbol{\nu}} \log q(\mathbf{z};\boldsymbol{\nu})(\log p(\mathbf{x},\mathbf{z}) - \log q(\mathbf{z};\boldsymbol{\nu}))]$$

Sometimes called likelihood ratio or REINFORCE gradients

[Glynn 1990; Williams, 1992; Wingate+ 2013; Ranganath+ 2014; Mnih+ 2014]

Noisy Unbiased Gradients

Gradient: $\mathbb{E}_{q(\mathbf{z};\,\boldsymbol{\nu})}[\nabla_{\boldsymbol{\nu}} \log q(\mathbf{z}; \boldsymbol{\nu})(\log p(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z}; \boldsymbol{\nu}))]$

Noisy unbiased gradients with Monte Carlo!

$$\frac{1}{S}\sum_{s=1}^{S} \nabla_{\boldsymbol{\nu}} \log q(\mathbf{z}_s; \boldsymbol{\nu})(\log p(\mathbf{x}, \mathbf{z}_s) - \log q(\mathbf{z}_s; \boldsymbol{\nu})),$$

$$\text{where } \mathbf{z}_s \sim q(\mathbf{z}; \boldsymbol{\nu})$$

Basic BBVI

Algorithm 1: Basic Black Box Variational Inference

Input : Model $\log p(\mathbf{x}, \mathbf{z})$,
Variational approximation $q(\mathbf{z}; \boldsymbol{\nu})$

Output : Variational Parameters: $\boldsymbol{\nu}$

while *not converged* **do**

$\mathbf{z}[s] \sim q$ // **Draw** S **samples from** q

$\rho = t$-th value of a Robbins Monro sequence

$\boldsymbol{\nu} = \boldsymbol{\nu} + \rho \frac{1}{S} \sum_{s=1}^{S} \nabla_{\boldsymbol{\nu}} \log q(\mathbf{z}[s]; \boldsymbol{\nu})(\log p(\mathbf{x}, \mathbf{z}[s]) - \log q(\mathbf{z}[s]; \boldsymbol{\nu}))$

$t = t + 1$

end

The requirements for inference

The noisy gradient:

$$\frac{1}{S}\sum_{s=1}^{S}\nabla_{\nu}\log q(\mathbf{z}_s;\boldsymbol{\nu})(\log p(\mathbf{x},\mathbf{z}_s)-\log q(\mathbf{z}_s;\boldsymbol{\nu})),$$

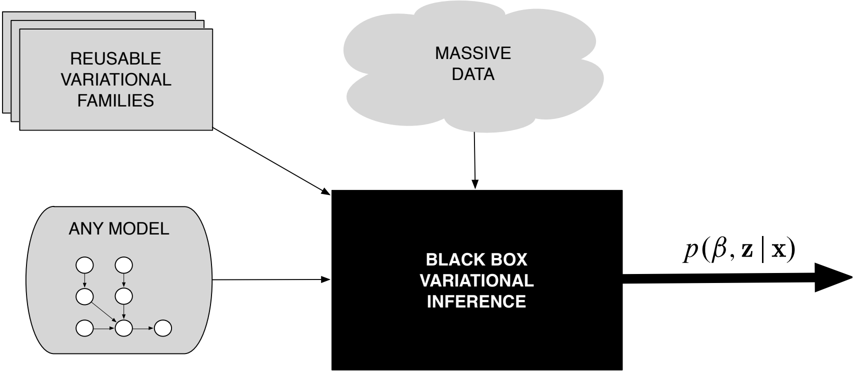
$$\text{where } \mathbf{z}_s \sim q(\mathbf{z};\boldsymbol{\nu})$$

To compute the noisy gradient of the ELBO we need

- Sampling from $q(\mathbf{z})$
- Evaluating $\nabla_{\nu}\log q(\mathbf{z};\boldsymbol{\nu})$
- Evaluating $\log p(\mathbf{x},\mathbf{z})$ and $\log q(\mathbf{z})$

There is no model specific work: black box criteria are satisfied

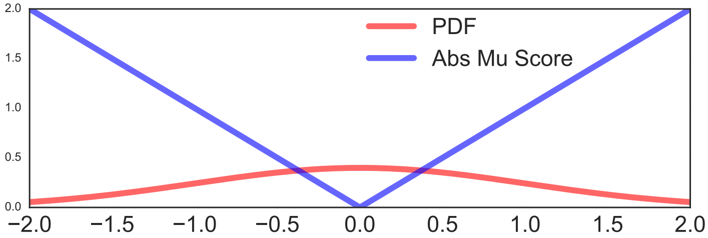
Black Box Variational Inference



Problem: Basic BBVI doesn't work

Variance of the gradient can be a problem

$$\mathrm{Var}_{q(\mathbf{z};\nu)} = \mathbb{E}_{q(\mathbf{z};\nu)}[(\nabla_\nu \log q(\mathbf{z};\nu)(\log p(\mathbf{x},\mathbf{z}) - \log q(\mathbf{z};\nu)) - \nabla_\nu \mathscr{L})^2].$$

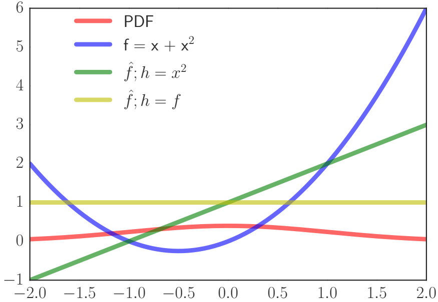


Intuition:
Sampling rare values can lead to large scores and thus high variance

Solution: Control Variates

Replace with f with $\hat{f}$ where $\mathbb{E}[\hat{f}(z)] = \mathbb{E}[f(z)]$. General such class:

$$\hat{f}(z) \triangleq f(z) - a(h(z) - \mathbb{E}[h(z)])$$

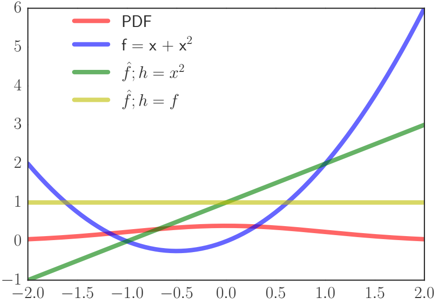


- h is a function of our choice
- a is chosen to minimize the variance
- Good h have high correlation with the original function f

Solution: Control Variates

Replace with f with $\hat{f}$ where $\mathbb{E}[\hat{f}(z)] = \mathbb{E}[f(z)]$. General such class:

$$\hat{f}(z) \triangleq f(z) - a(h(z) - \mathbb{E}[h(z)])$$

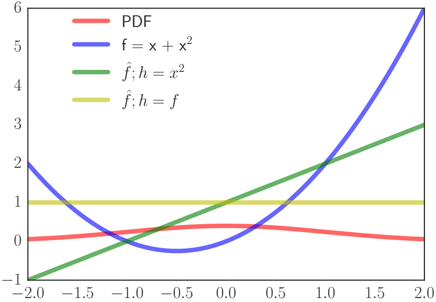


- For variational inference we need functions with known q expectation
- Set h as $\nabla_{\boldsymbol{\nu}} \log q(\mathbf{z}; \boldsymbol{\nu})$
- Simple as $\mathbb{E}_q[\nabla_{\boldsymbol{\nu}} \log q(\mathbf{z}; \boldsymbol{\nu})] = 0$ for any q

Solution: Control Variates

Replace with f with $\hat{f}$ where $\mathbb{E}[\hat{f}(z)] = \mathbb{E}[f(z)]$. General such class:

$$\hat{f}(z) \triangleq f(z) - a(h(z) - \mathbb{E}[h(z)])$$



Many of the other techniques from Monte Carlo can help:

- *Importance Sampling, Quasi Monte Carlo, Rao-Blackwellization*

[Ruiz+ 2016; Ranganath+2014; Titsias+2015; Mnih+2016]

Nonconjugate Models

- Nonlinear Time series Models
- Deep Latent Gaussian Models
- Models with Attention (such as DRAW)
- Generalized Linear Models (Poisson Regression)
- Stochastic Volatility Models
- Discrete Choice Models
- Bayesian Neural Networks
- Deep Exponential Families (e.g. Sparse Gamma or Poisson)
- Correlated Topic Model (including nonparametric variants)
- Sigmoid Belief Network

We can design models based on data rather than inference.

More Assumptions?

The current black box criteria

- Sampling from $q(\mathbf{z})$
- Evaluating $\nabla_{\boldsymbol{\nu}} \log q(\mathbf{z}; \boldsymbol{\nu})$
- Evaluating $\log p(\mathbf{x}, \mathbf{z})$ and $\log q(\mathbf{z})$

Can we make additional assumptions that are not too restrictive?

Pathwise Gradients of the ELBO

Pathwise Estimator

Assume

1. $\mathbf{z} = t(\boldsymbol{\epsilon}, \boldsymbol{\nu})$ for $\boldsymbol{\epsilon} \sim s(\boldsymbol{\epsilon})$ implies $\mathbf{z} \sim q(\mathbf{z}; \boldsymbol{\nu})$
   Example:

$$
\begin{aligned}
&\epsilon \sim \text{Normal}(0, 1) \\
&z = \epsilon\sigma + \mu \\
&\quad \to z \sim \text{Normal}(\mu, \sigma^2)
\end{aligned}
$$

2. $\log p(\mathbf{x}, \mathbf{z})$ and $\log q(\mathbf{z})$ are differentiable with respect to $\mathbf{z}$

Pathwise Estimator

Recall

$$\nabla_{\boldsymbol{\nu}} \mathscr{L} = \mathbb{E}_{q(\mathbf{z};\boldsymbol{\nu})}[\nabla_{\boldsymbol{\nu}} \log q(\mathbf{z};\boldsymbol{\nu}) g(\mathbf{z},\boldsymbol{\nu}) + \nabla_{\boldsymbol{\nu}} g(z,\boldsymbol{\nu})]$$

Rewrite using using $\mathbf{z} = t(\boldsymbol{\epsilon},\boldsymbol{\nu})$

$$\nabla_{\boldsymbol{\nu}} \mathscr{L} = \mathbb{E}_{s(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\nu}} \log s(\boldsymbol{\epsilon}) g(t(\boldsymbol{\epsilon},\boldsymbol{\nu}),\boldsymbol{\nu}) + \nabla_{\boldsymbol{\nu}} g(t(\boldsymbol{\epsilon},\boldsymbol{\nu}),\boldsymbol{\nu})]$$

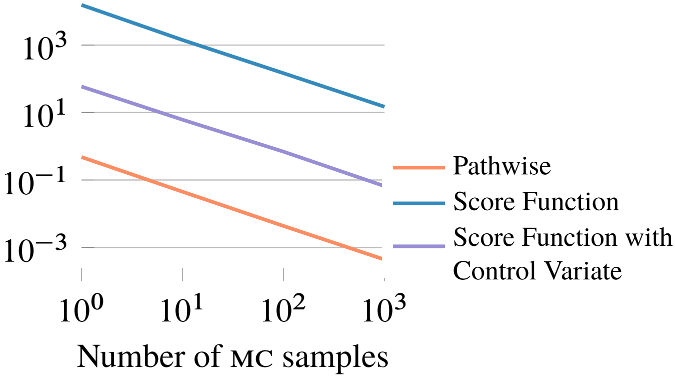
To differentiate:

$$\begin{aligned}
\nabla \mathscr{L}(\boldsymbol{\nu}) &= \mathbb{E}_{s(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\nu}} g(t(\boldsymbol{\epsilon},\boldsymbol{\nu}),\boldsymbol{\nu})] \\
&= \mathbb{E}_{s(\boldsymbol{\epsilon})}[\nabla_{\mathbf{z}}[\log p(\mathbf{x},\mathbf{z}) - \log q(\mathbf{z};\boldsymbol{\nu})] \nabla_{\boldsymbol{\nu}} t(\boldsymbol{\epsilon},\boldsymbol{\nu}) - \nabla_{\boldsymbol{\nu}} \log q(\mathbf{z};\boldsymbol{\nu})] \\
&= \mathbb{E}_{s(\boldsymbol{\epsilon})}[\nabla_{\mathbf{z}}[\log p(\mathbf{x},\mathbf{z}) - \log q(\mathbf{z};\boldsymbol{\nu})] \nabla_{\boldsymbol{\nu}} t(\boldsymbol{\epsilon},\boldsymbol{\nu})]
\end{aligned}$$

This is also known as the reparameterization gradient.

[Glasserman 1991; Fu 2006; Kingma+ 2014; Rezende+ 2014; Titsias+ 2014]

Variance Comparison



[Kucukelbir+ 2016]

Score Function Estimator vs. Pathwise Estimator

Score Function

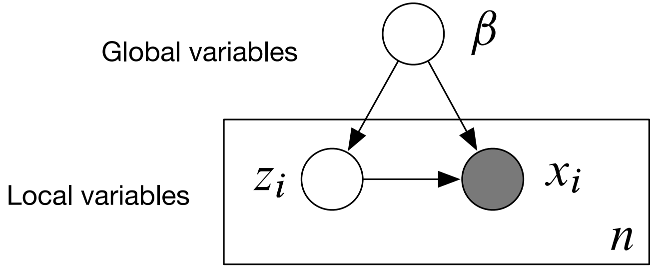
- Differentiates the density $\nabla_{\boldsymbol{\nu}} q(z; \boldsymbol{\nu})$
- Works for discrete and continuous models
- Works for large class of variational approximations
- Variance can be a big problem

Pathwise

- Differentiates the function $\nabla_z[\log p(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z}; \boldsymbol{\nu})]$
- Requires differentiable models
- Requires variational approximation to have form $\mathbf{z} = t(\boldsymbol{\epsilon}, \boldsymbol{\nu})$
- Generally better behaved variance

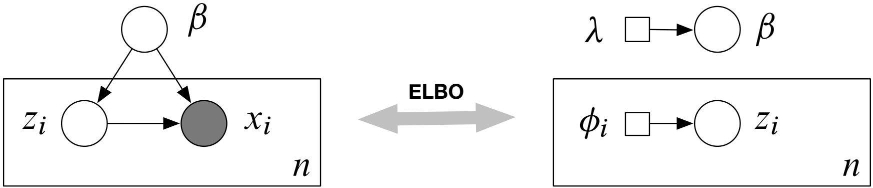
Amortized Inference

Hierarchical Models



$$p(\beta, \mathbf{z}, \mathbf{x}) = p(\beta) \prod_{i=1}^{n} p(z_i, x_i \mid \beta)$$

Mean Field Variational Approximation



SVI: Revisited

Input: data $\mathbf{x}$, model $p(\beta, \mathbf{z}, \mathbf{x})$.

Initialize λ randomly. Set ρ_t appropriately.

repeat

Sample $j \sim \text{Unif}(1, \ldots, n)$.

Set local parameter $\phi \leftarrow \mathbb{E}_\lambda\left[\eta_\ell(\beta, x_j)\right]$.

Set intermediate global parameter

$$\hat{\lambda} = \alpha + n\mathbb{E}_\phi[t(Z_j, x_j)].$$

Set global parameter

$$\lambda = (1 - \rho_t)\lambda + \rho_t\hat{\lambda}.$$

until *forever*

SVI: The problem

Input: data $\mathbf{x}$, model $p(\beta, \mathbf{z}, \mathbf{x})$.

Initialize λ randomly. Set ρ_t appropriately.

repeat

Sample $j \sim \text{Unif}(1, \ldots, n)$.

Set local parameter $\phi \leftarrow \mathbb{E}_{\lambda}\left[\eta_{\ell}(\beta, x_j)\right]$.

Set intermediate global parameter

$$\hat{\lambda} = \alpha + n\mathbb{E}_{\phi}[t(Z_j, x_j)].$$

Set global parameter

$$\lambda = (1 - \rho_t)\lambda + \rho_t \hat{\lambda}.$$

until *forever*

- These expectations are no longer tractable
- Inner stochastic optimization needed for each data point.

SVI: The problem

Input: data $\mathbf{x}$, model $p(\beta, \mathbf{z}, \mathbf{x})$.

Initialize λ randomly.   Set ρ_t appropriately.

repeat

Sample $j \sim \text{Unif}(1, \ldots, n)$.

Set local parameter $\phi \leftarrow \mathbb{E}_\lambda \left[\eta_\ell(\beta, x_j) \right]$.

Set intermediate global parameter

$$\hat{\lambda} = \alpha + n\mathbb{E}_\phi[t(Z_j, x_j)].$$

Set global parameter

$$\lambda = (1 - \rho_t)\lambda + \rho_t \hat{\lambda}.$$

until *forever*

Idea: Learn a mapping f from x_i to ϕ_i

Amortizing Inference

ELBO:

$$\mathscr{L}(\lambda, \phi_{1\dots n}) = \mathbb{E}_q\left[\log p(\beta, \mathbf{z}, \mathbf{x})\right] - \mathbb{E}_q\left[\log q(\beta; \lambda) + \sum_{i=1}^{n} q(z_i; \phi_i)\right]$$

Amortizing the ELBO with *inference network* f:

$$\mathscr{L}(\lambda, \theta) = \mathbb{E}_q\left[\log p(\beta, \mathbf{z}, \mathbf{x})\right] - \mathbb{E}_q\left[\log q(\beta; \lambda) + \sum_{i=1}^{n} q(z_i \,|\, x_i; \phi_i = f_\theta(x_i))\right]$$

[Dayan+ 1995; Heess+ 2013; Gershman+ 2014, many others]

Amortized SVI

Input: data $\mathbf{x}$, model $p(\beta, \mathbf{z}, \mathbf{x})$.

Initialize λ randomly. Set ρ_t appropriately.

repeat

Sample $\beta \sim q(\beta; \lambda)$.

Sample $j \sim \text{Unif}(1, \ldots, n)$.

Sample $z_j \sim q(z_j \,|\, x_j; \phi_\theta(x_j)$.

Compute stochastic gradients

$$
\begin{aligned}
\hat{\nabla}_\lambda \mathcal{L} &= \nabla_\lambda \log q(\beta; \lambda)(\log p(\beta) + n \log p(x_j, z_j \,|\, \beta) - \log q(\beta)) \\
\hat{\nabla}_\theta \mathcal{L} &= n \nabla_\theta \log q(z_j \,|\, x_j; \theta)(\log p(x_j, z_j \,|\, \beta) - \log q(z_j \,|\, x_k; \theta))
\end{aligned}
$$

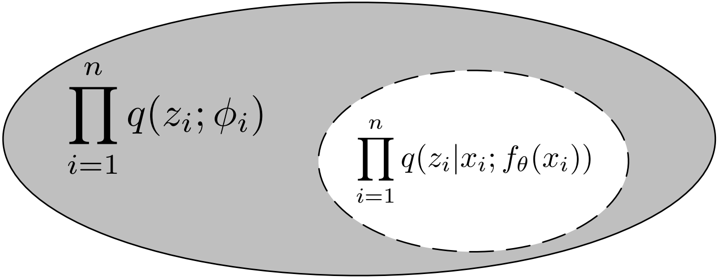
Update

$$
\begin{aligned}
\lambda &= \lambda + \rho_t \hat{\nabla}_\lambda \\
\theta &= \theta + \rho_t \hat{\nabla}_\theta .
\end{aligned}
$$

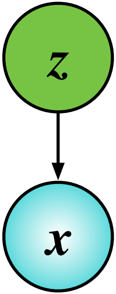
until *forever*

A computational-statistical tradeoff

- Amortized inference is faster, but admits a smaller class of approximations
- The size of the smaller class depends on the flexibility of f



Example: Variational Autoencoder (VAE)



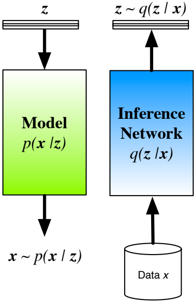
$$p(\mathbf{z}) = \mathrm{Normal}(0, 1)$$

$$p(\mathbf{x}|\mathbf{z}) = \mathrm{Normal}(\mu_\beta(\mathbf{z}), \sigma^2_\beta(\mathbf{z}))$$

μ and σ^2 are deep networks with parameters β.

[Kingma+ 2014; Rezende+ 2014]

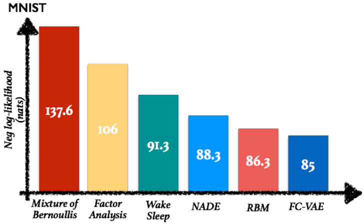
Example: Variational Autoencoder (VAE)

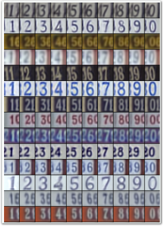


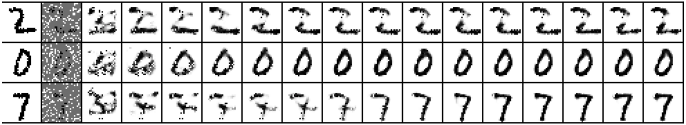
$$q(\mathbf{z}|\mathbf{x}) = \mathrm{Normal}(f_\theta^\mu(\mathbf{x}), f_\theta^{\sigma^2}(\mathbf{x}))$$

All functions are deep networks

Example: Variational Autoencoder (VAE)







Rules of Thumb for a New Model

If $\log p(\mathbf{x}, \mathbf{z})$ is $\mathbf{z}$ differentiable

- Try out an approximation q that is reparameterizable

If $\log p(\mathbf{x}, \mathbf{z})$ is not $\mathbf{z}$ differentiable

- Use score function estimator with control variates
- Add further variance reductions based on experimental evidence

Rules of Thumb for a New Model

If $\log p(\mathbf{x}, \mathbf{z})$ is $\mathbf{z}$ differentiable

- Try out an approximation q that is reparameterizable

If $\log p(\mathbf{x}, \mathbf{z})$ is not $\mathbf{z}$ differentiable

- Use score function estimator with control variates
- Add further variance reductions based on experimental evidence

General Advice:

- Use coordinate specific learning rates (e.g. RMSProp, AdaGrad)
- Annealing + Tempering
- Consider parallelizing across samples from q

Software

Systems with Variational Inference:

- Venture, WebPPL, Edward, Stan, PyMC3, Infer.net, Anglican

Good for trying out lots of models

Differentiation Tools:

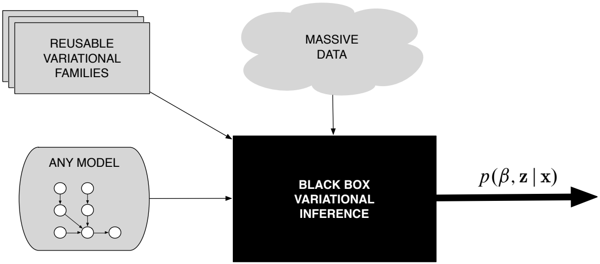
- Theano, Torch, Tensorflow, Stan Math, Caffe

Can lead to more scalable implementations of individual models

PART IV

Beyond the Mean Field

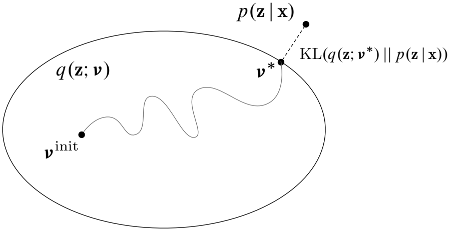
Review: Variational Bound and Optimisation

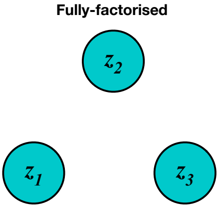


- Probabilistic modelling and variational inference.
- Scalable inference through stochastic optimisation.
- Black-box variational inference: Non-conjugate models, Monte Carlo gradient estimators and amortised inference.

These advances empower us with new way to design more flexible approximate posterior distributions $q(\mathbf{z})$

Mean-field Approximations





$$q_{MF}(z|x) = \prod_k q(z_k)$$

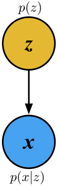
Key part of algorithm is the choice of approximate posterior $q(\mathbf{z})$.

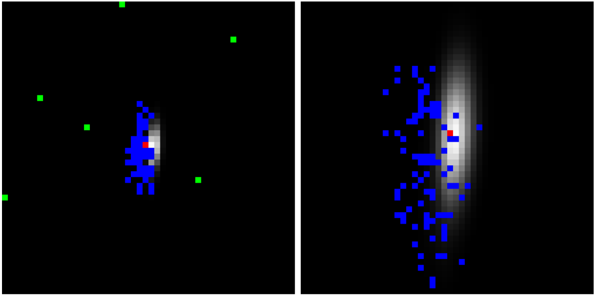
$$\log p(\mathbf{x}) \geq \mathscr{L} = \underbrace{\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}, \mathbf{z})]}_{\text{Expected likelihood}} \underbrace{- \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log q(\mathbf{z}|\mathbf{x})]}_{\text{Entropy}}$$

Mean-Field Posterior Approximations

Deep Latent Gaussian Model

Latent variable model p(x,z)

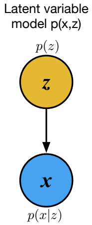


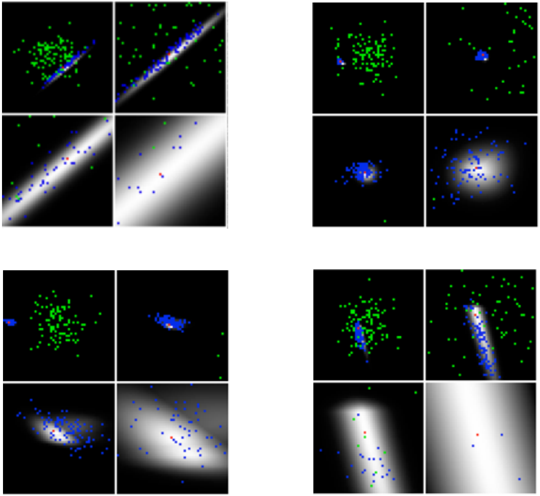


Mean-field or fully-factorised posterior is usually not sufficient

Real-world Posterior Distributions

Deep Latent Gaussian Model





Complex dependencies · Non-Gaussian distributions · Multiple modes

Families of Approximate Posteriors

Two high-level goals:

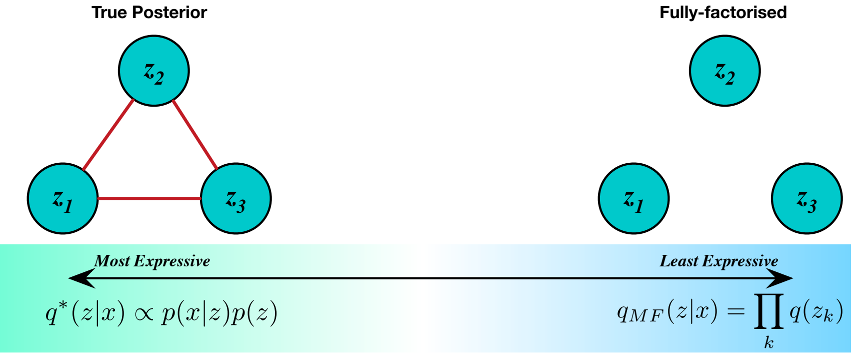
- Build richer approximate posterior distributions.

- Maintain computational efficiency and scalability.

Families of Approximate Posteriors

Two high-level goals:

- Build richer approximate posterior distributions.
- Maintain computational efficiency and scalability.

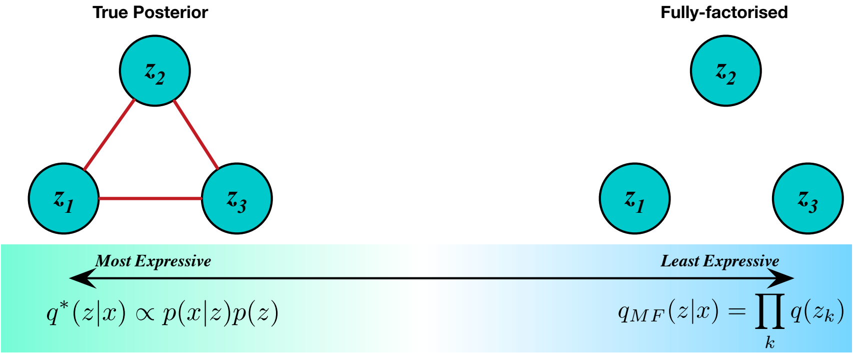


$$q^*(z|x) \propto p(x|z)p(z) \qquad q_{MF}(z|x) = \prod_k q(z_k)$$

Families of Approximate Posteriors

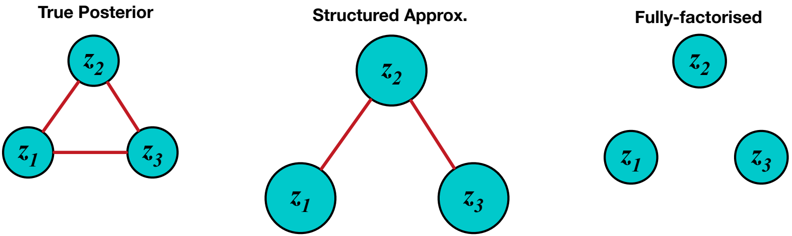
Two high-level goals:

- Build richer approximate posterior distributions.
- Maintain computational efficiency and scalability.



Same as the problem of specifying a model of the data itself.

Structured Posterior Approximations



Most Expressive ⟵ ⟶ *Least Expressive*

$$q^*(z|x) \propto p(x|z)p(z) \qquad q(z) = \prod_k q_k(z_k|\{z_j\}_{j\neq k}) \qquad q_{MF}(z|x) = \prod_k q(z_k)$$

Structured mean field: Introduce any form of dependency to provide a richer approximating class of distributions.

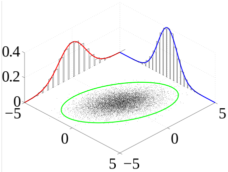
[Saul and Jordan, 1996.]

Gaussian Approximate Posteriors

Use a correlated Gaussian:

$$q_G(\mathbf{z}; \boldsymbol{\nu}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

Variational parameters $\boldsymbol{\nu} = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$

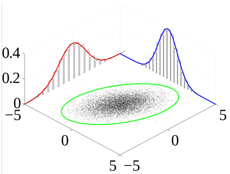


Gaussian Approximate Posteriors

Use a correlated Gaussian:

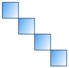
$$q_G(\mathbf{z};\boldsymbol{\nu}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu},\boldsymbol{\Sigma})$$

Variational parameters $\boldsymbol{\nu} = \{\boldsymbol{\mu},\boldsymbol{\Sigma}\}$



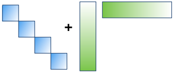
Covariance models: Structure of covariance $\boldsymbol{\Sigma}$ describes dependency. Full covariance is richest, but computationally expensive.

$\mathrm{diag}(\alpha_1,\ldots,\alpha_K)$



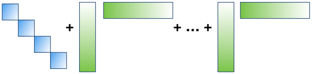
Mean-field

$\mathrm{diag}(\alpha_1,\ldots,\alpha_K)$
$+\mathbf{u}\mathbf{u}^\top$



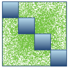
Rank-1

$\mathrm{diag}(\alpha_1,\ldots,\alpha_K)$
$+\sum_j \mathbf{u_j}\mathbf{u_j}^\top$

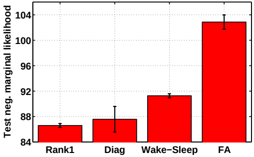


Rank-J

$\mathbf{U}\mathbf{U}^\top$



Full

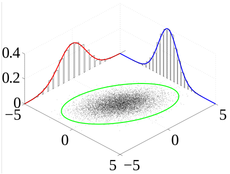


Gaussian Approximate Posteriors

Use a correlated Gaussian:

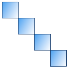
$$q_G(\mathbf{z};\boldsymbol{\nu}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu},\boldsymbol{\Sigma})$$

Variational parameters $\boldsymbol{\nu} = \{\boldsymbol{\mu},\boldsymbol{\Sigma}\}$



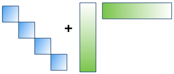
Covariance models: Structure of covariance $\boldsymbol{\Sigma}$ describes dependency. Full covariance is richest, but computationally expensive.

$\mathrm{diag}(\alpha_1,\ldots,\alpha_K)$



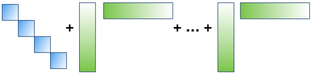
Mean-field

$\mathrm{diag}(\alpha_1,\ldots,\alpha_K)$
$+\mathbf{u}\mathbf{u}^\top$



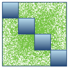
Rank-1

$\mathrm{diag}(\alpha_1,\ldots,\alpha_K)$
$+\sum_j \mathbf{u_j}\mathbf{u_j}^\top$

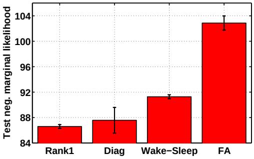


Rank-J

$\mathbf{U}\mathbf{U}^\top$



Full

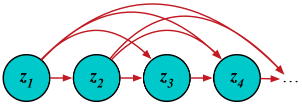


Approximate posterior is always Gaussian.

Beyond Gaussian Approximations

Autoregressive distributions: Impose an ordering and non-linear dependency on all preceding variables.

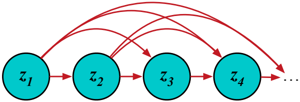
$$q_{AR}(\mathbf{z}; \boldsymbol{\nu}) = \prod_k q_k(z_k | z_{<k}; \boldsymbol{\nu}_k)$$



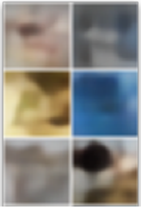
Beyond Gaussian Approximations

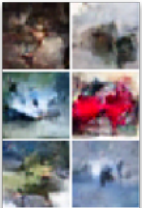
Autoregressive distributions: Impose an ordering and non-linear dependency on all preceding variables.

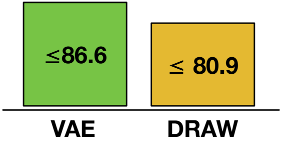
$$q_{AR}(\mathbf{z};\boldsymbol{\nu}) = \prod_k q_k(z_k|z_{<k};\boldsymbol{\nu}_k)$$



Compare DLGMs: Using Gaussian mean field (VAE) vs. auto-regressive posterior (DRAW) in fully-connected DLGMs on CIFAR10.





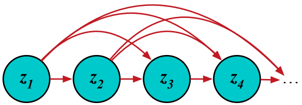


[Gregor et al., 2015]

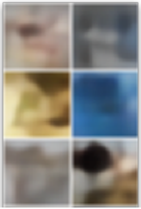
Beyond Gaussian Approximations

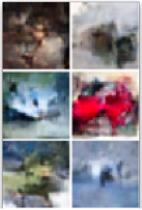
Autoregressive distributions: Impose an ordering and non-linear dependency on all preceding variables.

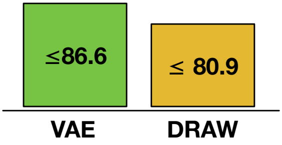
$$q_{AR}(\mathbf{z};\boldsymbol{\nu}) = \prod_k q_k(z_k|z_{<k};\boldsymbol{\nu}_k)$$



Compare DLGMs: Using Gaussian mean field (VAE) vs. auto-regressive posterior (DRAW) in fully-connected DLGMs on CIFAR10.





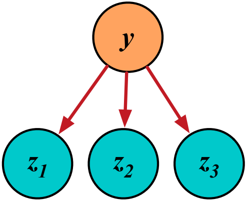


[Gregor et al., 2015]

Joint-distribution non-Gaussian, although conditionals are.

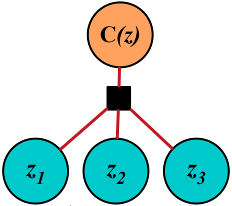
More Structured Posteriors

Mixture model



$$q_{mm}(\mathbf{z};\boldsymbol{\nu}) = \sum_r \rho_r q_r(\mathbf{z}_r|\boldsymbol{\nu}_r)$$

Linking functions

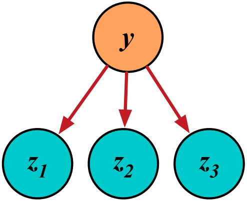


$$q_{lm}(\mathbf{z};\boldsymbol{\nu}) = \left(\prod_k q_k(z_k|\boldsymbol{\nu}_k)\right) C(\mathbf{z};\boldsymbol{\nu}_{k+1})$$

[Saul and Jordan, 1996, Tran et al., 2016]

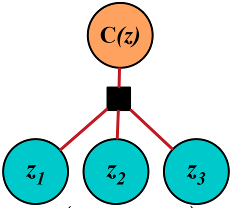
More Structured Posteriors

Mixture model



$$q_{mm}(\mathbf{z};\boldsymbol{\nu}) = \sum_r \rho_r q_r(\mathbf{z}_r|\boldsymbol{\nu}_r)$$

Linking functions



$$q_{lm}(\mathbf{z};\boldsymbol{\nu}) = \left(\prod_k q_k(z_k|\boldsymbol{\nu}_k)\right) C(\mathbf{z};\boldsymbol{\nu}_{k+1})$$

[Saul and Jordan, 1996, Tran et al., 2016]

Suggests a general way to improve posterior approximations:

Introduce additional variables that induce dependencies, but that remain tractable and efficient.

Designing Richer Posteriors

1. **Introduce new variables $\boldsymbol{\omega}$** that help to form a richer approximate posterior distribution.

$$q(\mathbf{z}; \boldsymbol{\nu}) = \int q(\mathbf{z}, \boldsymbol{\omega}; \boldsymbol{\nu}) d\boldsymbol{\omega}$$

Designing Richer Posteriors

1. **Introduce new variables $\boldsymbol{\omega}$** that help to form a richer approximate posterior distribution.

$$q(\mathbf{z}; \boldsymbol{\nu}) = \int q(\mathbf{z}, \boldsymbol{\omega}; \boldsymbol{\nu}) d\boldsymbol{\omega}$$

2. **Adapt bound** to compute entropy or a bound.

$$\log p(\mathbf{x}) \geq \mathcal{L} = \underbrace{\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}, \mathbf{z})]}_{\text{Expected likelihood}} \underbrace{-\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log q(\mathbf{z}|\mathbf{x})]}_{\text{Entropy}}$$

Designing Richer Posteriors

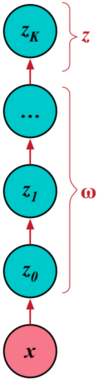
1. **Introduce new variables $\boldsymbol{\omega}$** that help to form a richer approximate posterior distribution.

$$q(\mathbf{z}; \boldsymbol{\nu}) = \int q(\mathbf{z}, \boldsymbol{\omega}; \boldsymbol{\nu}) d\boldsymbol{\omega}$$

2. **Adapt bound** to compute entropy or a bound.

$$\log p(\mathbf{x}) \geq \mathcal{L} = \underbrace{\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}, \mathbf{z})]}_{\text{Expected likelihood}} \underbrace{- \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log q(\mathbf{z}|\mathbf{x})]}_{\text{Entropy}}$$

3. Maintain **computational efficiency**: linear in number of latent variables.



Designing Richer Posteriors

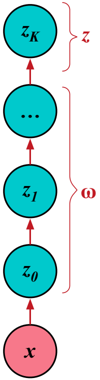
1. **Introduce new variables** $\boldsymbol{\omega}$ that help to form a richer approximate posterior distribution.

$$q(\mathbf{z};\nu) = \int q(\mathbf{z},\boldsymbol{\omega};\nu)d\boldsymbol{\omega}$$

2. **Adapt bound** to compute entropy or a bound.

$$\log p(\mathbf{x}) \geq \mathscr{L} = \underbrace{\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x},\mathbf{z})]}_{\text{Expected likelihood}} \underbrace{-\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log q(\mathbf{z}|\mathbf{x})]}_{\text{Entropy}}$$

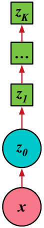
3. Maintain **computational efficiency**: linear in number of latent variables.



Look at two different approaches

- ***Change-of-variables:*** Normalising flows and invertible transforms.
- ***Auxiliary variables:*** Entropy bounds, Monte Carlo sampling.

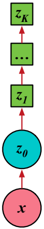
Approximations using Change-of-variables



Exploit the rule for change of variables for random variables:

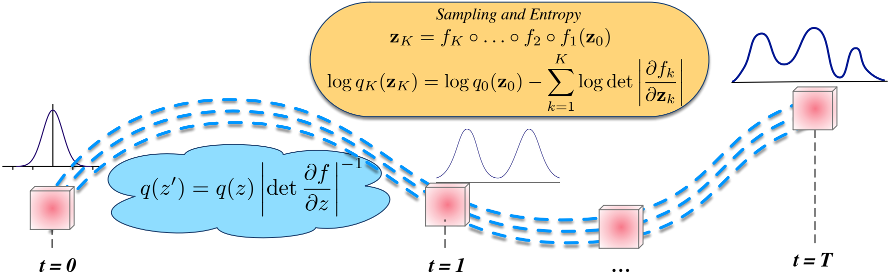
- Begin with an initial distribution $q_0(\mathbf{z}_0|\mathbf{x})$.
- Apply a sequence of K invertible functions f_k.

Approximations using Change-of-variables



Exploit the rule for change of variables for random variables:

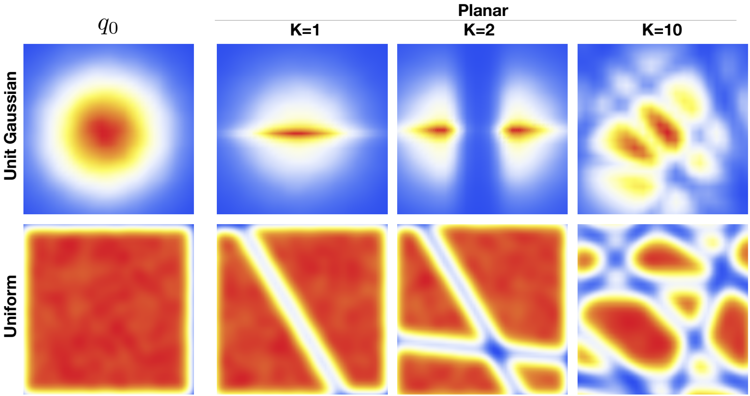
- Begin with an initial distribution $q_0(\mathbf{z}_0|\mathbf{x})$.
- Apply a sequence of K invertible functions f_k.



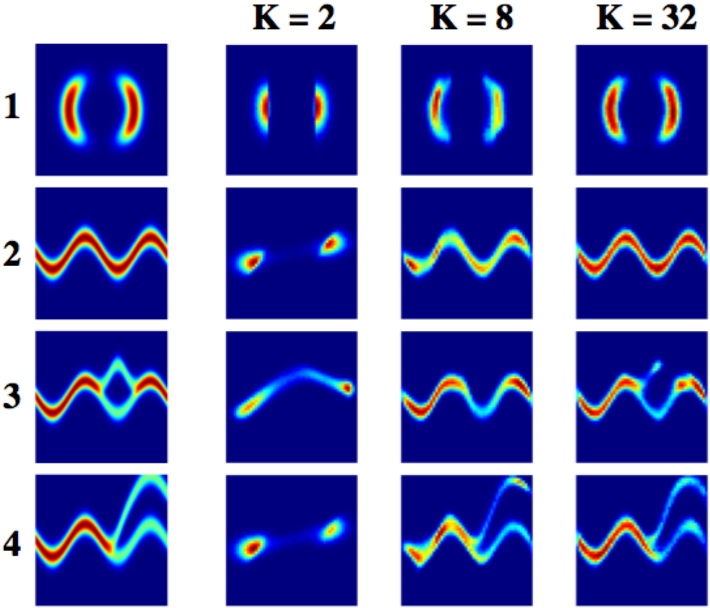
Distribution flows through a sequence of invertible transforms

[Rezende and Mohamed, 2015]

Normalising Flows



Normalising Flows



Choice of Transformation Function

$$\mathscr{L} = \mathbb{E}_{q_0(\mathbf{z}_0)}[\log p(\mathbf{x}, \mathbf{z}_K)] - \mathbb{E}_{q_0(\mathbf{z}_0)}[\log q_0(\mathbf{z}_0)] - \mathbb{E}_{q_0(\mathbf{z}_0)}\left[\sum_{k=1}^{K} \log \det \left| \frac{\partial f_k}{\partial \mathbf{z}_k} \right| \right]$$

Choice of Transformation Function

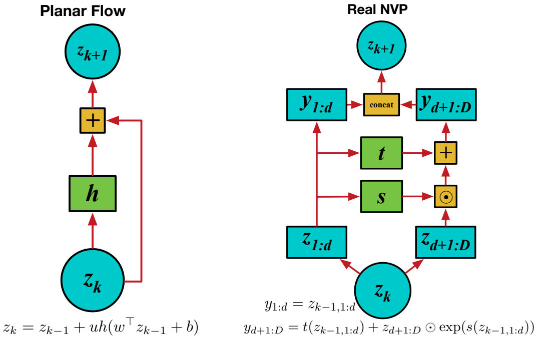
$$\mathcal{L} = \mathbb{E}_{q_0(\mathbf{z}_0)}[\log p(\mathbf{x}, \mathbf{z}_K)] - \mathbb{E}_{q_0(\mathbf{z}_0)}[\log q_0(\mathbf{z}_0)] - \mathbb{E}_{q_0(\mathbf{z}_0)}\left[\sum_{k=1}^{K} \log \det \left| \frac{\partial f_k}{\partial \mathbf{z}_k} \right| \right]$$

- Begin with a fully-factorised Gaussian and improve by change of variables.
- Triangular Jacobians allow for computational efficiency.

Choice of Transformation Function

$$\mathcal{L} = \mathbb{E}_{q_0(\mathbf{z}_0)}[\log p(\mathbf{x}, \mathbf{z}_K)] - \mathbb{E}_{q_0(\mathbf{z}_0)}[\log q_0(\mathbf{z}_0)] - \mathbb{E}_{q_0(\mathbf{z}_0)}\left[\sum_{k=1}^{K} \log \det \left| \frac{\partial f_k}{\partial \mathbf{z}_k} \right| \right]$$

- Begin with a fully-factorised Gaussian and improve by change of variables.
- Triangular Jacobians allow for computational efficiency.

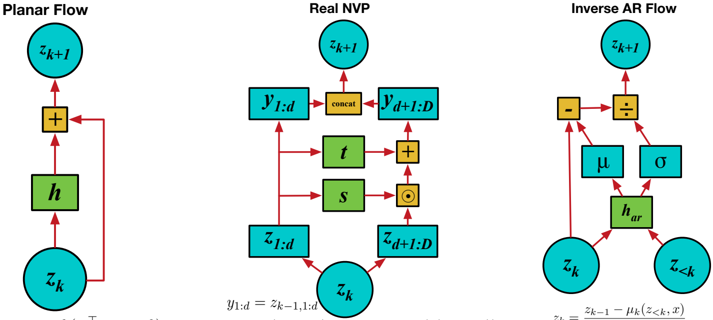


[Rezende and Mohamed, 2016; Dinh et al., 2016; Kingma et al., 2016]

Choice of Transformation Function

$$\mathscr{L} = \mathbb{E}_{q_0(\mathbf{z}_0)}[\log p(\mathbf{x}, \mathbf{z}_K)] - \mathbb{E}_{q_0(\mathbf{z}_0)}[\log q_0(\mathbf{z}_0)] - \mathbb{E}_{q_0(\mathbf{z}_0)}\left[\sum_{k=1}^{K} \log \det \left| \frac{\partial f_k}{\partial \mathbf{z}_k} \right| \right]$$

- Begin with a fully-factorised Gaussian and improve by change of variables.
- Triangular Jacobians allow for computational efficiency.



$z_k = z_{k-1} + uh(w^\top z_{k-1} + b)$

$y_{1:d} = z_{k-1,1:d}$

$y_{d+1:D} = t(z_{k-1,1:d}) + z_{d+1:D} \odot \exp(s(z_{k-1,1:d}))$

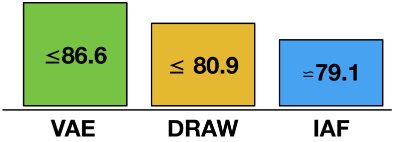
$z_k = \dfrac{z_{k-1} - \mu_k(z_{<k}, x)}{\sigma_k(z_{<k}, x)}$

[Rezende and Mohamed, 2016; Dinh et al., 2016; Kingma et al., 2016]

Linear time computation of the determinant and its gradient.

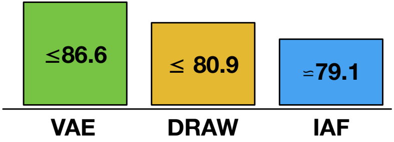
Modelling Improvements

VAE-type algorithms on the MNIST benchmark

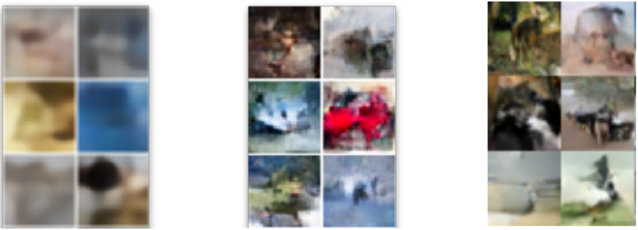


Modelling Improvements

VAE-type algorithms on the MNIST benchmark



Samples generated from model on CIFAR10 images



Hierarchical Approximate Posteriors

We can use **'latent variables'** $\boldsymbol{\omega}$ to enrich the approximate posterior distribution, like we do for our density models.

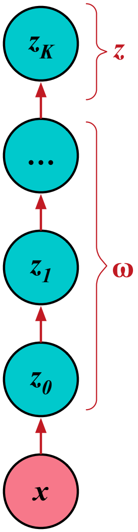
$$q(\mathbf{z}|\mathbf{x}) = \int q(\mathbf{z}|\boldsymbol{\omega}, \mathbf{x}) q(\boldsymbol{\omega}|\mathbf{x}) d\boldsymbol{\omega}$$

Hierarchical Approximate Posteriors

We can use **'latent variables'** $\boldsymbol{\omega}$ to enrich the approximate posterior distribution, like we do for our density models.

$$q(\mathbf{z}|\mathbf{x}) = \int q(\mathbf{z}|\boldsymbol{\omega}, \mathbf{x}) q(\boldsymbol{\omega}|\mathbf{x}) d\boldsymbol{\omega}$$

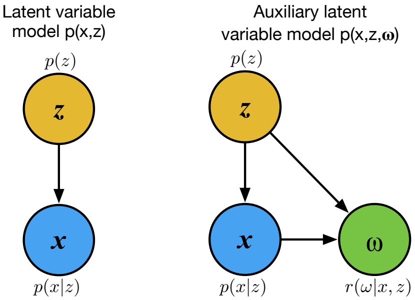
- Use a **hierarchical model** for the approximate posterior.
- **Stochastic variables** $\boldsymbol{\omega}$ rather than deterministic in the change-of-variables approach.
- **Both continuous and discrete** latent variables can be modelled.



[Ranganath et al., 2016]

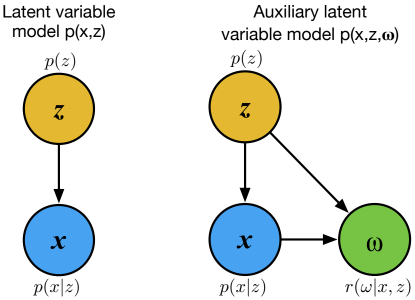
Auxiliary-variable Methods

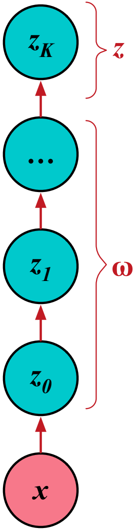
Modify the model to include $\boldsymbol{\omega} = (\mathbf{z}_0, \ldots, \mathbf{z}_{K-1})$.



Auxiliary-variable Methods

Modify the model to include $\boldsymbol{\omega} = (\mathbf{z}_0, \ldots, \mathbf{z}_{K-1})$.



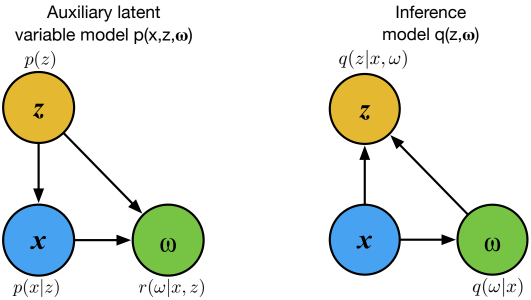


- ***Auxiliary variables*** leave the original model unchanged.
- They capture structure of correlated variables because they turn the posterior into a mixture of distributions $q(\mathbf{z}|\mathbf{x}, \boldsymbol{\omega})$.

[Agakov and Barber, 2004; Maaløe et al., 2016]

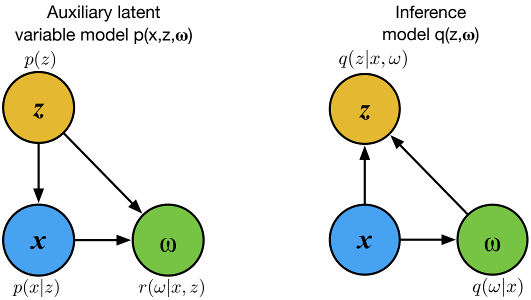
Auxiliary Variational Lower Bounds

Standard bound: $\log p(\mathbf{x}) \geq \mathscr{L} = \underbrace{\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}, \mathbf{z})]}_{\text{Expected likelihood}} \underbrace{- \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log q(\mathbf{z}|\mathbf{x})]}_{\text{Entropy}}$



Auxiliary Variational Lower Bounds

Standard bound: $\log p(\mathbf{x}) \geq \mathcal{L} = \underbrace{\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}, \mathbf{z})]}_{\text{Expected likelihood}} \underbrace{- \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log q(\mathbf{z}|\mathbf{x})]}_{\text{Entropy}}$



Auxiliary variational bound: Bound the entropy for tractability.

$$\log p(\mathbf{x}) \geq \mathbb{E}_{q(\boldsymbol{\omega},\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}, \mathbf{z}) + \log r(\boldsymbol{\omega}|\mathbf{z}, \mathbf{x})] - \mathbb{E}_{q(\boldsymbol{\omega},\mathbf{z}|\mathbf{x})}[\log q(\mathbf{z}, \boldsymbol{\omega}|\mathbf{x})]$$
$$\geq \mathcal{L} - \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\mathrm{KL}[q(\boldsymbol{\omega}|\mathbf{z}, \mathbf{x}) \| r(\omega|\mathbf{z}, \mathbf{x})]]$$

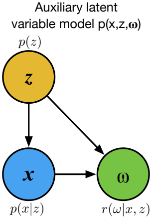
[Salimans et al., 2015; Ranganath et al., 2016; Maaløe et al., 2016]

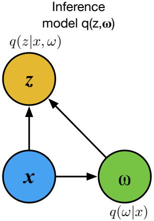
Auxiliary Variational Methods

Choose an auxiliary prior $r(\boldsymbol{\omega}|\mathbf{z},\mathbf{x})$ and auxiliary posterior $q(\boldsymbol{\omega}|\mathbf{x},\mathbf{z})$

Auxiliary Variational Methods

Choose an auxiliary prior $r(\boldsymbol{\omega}|\mathbf{z},\mathbf{x})$ and auxiliary posterior $q(\boldsymbol{\omega}|\mathbf{x},\mathbf{z})$



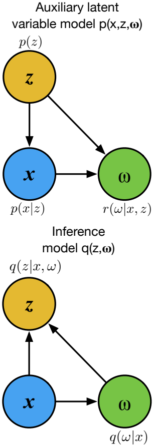


- Hamiltonian flow: $r(\boldsymbol{\omega}) = \mathcal{N}(\boldsymbol{\omega}|\mathbf{0},\mathbf{M})$
- Input-dependent Gaussian: $r(\boldsymbol{\omega}|\mathbf{x},\mathbf{z})$
- Auto-regressive: $r(\boldsymbol{\omega}|\mathbf{x},\mathbf{z}) = \prod_t r(\boldsymbol{\omega}_t|f_\theta(\boldsymbol{\omega}_{<t},\mathbf{x}))$

- $q(\boldsymbol{\omega}|\mathbf{x},\mathbf{z})$ can be a mixture model, normalising flow, Gaussian process.

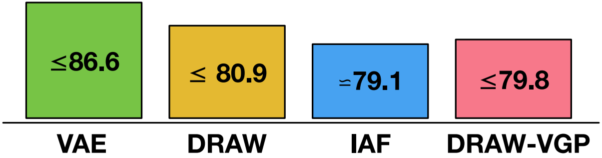
Auxiliary Variational Methods

Choose an auxiliary prior $r(\boldsymbol{\omega}|\mathbf{z},\mathbf{x})$ and auxiliary posterior $q(\boldsymbol{\omega}|\mathbf{x},\mathbf{z})$



- Hamiltonian flow: $r(\boldsymbol{\omega}) = \mathcal{N}(\boldsymbol{\omega}|\mathbf{0},\mathbf{M})$
- Input-dependent Gaussian: $r(\boldsymbol{\omega}|\mathbf{x},\mathbf{z})$
- Auto-regressive: $r(\boldsymbol{\omega}|\mathbf{x},\mathbf{z}) = \prod_t r(\boldsymbol{\omega}_t|f_\theta(\boldsymbol{\omega}_{<t},\mathbf{x}))$

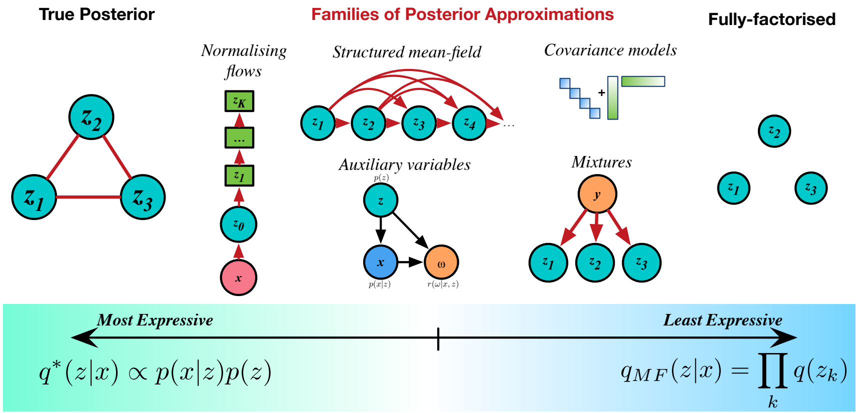
- $q(\boldsymbol{\omega}|\mathbf{x},\mathbf{z})$ can be a mixture model, normalising flow, Gaussian process.



[Tran et al., 2016]

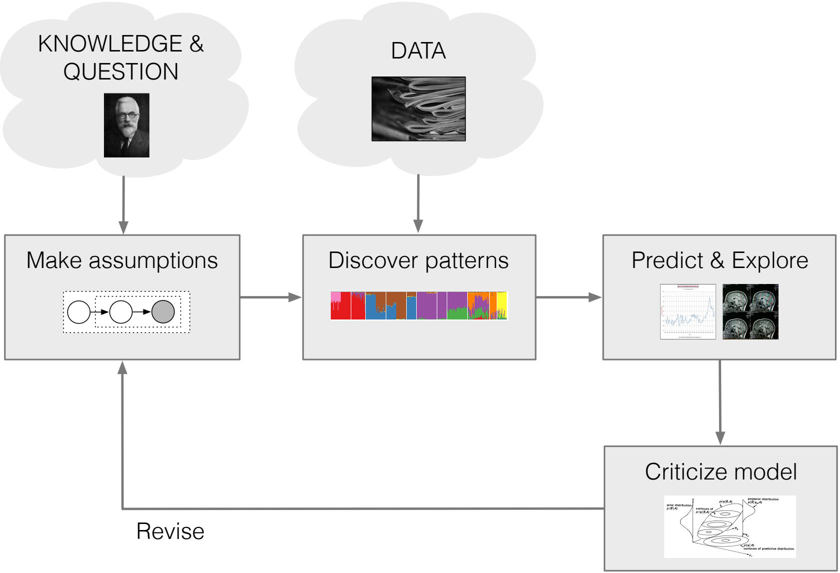
Easy sampling, evaluation of bound and gradients.

Summary



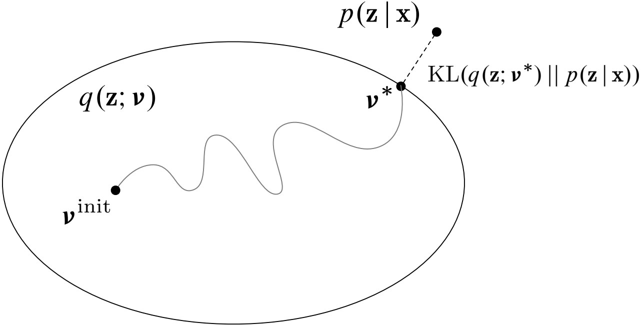
True Posterior
Families of Posterior Approximations
Fully-factorised
Normalising flows
Structured mean-field
Covariance models
Auxiliary variables
Mixtures
Most Expressive
Least Expressive
$q^*(z|x) \propto p(x|z)p(z)$
$q_{MF}(z|x) = \prod_k q(z_k)$

Choosing your Approximation



Summary

Variational Inference:
Foundations and Modern Methods



VI approximates difficult quantities from complex models.

With **stochastic optimization** we can

- scale up VI to massive data
- enable VI on a wide class of difficult models
- enable VI with elaborate and flexible families of approximations

Bibliography

Introductory Variational Inference

- Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. Machine learning, 37(2), 183-233.
- Beal, Matthew James. Variational algorithms for approximate Bayesian inference. Diss. University of London, 2003.
- Wainwright, Martin J., and Michael I. Jordan. "Graphical models, exponential families, and variational inference." Foundations and Trends in Machine Learning 1, no. 1-2 (2008): 1-305.

Bibliography

Applications of Variational Inference

- Frey, Brendan J., and Geoffrey E. Hinton. "Variational learning in nonlinear Gaussian belief networks." Neural Computation 11, no. 1 (1999): 193-213.
- Eslami, S. M., Heess, N., Weber, T., Tassa, Y., Kavukcuoglu, K., and Hinton, G. E. Attend, Infer, Repeat: Fast Scene Understanding with Generative Models. NIPS (2016).
- Rezende, Danilo Jimenez, Shakir Mohamed, Ivo Danihelka, Karol Gregor, and Daan Wierstra. "One-Shot Generalization in Deep Generative Models." ICML (2016).
- Kingma, Diederik P., Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. "Semi-supervised learning with deep generative models." In Advances in Neural Information Processing Systems, pp. 3581-3589. 2014.

Bibliography

Monte Carlo Gradient Estimation

- Pierre L'Ecuyer, Note: On the interchange of derivative and expectation for likelihood ratio derivative estimators, Management Science, 1995
- Peter W Glynn, Likelihood ratio gradient estimation for stochastic systems, Communications of the ACM, 1990
- Michael C Fu, Gradient estimation, Handbooks in operations research and management science, 2006
- Ronald J Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Machine learning, 1992
- Paul Glasserman, Monte Carlo methods in financial engineering, 2003
- Omiros Papaspiliopoulos, Gareth O Roberts, Martin Skold, A general framework for the parametrization of hierarchical models, Statistical Science, 2007
- Michael C Fu, Gradient estimation, Handbooks in operations research and management science, 2006
- Rajesh Ranganath, Sean Gerrish, and David M. Blei. "Black Box Variational Inference." In AISTATS, pp. 814-822. 2014.
- Andriy Mnih, and Karol Gregor. "Neural variational inference and learning in belief networks." arXiv preprint arXiv:1402.0030 (2014).

Bibliography

Monte Carlo Gradient Estimation (cont.)

- Michalis Titsias and Miguel Lázaro-Gredilla. "Doubly stochastic variational Bayes for non-conjugate inference." (2014).
- David Wingate and Theophane Weber. "Automated variational inference in probabilistic programming." arXiv preprint arXiv:1301.1299 (2013).
- John Paisley, David Blei, and Michael Jordan. "Variational Bayesian inference with stochastic search." arXiv preprint arXiv:1206.6430 (2012).
- Durk Kingma and Max Welling. "Auto-encoding Variational Bayes." ICLR (2014).
- Danilo Jimenez Rezende, Shakir Mohamed, Daan Wierstra. "Stochastic Backpropagation and Approximate Inference in Deep Generative Models." ICML (2014).

Bibliography

Amortized Inference

- Dayan, Peter, Geoffrey E. Hinton, Radford M. Neal, and Richard S. Zemel. "The helmholtz machine." Neural computation 7, no. 5 (1995): 889-904.
- Gershman, Samuel J., and Noah D. Goodman. "Amortized inference in probabilistic reasoning." In Proceedings of the 36th Annual Conference of the Cognitive Science Society. 2014.
- Heess, Nicolas, Daniel Tarlow, and John Winn. "Learning to pass expectation propagation messages." In Advances in Neural Information Processing Systems, pp. 3219-3227. 2013.
- Jitkrittum, Wittawat, Arthur Gretton, Nicolas Heess, S. M. Eslami, Balaji Lakshminarayanan, Dino Sejdinovic, and ZoltÃąn SzabÃş. "Kernel-based just-in-time learning for passing expectation propagation messages." arXiv preprint arXiv:1503.02551 (2015).
- Korattikara, Anoop, Vivek Rathod, Kevin Murphy, and Max Welling. "Bayesian dark knowledge." arXiv preprint arXiv:1506.04416 (2015).

Bibliography

Structured Mean Field

- Jaakkola, T. S., and Jordan, M. I. (1998). Improving the mean field approximation via the use of mixture distributions. In Learning in graphical models (pp. 163-173). Springer Netherlands.
- Saul, L.K. and Jordan, M.I., 1996. Exploiting tractable substructures in intractable networks. Advances in neural information processing systems, pp.486-492.
- Gregor, Karol, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. "DRAW: A recurrent neural network for image generation." ICML (2015).
- Gershman, S., Hoffman, M. and Blei, D., 2012. Nonparametric variational inference. arXiv preprint arXiv:1206.4665.

Bibliography

Change-of-variables and Normalising Flows

- Tabak, E. G., and Cristina V. Turner. "A family of nonparametric density estimation algorithms." Communications on Pure and Applied Mathematics 66, no. 2 (2013): 145-164.
- Rezende, Danilo Jimenez, and Shakir Mohamed. "Variational inference with normalizing flows." ICML (2015).
- Kingma, D.P., Salimans, T. and Welling, M., 2016. Improving variational inference with inverse autoregressive flow. arXiv preprint arXiv:1606.04934.
- Dinh, L., Sohl-Dickstein, J. and Bengio, S., 2016. Density estimation using Real NVP. arXiv preprint arXiv:1605.08803.

Bibliography

Auxiliary Variational Methods

- Felix V. Agakov, and David Barber. "An auxiliary variational method." NIPS (2004).
- Rajesh Ranganath, Dustin Tran, and David M. Blei. "Hierarchical Variational Models." ICML (2016).
- Lars Maaløe et al. "Auxiliary Deep Generative Models." ICML (2016).
- Tim Salimans, Durk Kingma, Max Welling. "Markov chain Monte Carlo and variational inference: Bridging the gap. In International Conference on Machine Learning." ICML (2015).

Bibliography

Related Variational Objectives

- Yuri Burda, Roger Grosse, Ruslan Salakhutidinov. "Importance weighted autoencoders." ICLR (2015).
- Yingzhen Li, Richard E. Turner. "Rényi divergence variational inference." NIPS (2016).
- Guillaume and Balaji Lakshminarayanan. "Approximate Inference with the Variational Holder Bound." ArXiv (2015).
- José Miguel Hernández-Lobato, Yingzhen Li, Daniel Hernández-Lobato, Thang Bui, and Richard E. Turner. Black-box α-divergence Minimization. ICML (2016).
- Rajesh Ranganath, Jaan Altosaar, Dustin Tran, David M. Blei. Operator Variational Inference. NIPS (2016).

Bibliography

Discrete Latent Variable Models and Posterior Approximations

- Radford Neal. "Learning stochastic feedforward networks." Tech. Rep. CRG-TR-90-7: Department of Computer Science, University of Toronto (1990).
- Lawrence K. Saul, Tommi Jaakkola, and Michael I. Jordan. "Mean field theory for sigmoid belief networks." Journal of artificial intelligence research 4, no. 1 (1996): 61-76.
- Karol Gregor, Ivo Danihelka, Andriy Mnih, Charles Blundell, and Daan Wierstra. "Deep autoregressive networks." ICML (2014).
- Rajesh Ranganath, Linpeng Tang, Laurent Charlin, and David M. Blei. "Deep Exponential Families." AISTATS (2015).
- Rajesh Ranganath, Dustin Tran, and David M. Blei. "Hierarchical Variational Models." ICML (2016).