Variational Inference for Machine Learning

Shakir Mohamed

Google DeepMind

shakirm.com @shakir_za

18 February 2015, Imperial College, London

Abstract

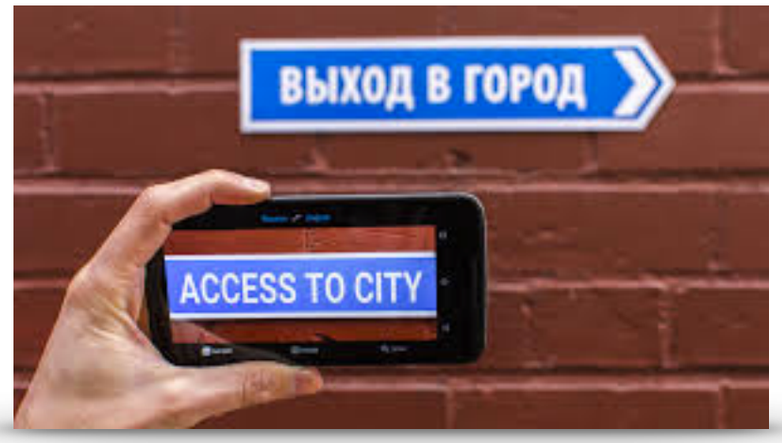
Variational inference is one of the tools that now lies at the heart of the modern data analysis lifecycle. Variational inference is the term used to encompass approximation techniques for the solution of intractable integrals and complex distributions and operates by transforming the hard problem of integration into one of optimisation. As a result, using variational inference we are now able to derive algorithms that allow us to apply increasingly complex probabilistic models to ever larger data sets on ever more powerful computing resources.

This tutorial is meant as a broad introduction to modern approaches for approximate, large-scale inference and reasoning in probabilistic models. It is designed to be of interest to both new and experienced researchers in machine learning, statistics and engineering and is intended to leave everyone with an understanding of an invaluable tool for probabilistic inference and its connections to a broad range of fields, such as Bayesian analysis, deep learning, information theory, and statistical mechanics.

The tutorial will begin by motivating probabilistic data analysis and the problem of inference for statistical applications, such as density estimation, missing data imputation and model selection, and for industrial problems in search and recommendation, text mining and community discovery. We will then examine importance sampling as one widely-used Monte Carlo inference mechanism and from this begin our journey towards the variational approach for inference. The principle of variational inference and basic tools from variational calculus will be introduced, as well as the class of latent Gaussian models that will be used throughout the tutorial as a running example. Using this foundation, we shall discuss different approaches for approximating posterior distributions, the smorgasbord of techniques for optimising the variational objective function, a discussion of implementation and large-scale applications, a brief look at the available theory for variational methods, and an overview of other variational problems in machine learning and statistics.

Machine Learning Problems

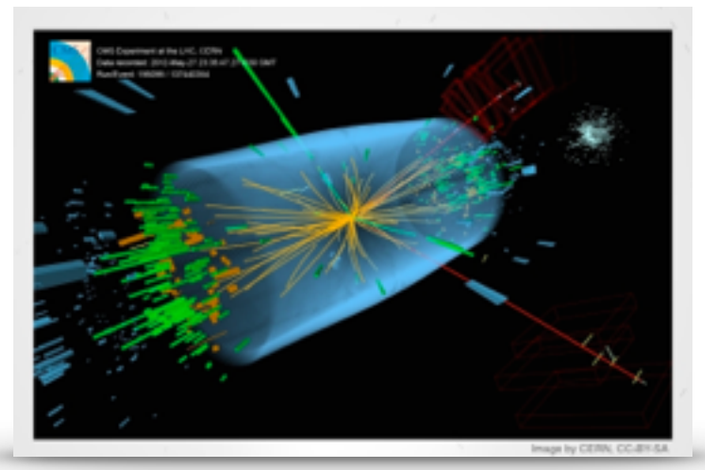
Google translate and word lens



Automatic image captioning

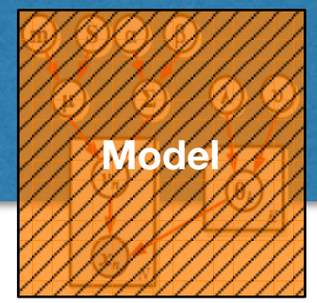
Netflix challenge

Finding the Higg's boson



Heritage Health prize - predict hopitalisation using insurance data

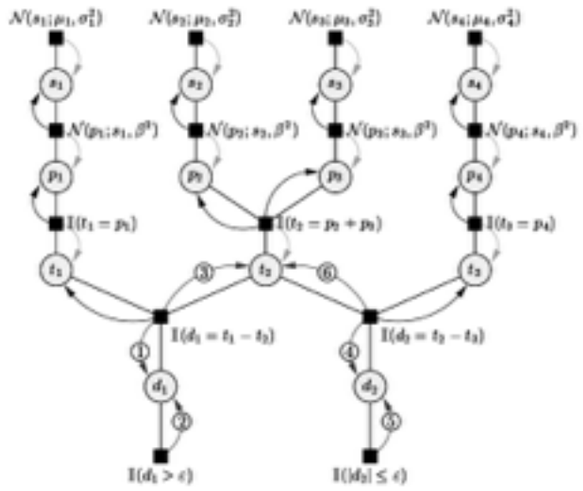
Probabilistic Reasoning



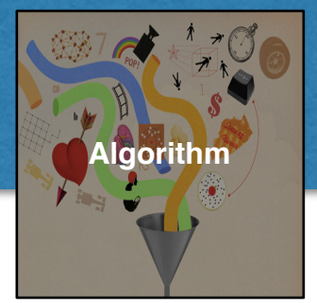
Modern applications and data strongly favour **probabilistic modelling:**

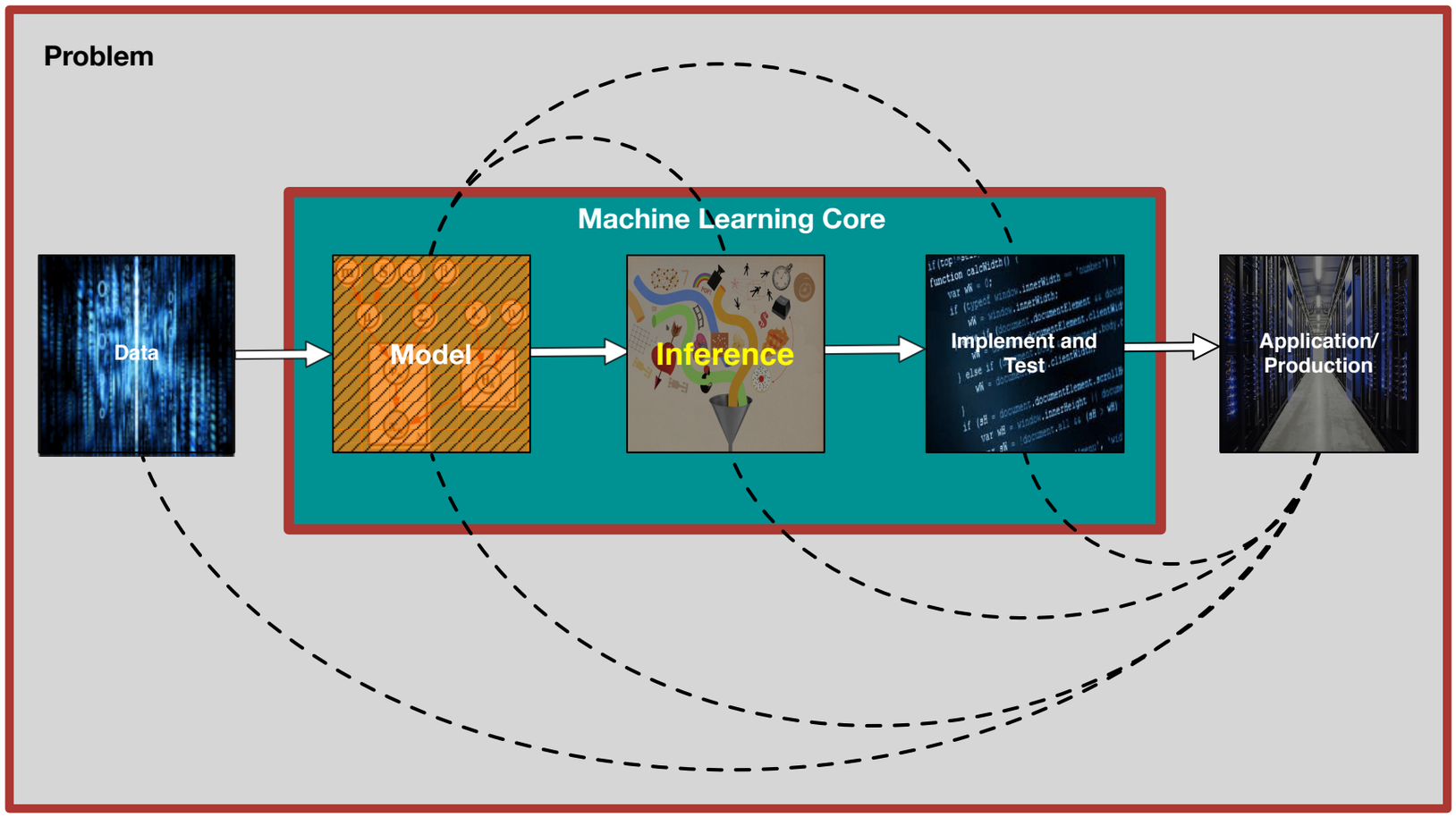
- ***Noise in the data*** and account for our lack of knowledge
- ***Non-iid, non-stationary*** data.
- Explore and extract the ***underlying structure*** in the data
- ***Consistency in our beliefs*** about the data and systems we study.

Trueskill: large scale probbabilistic models using factor graphs



Probabilistic Inference





In probabilistic models, we must reason over the probability of events.

Statistical Inference

Any mechanism by which we deduce the probabilities in our model based on data.

Inference links the observed data with our statistical assumptions and allows us to ask questions of our data: predictions, visualisation, model selection.

Outline

- Probabilistic Modelling and Inference
- Variational Inference

- Approximate Posteriors
- Variational Optimisation
- Gradient Computation
- Implementation

Part I:

Probabilistic modelling and the variational principle

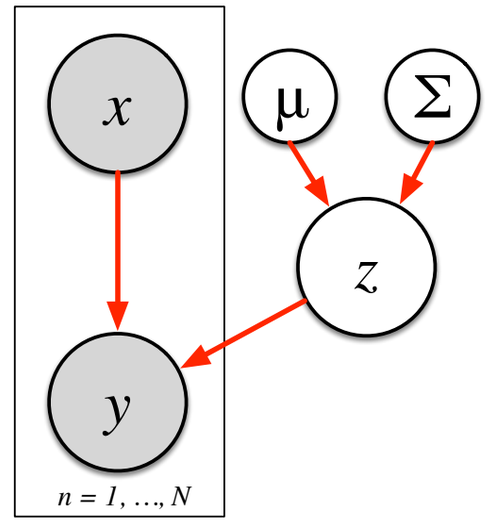
Part II:

Design and implementation of variational algorithms

Modelling and Inference

Probabilistic modelling will involve:

- Decide on a priori beliefs.
- Posit an explanation of how the observed data is generated, i.e. provide a probabilistic description.



Regression: Linear combination of inputs to give response.

Bayes' rule highlights many of the inferential problems we will face.

$$\underbrace{p(z|y)}_{\text{Posterior}} = \frac{\overbrace{p(y|z)}^{\text{Likelihood}}\ \overbrace{p(z)}^{\text{Prior}}}{\underbrace{\int p(y,z)dz}_{\text{Marginal likelihood/ Model evidence}}}$$

Inferential problems

$$\underbrace{p(z|y)}_{\text{Posterior}} = \frac{\overbrace{p(y|z)}^{\text{Likelihood}}\ \overbrace{p(z)}^{\text{Prior}}}{\underbrace{\int p(y,z)dz}_{\text{Marginal likelihood/ Model evidence}}}$$

Most inference problems will be one of:

Marginalisation $$p(y) = \int p(y,\theta)d\theta$$

Expectation $$\mathbb{E}[f(y)|x] = \int f(y)p(y|x)dy$$

Prediction $$p(y_{t+1}) = \int p(y_{t+1}|y_t)p(y_t)dy_t$$

Different Communities

Statistics, no distinction between learning and inference - only inference (or **estimation**).

Machine learning makes a distinction between **inference and learning:**

- *Inference*: reason about (and compute) unknown probability distributions.
- (Parameter) *Learning* is finding point estimates of quantities in the model.

Bayesian statistics, all quantities are probability distributions, so there is only the problem of **inference.**

Software engineering, **inference** is the forward evaluation of a trained model (to get predictions).

Decision making and *AI*, refer to **learning** in general as the means of understanding and acting based on past experience (data).

A Smorgasbord of Inference Methods

For a given model, there are many competing inference methods.

Exact methods (conjugacy, enumeration)

Numerical integration (Quadrature)

Generalised method of moments

Maximum likelihood (ML)

Maximum a posteriori (MAP)

Laplace approximation

Integrated nested Laplace approximations (INLA)

Monte Carlo methods (MCMC, SMC, ABC)

Cavity Methods (EP)

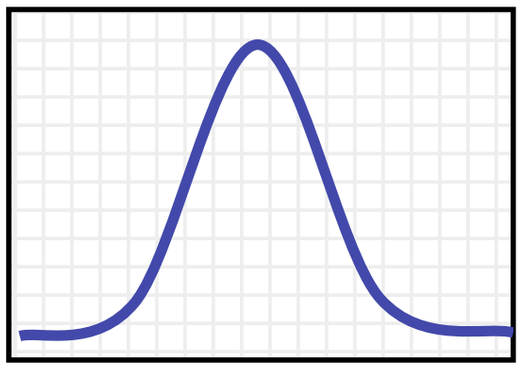
Variational methods

Model Case Study

Latent Gaussian Models

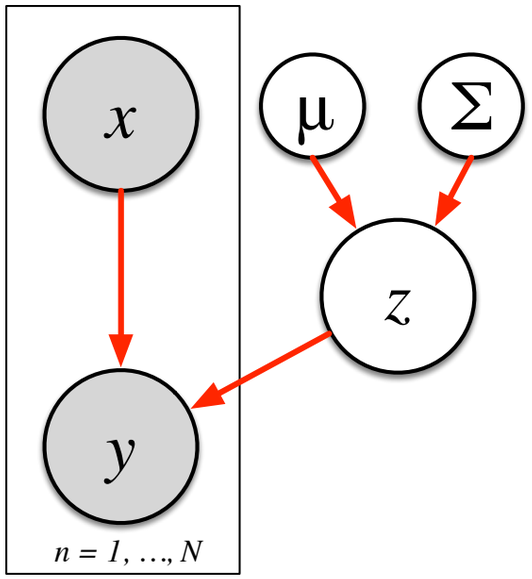
General class of models that is widely used throughout machine learning and statistics.

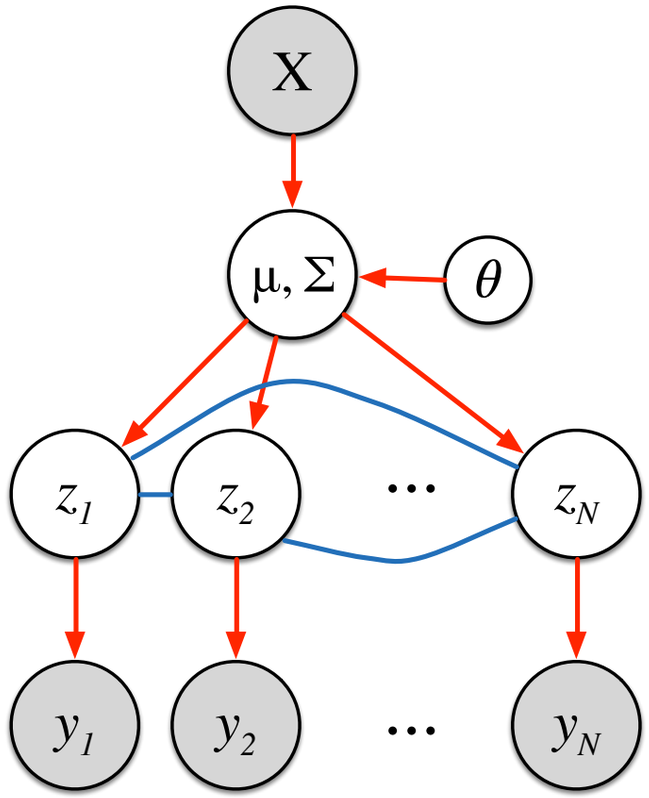
Models with Gaussian latent variables.



Latent Gaussian Models

Regression and Classification





Generalised Linear Models

$$z \sim \mathcal{N}(z|\mu, \Sigma)$$

$$y \sim \mathcal{N}(y|z^\top x, \sigma_y^2)$$

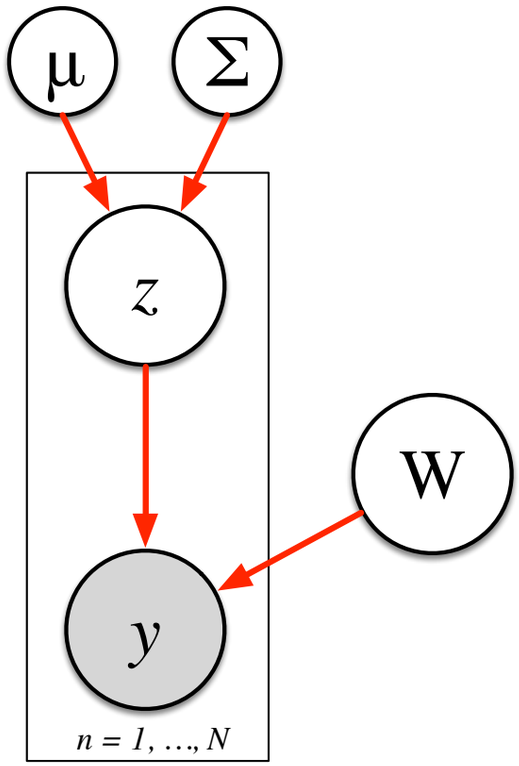
Gaussian process regression

$$z \sim \mathcal{N}(\mu(X), \Sigma(X, X))$$

$$y \sim \mathcal{N}(y|z, \sigma_y^2)$$

Latent Gaussian Models

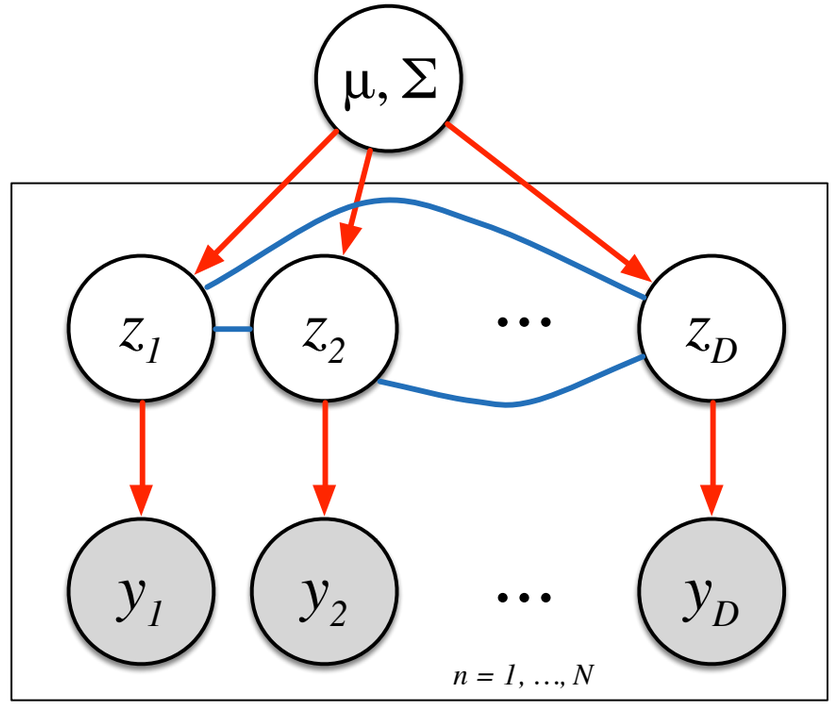
Density Estimation



Factor Analysis / PCA

$$z \sim \mathcal{N}(z|\mu, \Sigma)$$

$$y \sim \mathcal{N}(y|Wz, \sigma_y^2 I)$$



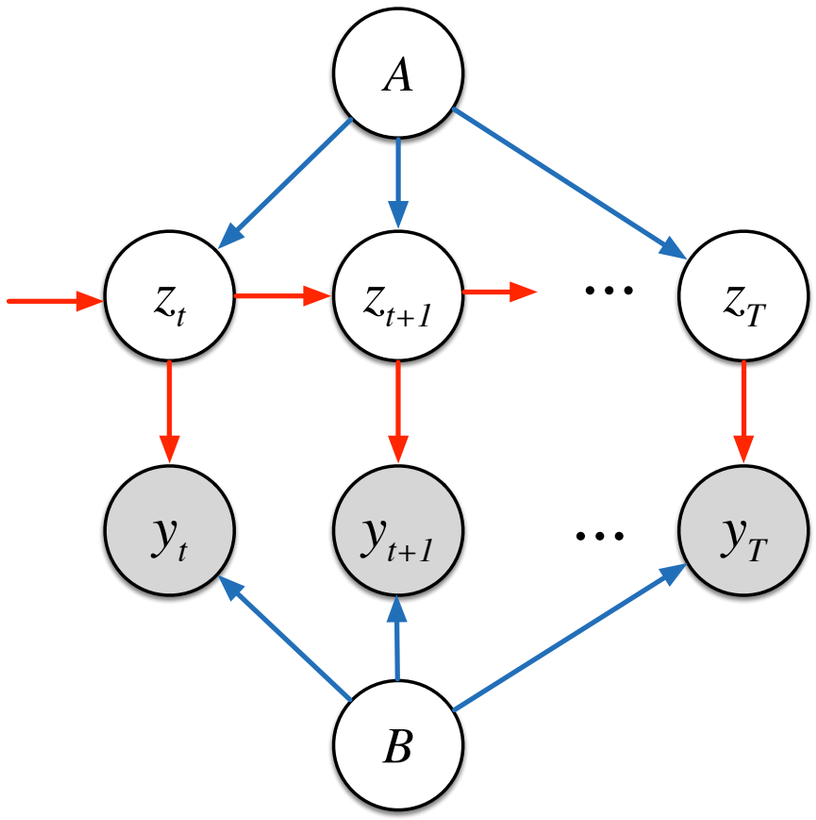
Latent Gaussian Graphical Models

$$z \sim \mathcal{N}(z|\mu, \Sigma)$$

$$y \sim \mathcal{N}(y|z, \sigma_y^2)$$

Latent Gaussian Models

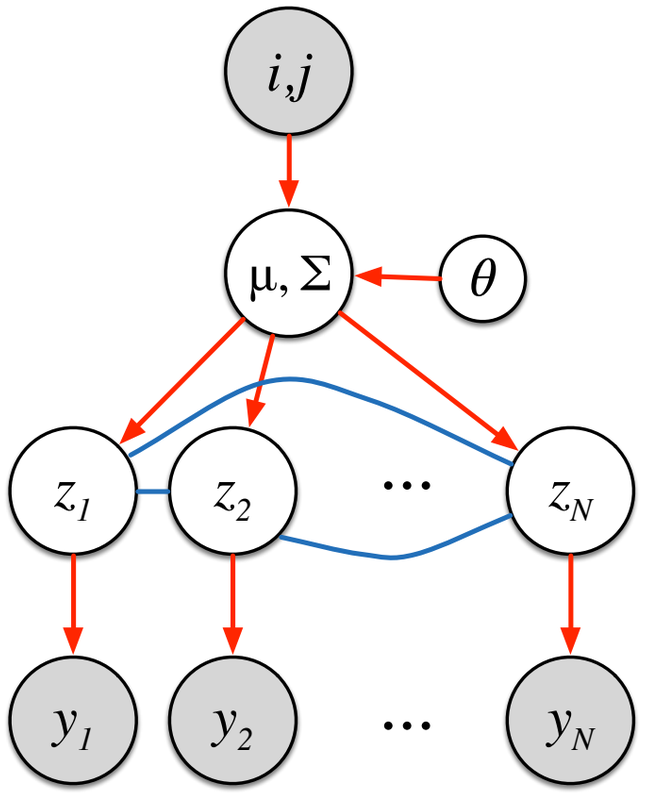
Temporal and Spatial Models



Gaussian Linear State Space Model
Kalman Filter

$$z_t \sim \mathcal{N}(z_t | A z_{t-1}, \sigma_z^2 I)$$

$$y_t \sim \mathcal{N}(y_t | B z_t, \sigma_y^2 I)$$



Latent Gaussian Cox Point Process

$$x \sim \mathcal{N}(x | \mu(i,j), \Sigma(i,j))$$

$$y_{ij} \sim \mathcal{P}(c \exp(x_{ij}))$$

Exponential Family Factor Models

Look at two specific instances of this model class:

- Bayesian exponential family PCA (***BXPCA***)
- Deep Latent Gaussian Models (***DLGM***)

BXPCA

Latent Variable

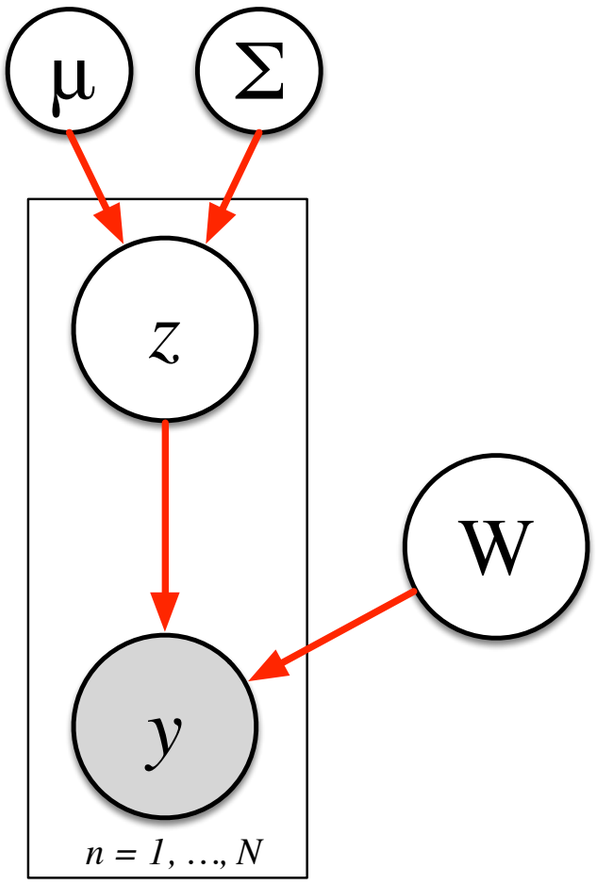
$$\mathbf{z} \sim \mathcal{N}(\mathbf{z}|\mu, \boldsymbol{\Sigma})$$

Observation Model

$$\boldsymbol{\eta} = \mathbf{W}\mathbf{z} + \mathbf{b}$$

$$\mathbf{y} \sim \text{Expon}(\mathbf{y}|\boldsymbol{\eta})$$

Exponential family with
natural parameters η.



Exponential Family Factor Models

Rich extension of previous model using deep neural networks: Deep Latent Gaussian Model (DLGM).

DLGM

Latent Variables (Stochastic layers)

$$\mathbf{z}_l \sim \mathcal{N}(\mathbf{z}_l | f_l(\mathbf{z}_{l+1}), \boldsymbol{\Sigma}_l)$$

$$f_l(\mathbf{z}) = \sigma(\mathbf{W} h(\mathbf{z}) + \mathbf{b})$$

Deterministic layers

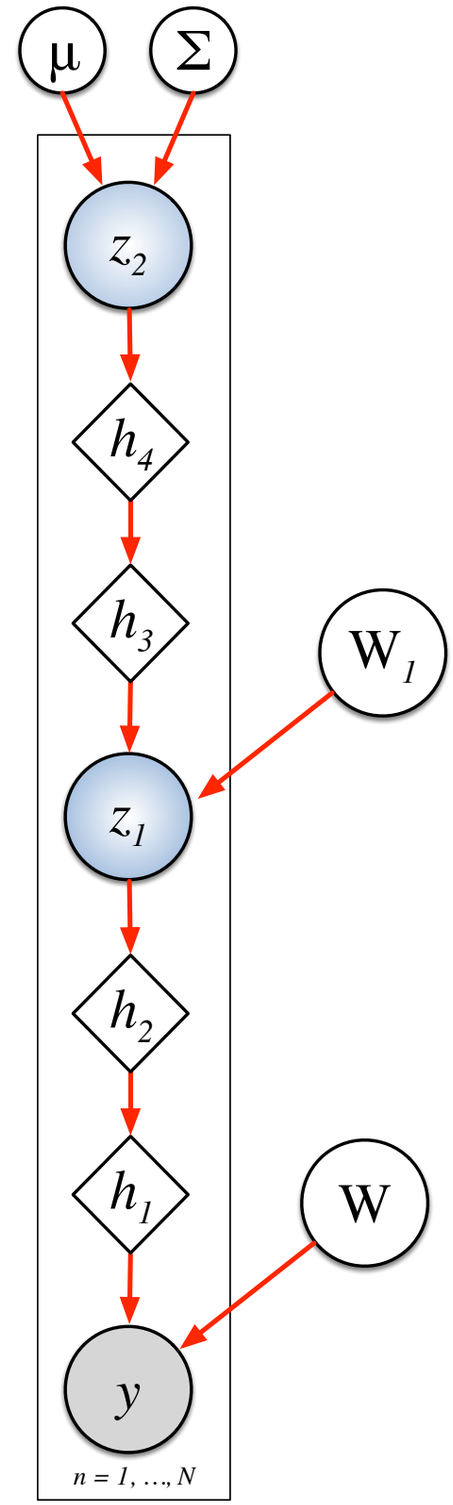
$$h_i(\mathbf{x}) = \sigma(\mathbf{A}\mathbf{x} + \mathbf{c})$$

Observation Model

$$\boldsymbol{\eta} = \mathbf{W}\mathbf{h_1} + \mathbf{b}$$

$$\mathbf{y} \sim \text{Expon}(\mathbf{y} | \boldsymbol{\eta})$$

Can also use non-exponential family.



Exponential Family Factor Models

Our inferential tasks are:

1. Explain this data

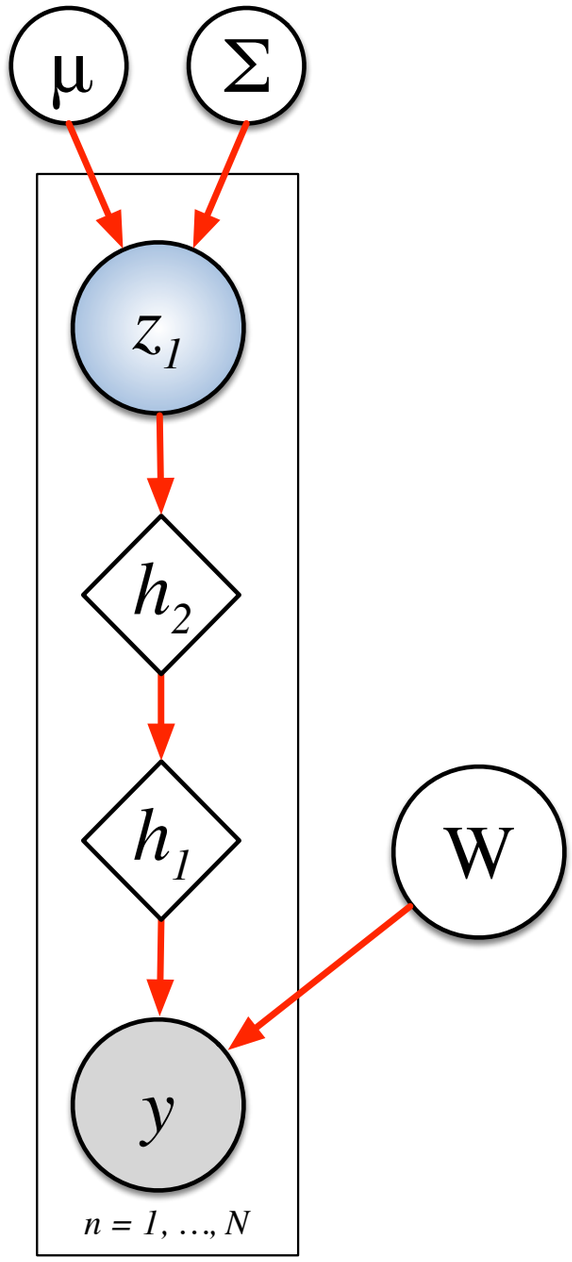
$$p(\mathbf{z}|\mathbf{y}, \mathbf{W}) \propto p(\mathbf{y}|\mathbf{z}, \mathbf{W})p(\mathbf{z})$$

2. Make predictions:

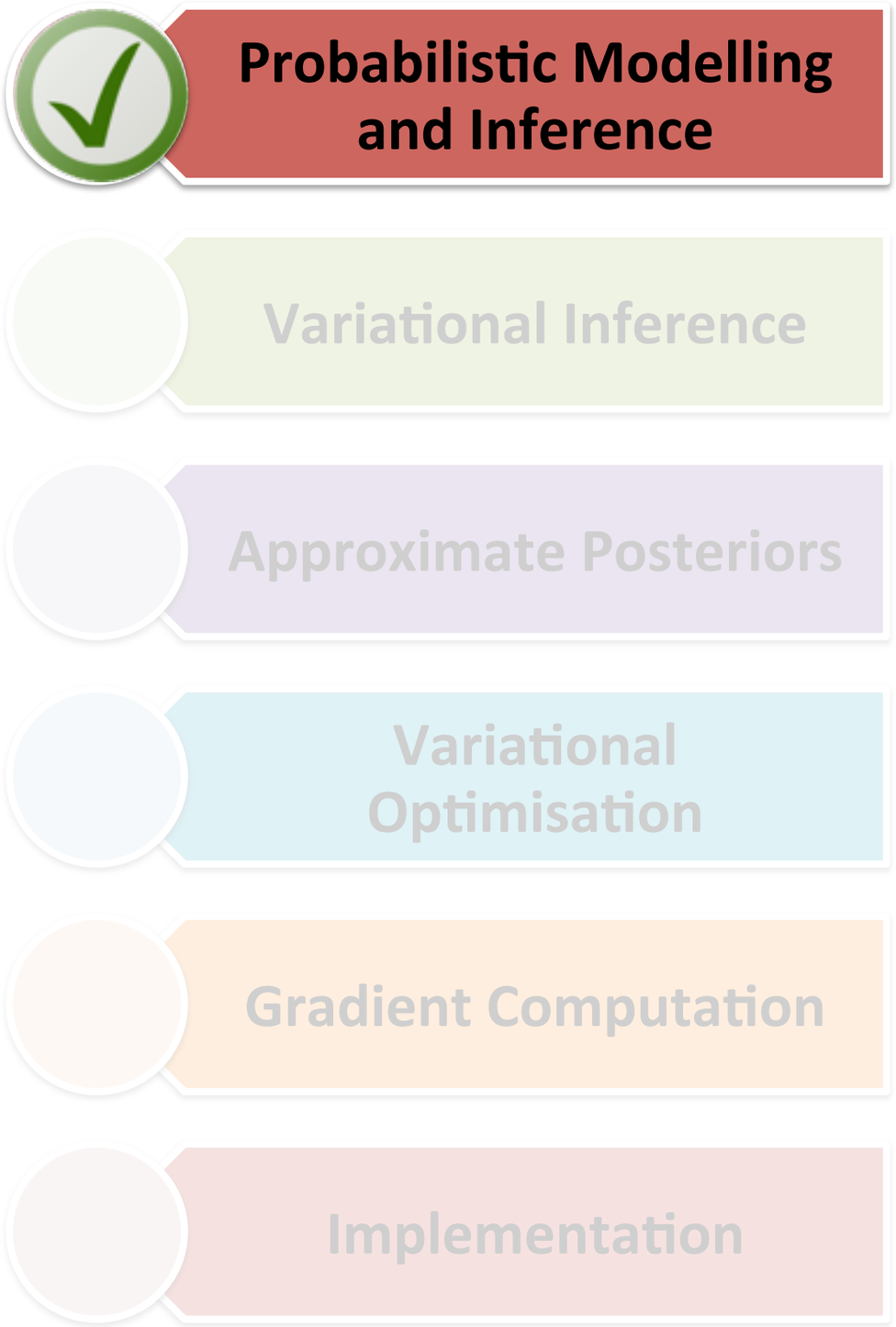
$$p(\mathbf{y}^*|\mathbf{y}) = \int p(\mathbf{y}^*|\mathbf{z}, \mathbf{W})p(\mathbf{z}|\mathbf{y}, \mathbf{W})d\mathbf{z}$$

3. Choose the best model

$$p(\mathbf{y}|\mathbf{W}) = \int p(\mathbf{y}|\mathbf{z}, \mathbf{W})p(\mathbf{z})d\mathbf{z}$$



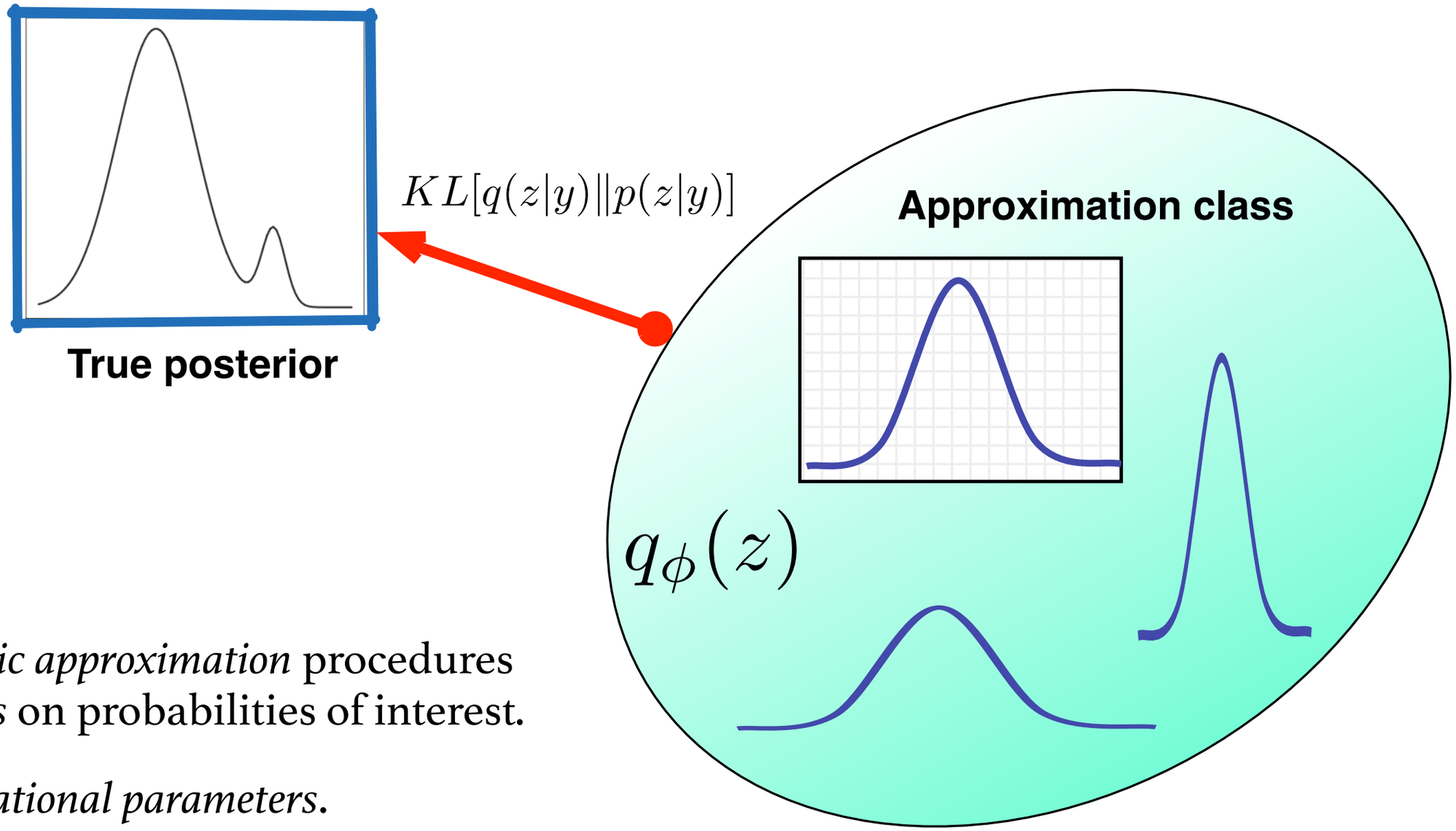
Progress ...


Probabilistic Modelling and Inference
Variational Inference
Approximate Posteriors
Variational Optimisation
Gradient Computation
Implementation

What is a Variational Method?

Variational Principle

General family of methods for approximating complicated densities by a simpler class of densities.



D*eterministic approximation* procedures with *bounds* on probabilities of interest.

Fit the *variational parameters*.

Variational Calculus

Called a variational method because it derives from the **Calculus of Variations**.

Functions:

- Variables as input, output is a value.
- Full and partial derivatives $\frac{df}{dx}$
- E.g., Maximise likelihood p(x|θ) w.r.t. parameters θ

Functionals:

- Functions as input, output is a value.
- Functional derivatives $\frac{\delta F}{\delta f}$
- E.g., Maximise the entropy H[*p(x)*] w.r.t. *p(x)*

We exploit both types of derivatives in variational inference.

Variational Calculus

Two basic rules

- **Functional derivative:**

$$\frac{\delta f(x)}{\delta f(x')} = \delta(x - x')$$

- **Commutative rule:**

$$\frac{\delta}{\delta f(x')} \frac{\partial f(x)}{\partial x} = \frac{\partial}{\partial x} \frac{\delta f(x)}{\delta f(x')}$$

Simple example: Maximise the entropy w.r.t. *p(x)*

$$H[p(x)] = -\int p(x) \log p(x) dx$$

Compute: $\frac{\delta H[p(x)]}{\delta p(x)}$

$$-\frac{\delta}{\delta p(x)} \int p(x) \log p(x) dx$$

$$-\int p(x) \frac{1}{p(x)} \delta(x - x') dx' - \int \log p(x) \delta(x - x') dx'$$

$$-1 - \log p(x)$$

Inferential problems

$$\underbrace{p(z|y)}_{\text{Posterior}} = \frac{\overbrace{p(y|z)}^{\text{Likelihood}}\ \overbrace{p(z)}^{\text{Prior}}}{\underbrace{\int p(y,z)dz}_{\text{Marginal likelihood/ Model evidence}}}$$

Most inference problems will be one of:

Marginalisation $$p(y) = \int p(y,\theta)d\theta$$

Expectation $$\mathbb{E}[f(y)|x] = \int f(y)p(y|x)dy$$

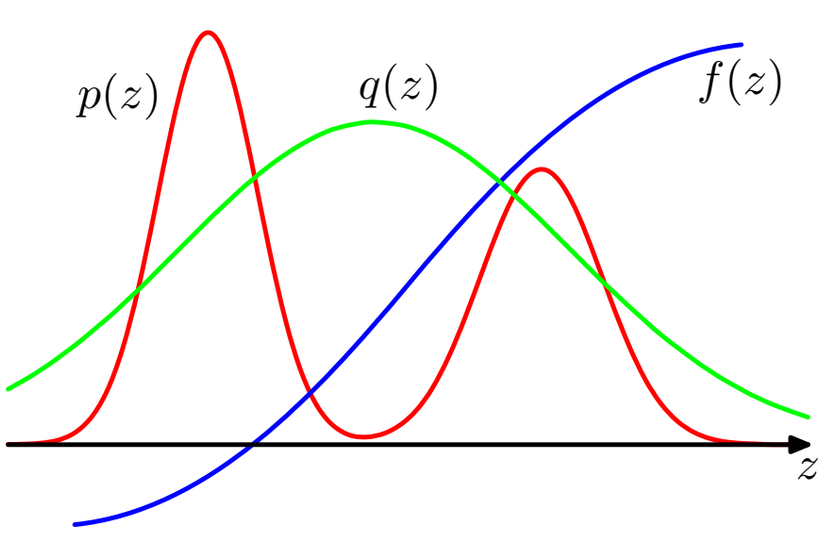
Prediction $$p(y_{t+1}) = \int p(y_{t+1}|y_t)p(y_t)dy_t$$

Importance Sampling

Basic idea:
Transform the integral into an expectation over a simple, known distribution.

Integral problem $$p(y) = \int p(y|z)p(z)dz$$

Proposal $$p(y) = \int p(y|z)p(z)\frac{q(z)}{q(z)}dz$$



Importance Weight $$p(y) = \int p(y|z)\frac{p(z)}{q(z)}q(z)dz$$

$$w^{(s)} = \frac{p(z)}{q(z)} \quad z^{(s)} \sim q(z)$$

Conditions

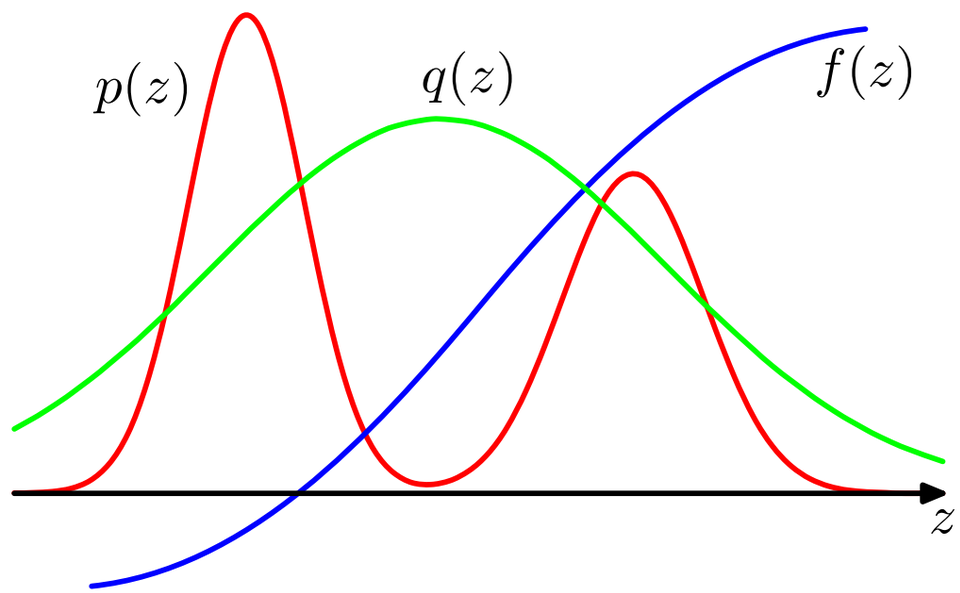
- $q(z)>0$, when $f(z)p(z) \neq 0$.
- Easy to sample from $q(z)$.

Monte Carlo $$p(y) = \frac{1}{S}\sum_s w^{(s)}p(y|z^{(s)})$$

Importance Sampling

$$p(x) = \frac{1}{S} \sum_s w^{(s)} p(y|z^{(s)})$$

$$w^{(s)} = \frac{p(z)}{q(z)} \quad z^{(s)} \sim q(z)$$



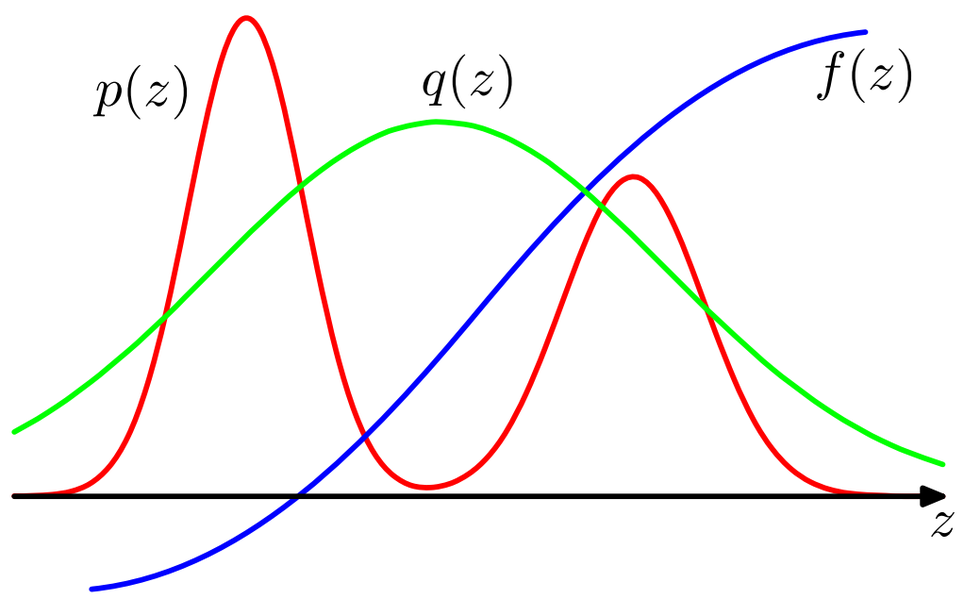
Properties:

- ***Unbiased*** estimate of the expectation.
- ***No independent samples*** from the posterior distribution.
- **Many draws** from proposal needed, especially in high dimensions.

Importance Sampling

$$p(x) = \frac{1}{S}\sum_{s} w^{(s)} p(y|z^{(s)})$$

$$w^{(s)} = \frac{p(z)}{q(z)} \quad z^{(s)} \sim q(z)$$

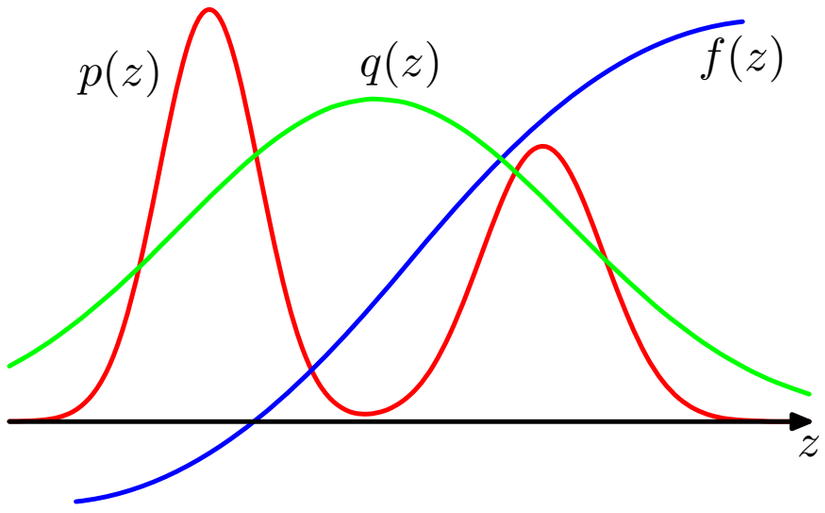


Can we take inspiration from importance sampling, but instead:

- Obtain a ***deterministic*** algorithm,
- Scaled up to ***high-dimensional and large data*** problems,
- Easy **convergence assessment**.

Now, from importance sampling to variational inference ...

Importance Sampling



Instead of Monte Carlo integration, can we manipulate the integral using a different technique?

Integral problem $$p(y) = \int p(y|z)p(z)dz$$

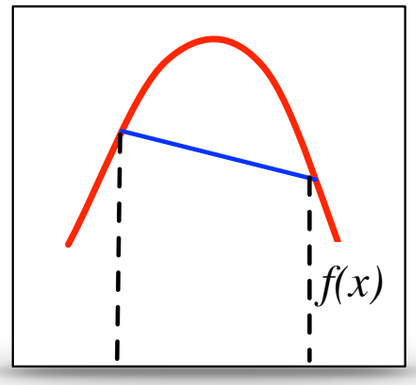
Proposal $$p(y) = \int p(y|z)p(z)\frac{q(z)}{q(z)}dz$$

Importance Weight $$p(y) = \int p(y|z)\frac{p(z)}{q(z)}q(z)dz$$

$$w^{(s)} = \frac{p(z)}{q(z)} \quad z^{(s)} \sim q(z)$$

Monte Carlo $$p(y) = \frac{1}{S}\sum_s w^{(s)}p(y|z^{(s)})$$

Jensen's Inequality



An important result from convex analysis:

For concave functions *f(.)*

$$f(\mathbb{E}[x]) \geq \mathbb{E}[f(x)]$$

Logarithms are strictly *concave* allowing us to use Jensen's inequality.

$$\log \int p(x)g(x)dx \geq \int p(x) \log g(x)dx$$

Instead of Monte Carlo Integration, use Jensen's inequality.

From IS to Variational Inference

Integral problem

$$\log p(y) = \log \int p(y|z)p(z)dz$$

Proposal

$$\log p(y) = \log \int p(y|z)p(z)\frac{q(z)}{q(z)}dz$$

Importance Weight

$$\log p(y) = \log \int p(y|z)\frac{p(z)}{q(z)}q(z)dz$$

Jensen's inequality

$$\log \int p(x)g(x)dx \geq \int p(x) \log g(x)dx$$

$$\log p(y) \geq \int q(z) \log \left(p(y|z)\frac{p(z)}{q(z)} \right) dz$$

$$= \int q(z) \log p(y|z) - \int q(z) \log \frac{q(z)}{p(z)}$$

Variational lower bound

$$= \mathbb{E}_{q(z)}[\log p(y|z)] - KL[q(z)\|p(z)]$$

Variational Inference

$$\mathcal{F}(y, q) = \mathbb{E}_{q(z)}[\log p(y|z)] - KL[q(z)\|p(z)]$$

This bound is exactly of the form we are looking for.

- **Variational free energy:** We obtain a functional and are free to choose the distribution $q(z)$ that best matches the true posterior.

- **Evidence lower bound (ELBO):** principled bound on the marginal likelihood, or model evidence.
- Certain choices of $q(z)$ makes this quantity easier to compute. Examples to come.

Variational Inference

$$\mathcal{F}(y, q) = \mathbb{E}_{q(z)}[\log p(y|z)] - KL[q(z)\|p(z)]$$

Approx. Posterior | Reconstruction | Penalty

Interpreting the bound:

- **Approximate posterior distribution *q(z):*** Best match to true posterior $p(z|y)$, one of the unknown inferential quantities of interest to us.
- **Reconstruction cost**: The expected log-likelihood measure how well samples from $q(z)$ are able to explain the data y.
- **Penalty:** Ensures the the explanation of the data $q(z)$ doesn't deviate too far from your beliefs $p(z)$. A mechanism for realising Okham's razor.

Variational Inference

$$\mathcal{F}(y, q) = \mathbb{E}_{q(z)}[\log p(y|z)] - KL[q(z)||p(z)]$$

Some comments on *q*:

- **Integration is now optimisation**: optimise for *q(z)* directly.
  - I write *q(z)* to simplify the notation, but it depends on the data, *q(z|y)*.
  - *Easy convergence assessment* since we wait until the free energy (loss) reaches convergence.
- **Variational parameters:** parameters of *q(z)*
  - E.g., if a Gaussian, variational parameters are mean and variance.
  - Optimisation allows us to *tighten the bound* and get as close as possible to the true marginal likelihood.

Minimum Description Length (MDL)

$$\mathcal{F}(y, q) = \mathbb{E}_{q(z)}[\log p(y|z)] - KL[q(z)\|p(z)]$$

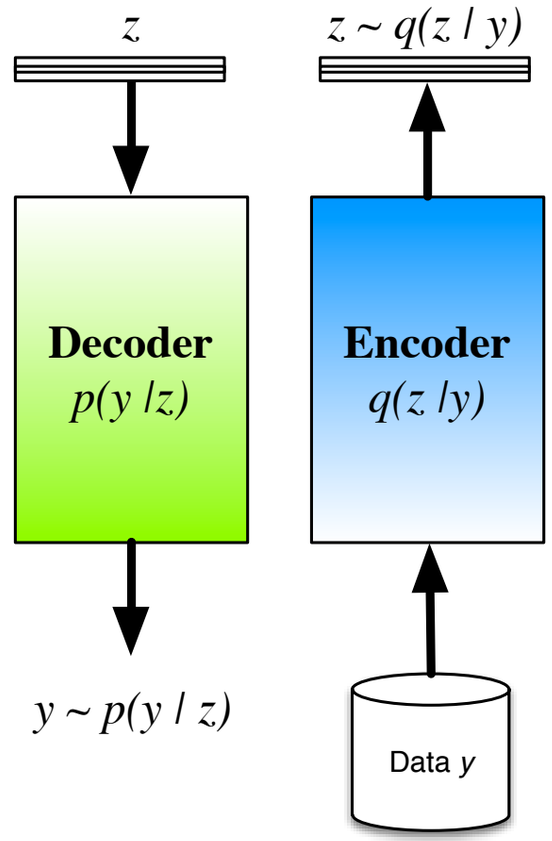
Stochastic encoder

Data code-length

Hypothesis code

Stochastic encoder-decoder systems implement variational inference.

- Regularity in our data that can be explained with latent variables, implies that the data is compressible.
- MDL: inference seen as a problem of compression — we must find the ideal shortest message of our data *y:* marginal likelihood.
- Must introduce an approximation to the ideal message.
- **Encoder:** variational distribution $q(z|y)$,
- **Decoder:** likelihood $p(y|z)$.



Denoising Auto-encoders (DAE)

$$\mathcal{F}(y, q) = \mathbb{E}_{q(z)}[\log p(y|z)] - \Omega(z, y)$$

Stochastic encoder

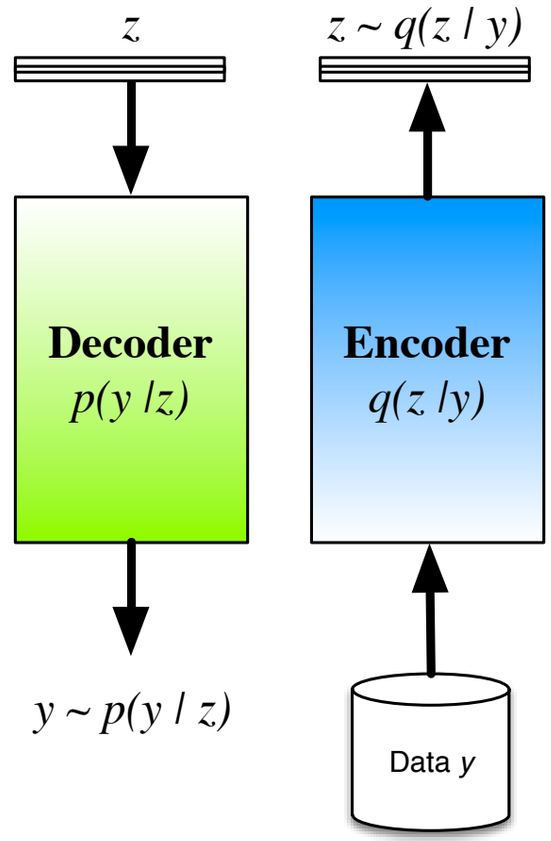
Reconstruction

Penalty

Stochastic encoder-decoder systems implement variational inference.

- DAE: A mechanism for finding representations or features of data (i.e. latent variable explanations).
- **Encoder:** variational distribution *q(z|y)*,
- **Decoder:** likelihood *p(y|z)*.

The variational approach requires you to be explicit about your assumptions. Penalty is derived from your model and does not need to be designed.



Variational Inference vs. Variational Bayes

$$p(y|z,\theta)p(z,\theta)$$

Variational Inference (VI)

Apply the variational principle only to some parts of the model.

Widely-used case: latent variables are assigned probability distributions; maximum likelihood estimates for others.

$$q(z); \quad \theta_{ML}$$

Inference *Learning*

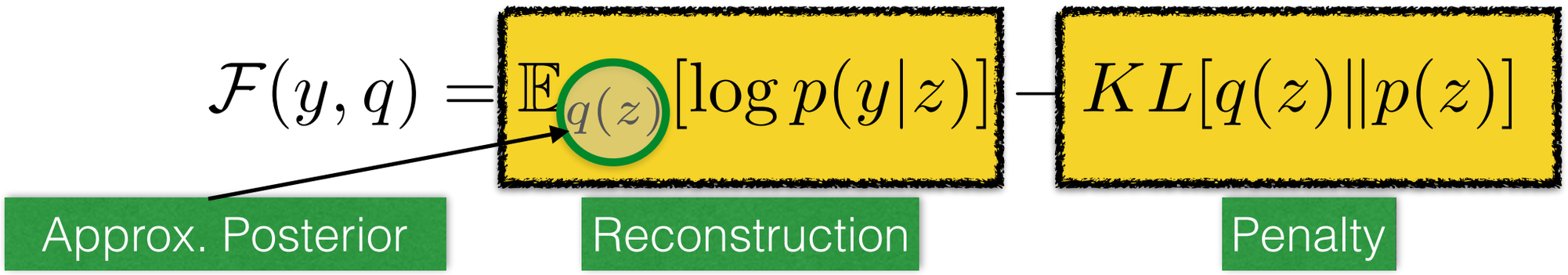
Variational Bayesian Inference (VB)

All unknown quantities are probability distributions and use a variational approximation for all posterior distributions.

$$q(z,\theta|y)$$

Inference

Designing Variational Algorithms



1. Choice of the variational distribution *q(z)*:

VI or VB? Specification of q, what structure does it have?

2. Computation of expectations and gradients:

Expectation might be difficult to compute in general. How to efficiently compute it.

> Rest of the tutorial, we'll discuss these two options and how to implement them.

Why Variational Inference?

Disadvantages:

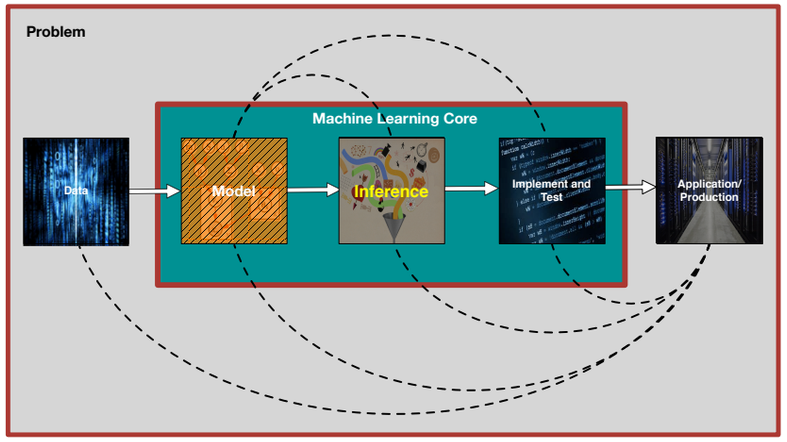
- An **approximate posterior** only - not always guaranteed to find exact posterior in the limit.
- **Difficulty in optimisation** — can get stuck in local minima.
- Typically **under-estimates the variance** of the posterior and can bias maximum likelihood parameter estimates.
- **Limited theory** and guarantees for variational methods.

Why Variational Inference?

Advantages:

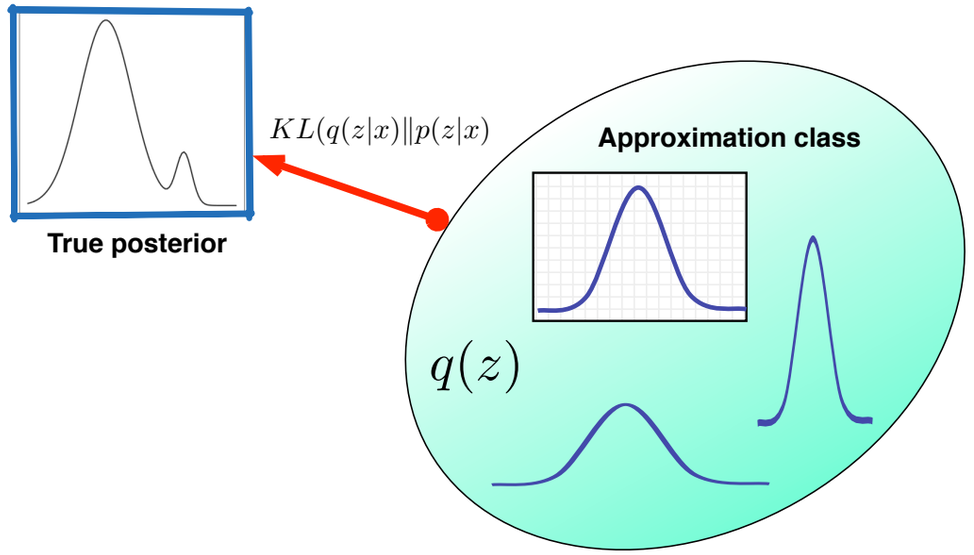
- Applicable to almost **all probabilistic models:** non-linear, non-conjugate, high-dimensional, directed and undirected.
- Transforms problem of **integration into one of optimisation**.
- Easy **convergence assessment.**
- Principled and scalable approach for **model selection**.
- **Compact representation** of the posterior distribution.
- Can be **faster to converge** than competing methods.
- **Numerically stable.**
- Can be used on **modern computing architectures** (CPUs and GPUs)

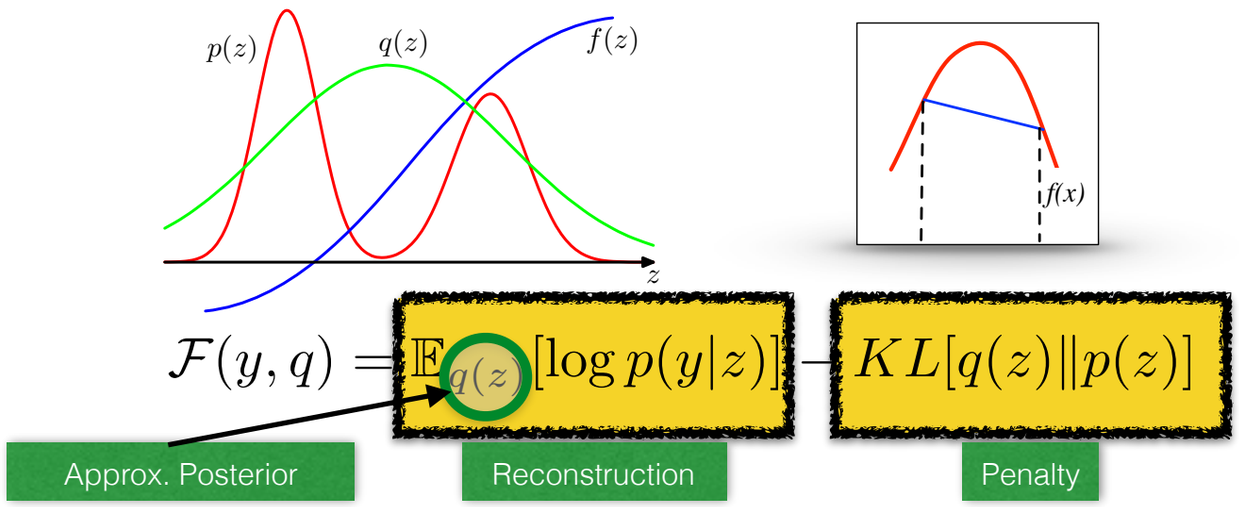
In Review ...



Explored the **central role of statistical inference** in Machine Learning and data science.

Looked at the **variational approach** as one powerful and compelling method for inference.





Moved from importance sampling to variational inference by applying the variational principle giving us the **variational lower bound.**

Progress ...

Probabilistic Modelling and Inference

Variational Inference

Approximate Posteriors

Variational Optimisation

Gradient Computation

Implementation

End of Part I:

Probabilistic modelling and the variational principle

Next:

Design and implementation of variational algorithms

Progress ...

- ✓ Probabilistic Modelling and Inference
- ✓ Variational Inference
- Approximate Posteriors
- Variational Optimisation
- Gradient Computation
- Implementation

Part I:

Probabilistic modelling and the variational principle

Now:

Design and implementation of variational algorithms

Thus far ...

$$\max_{q,\theta} \mathcal{F}(y, q) = \mathbb{E}_{q(z)}[\log p(y|z, \theta)] - KL[q(z)\|p(z)]$$

Approx. Posterior

Reconstruction

Penalty

- *What exactly is q(z)?*
- *How do we find the variational parameters?*
- *How do we optimise the model parameters?*
- *How do we compute the gradients?*

Free-form and Fixed-form

Free-form variational method solves for the exact distribution setting the functional derivative to zero.

$$\frac{\delta \mathcal{F}(y, q)}{\delta q(z)} = 0 \quad s.t. \int q(z) dz = 1$$

$$q(z) \propto p(z) \exp(\log p(y|z, \theta))$$

Great! The optimal solution is the true posterior distribution.

But solving for the normalisation is our original problem.

Fixed-form variational method specifies an explicit form of the q-disribution.

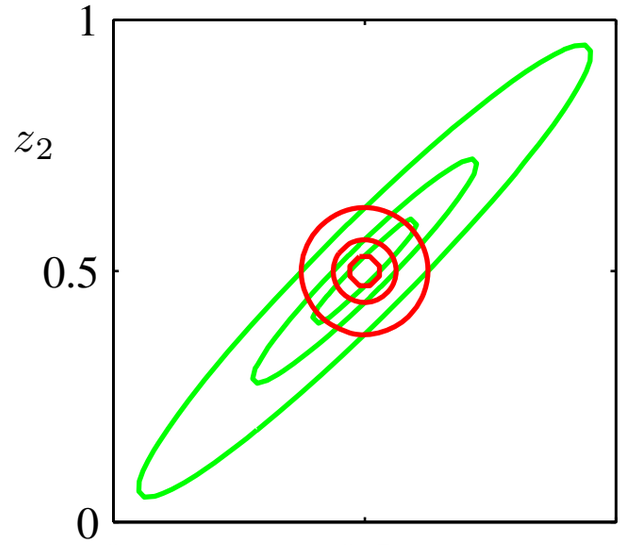
$$q_\phi(z) = f(z; \phi)$$

This is ideally a rich class of distributions. Parameters ϕ are called variational parameters.

Mean-field Variational Inference

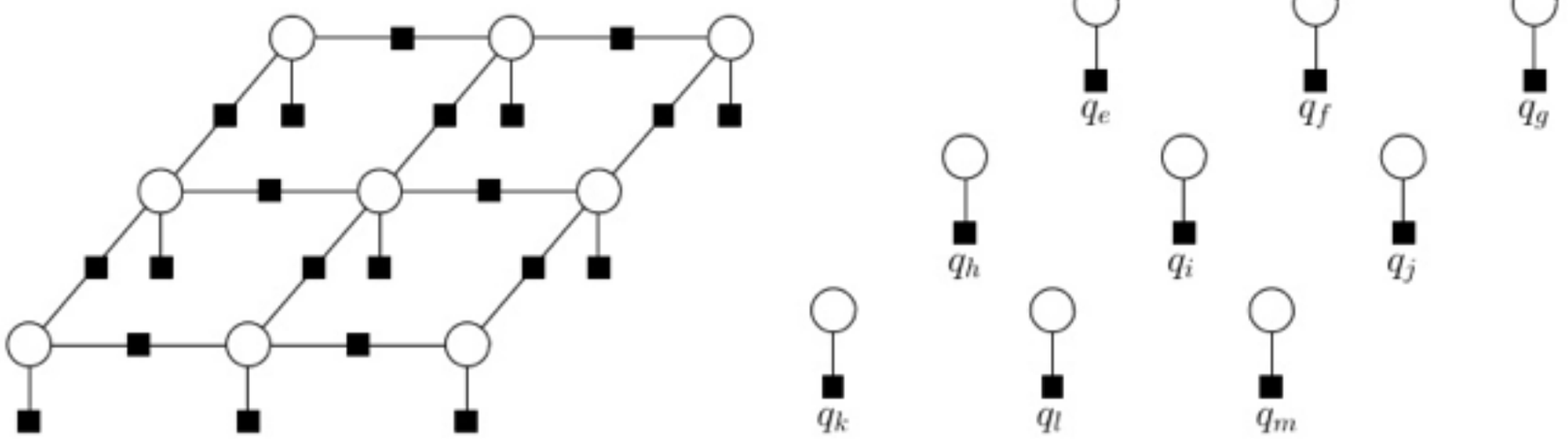
Mean-field methods assume that the distribution is factorised.

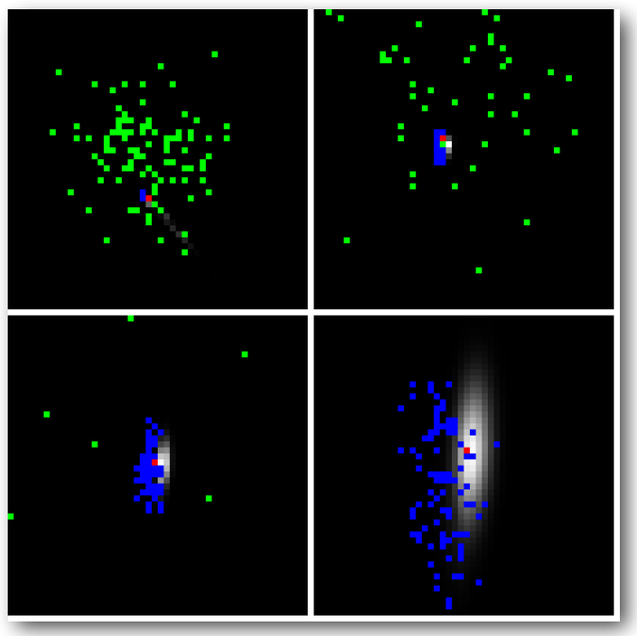
$$q(z) = \prod_i q_i(z_i)$$



Restricted class of approximations: every dimension (or subset of dimensions) of the posterior is independent.

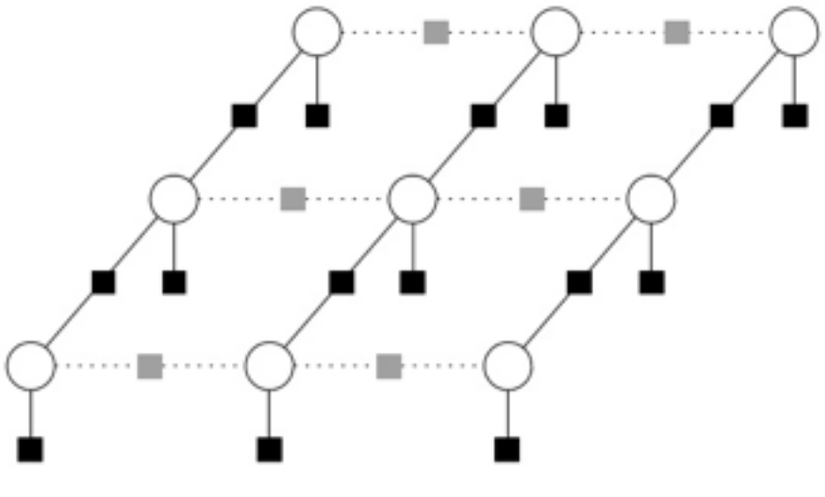
$$q(z) = \prod_i \mathcal{N}(z_i|\mu_i, \sigma_i^2)$$



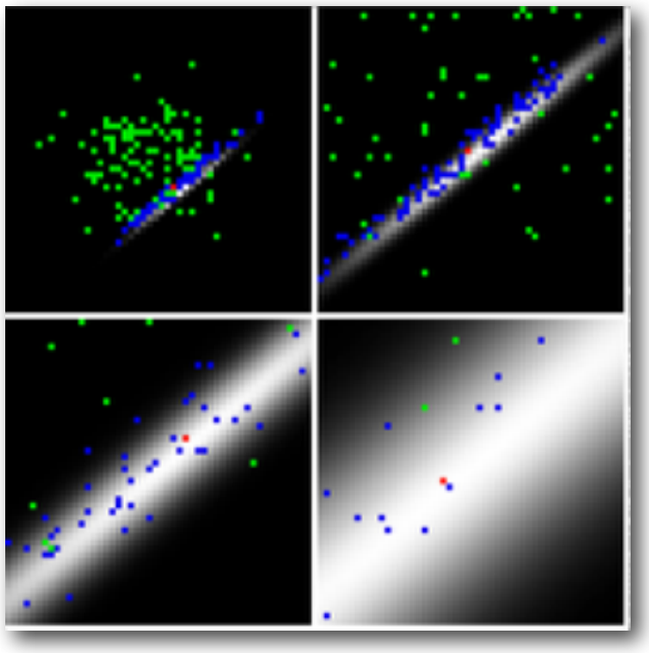


Structured Mean-field

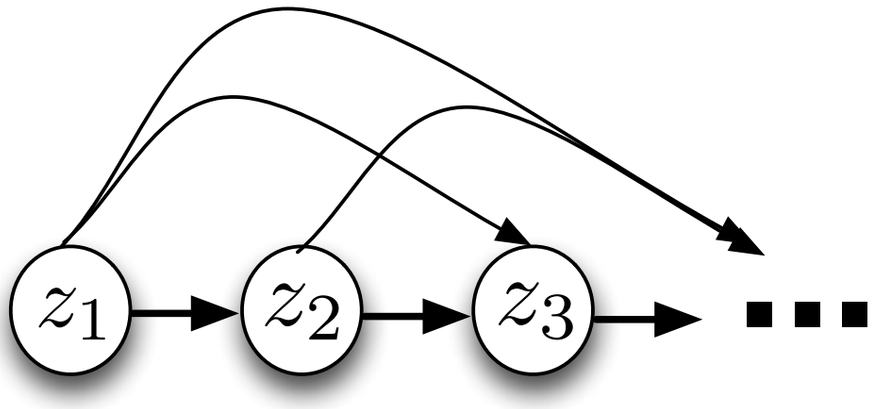
Structured mean-field: introduce dependencies into our factorisation.



$$q(\mathbf{z}) = \prod_i q_i(z_i | \{z_j\}_{j \neq i})$$



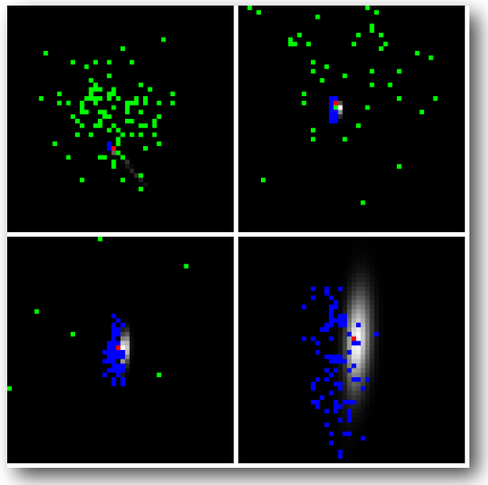
Autoregressive approximation: One very useful and powerful structured specification is to condition on all previous variables.

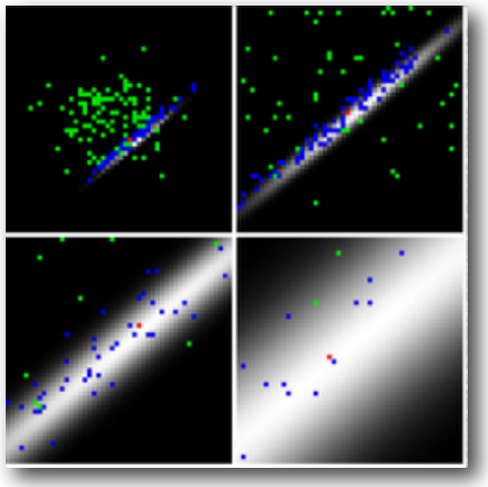


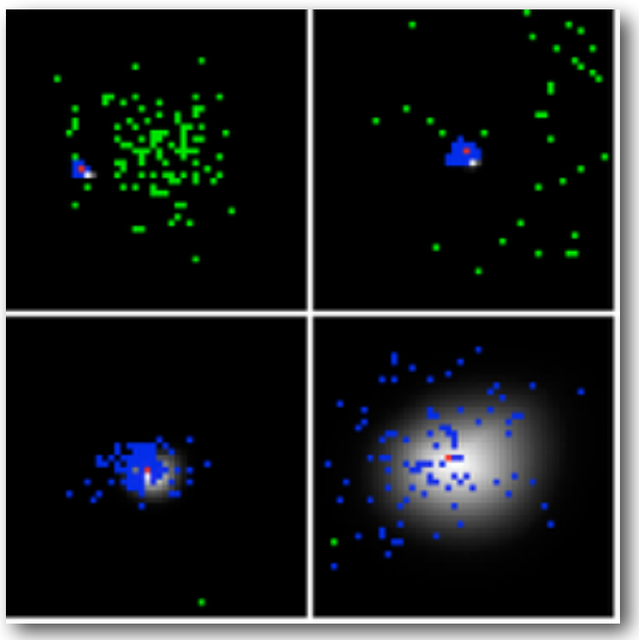
$$q(\mathbf{z}) = \prod_i q_i(z_i | z_{<i})$$

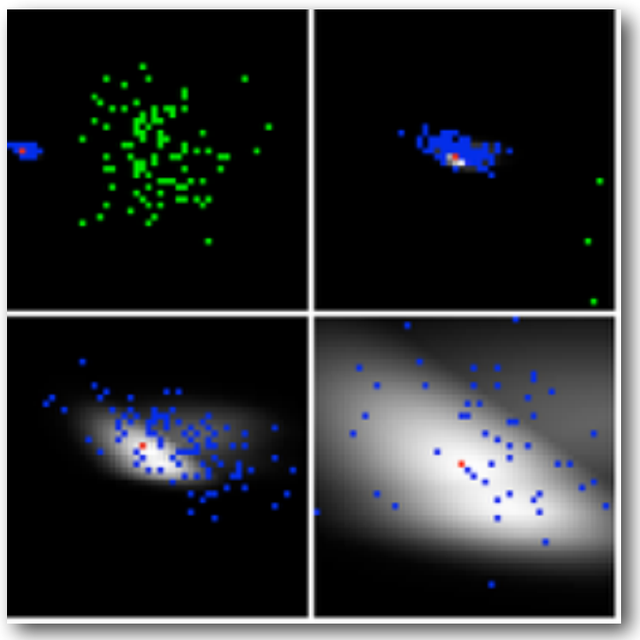
Fixed-form approximations

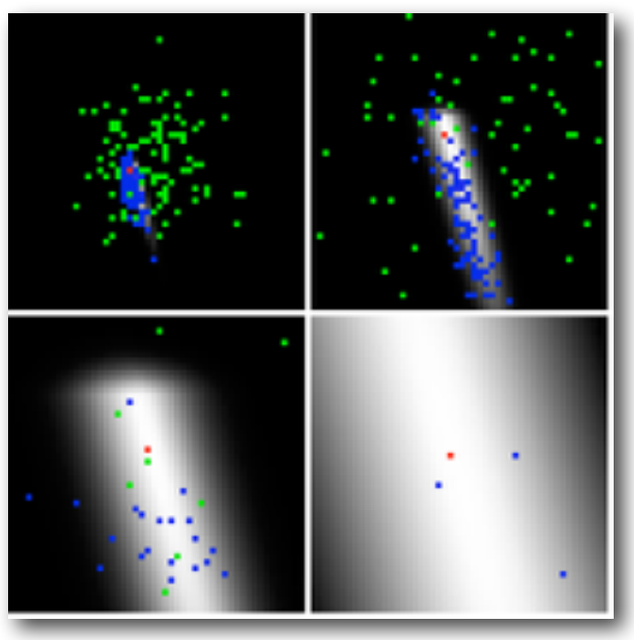
Require flexible approximations for the types of posteriors we are likely to see.











Variational Latent Gaussian Models

Examples: GP regression, BXPCA or DLGM.

$$z \sim \mathcal{N}(z|0,1) \qquad y \sim p(y|f_\theta(z)) \qquad q(z) = \prod_i \mathcal{N}(z_i|\mu_i, \sigma_i^2)$$

$$\mathcal{F}(y,q) = \mathbb{E}_{q(z)}[\log p(y|z)] - KL[q(z)\|p(z)]$$

$$\mathcal{F}(y,q) = \mathbb{E}_{q(z)}[\log p(y|z)] - \sum_i KL[q(z_i)\|p(z_i)]$$

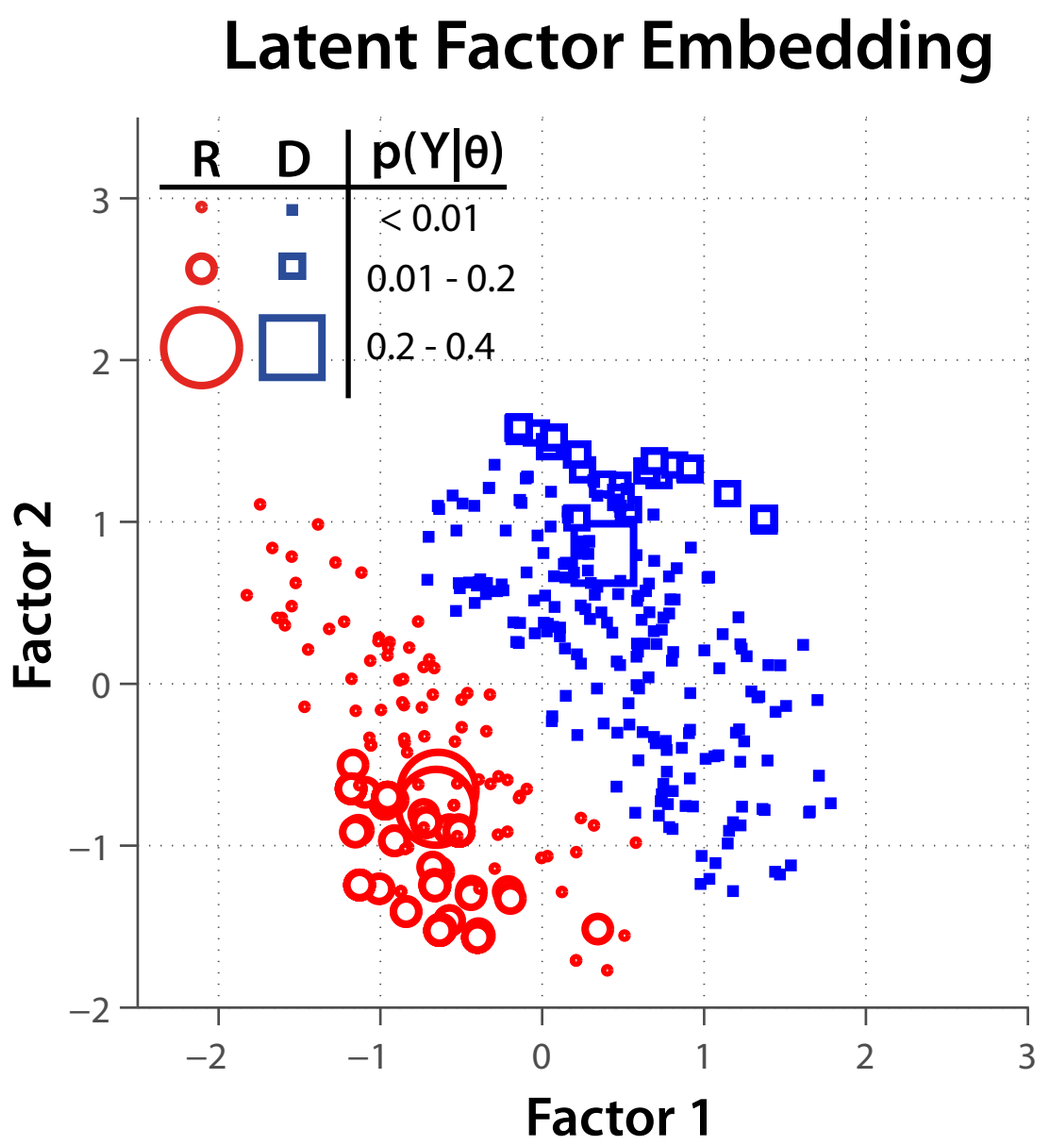
$$\mathcal{F}(y,q) = \mathbb{E}_{q(z)}[\log p(y|z)] - \sum_i KL[\mathcal{N}(z_i|\mu_i, \sigma_i^2)\|\mathcal{N}(z_i|0,1)]$$

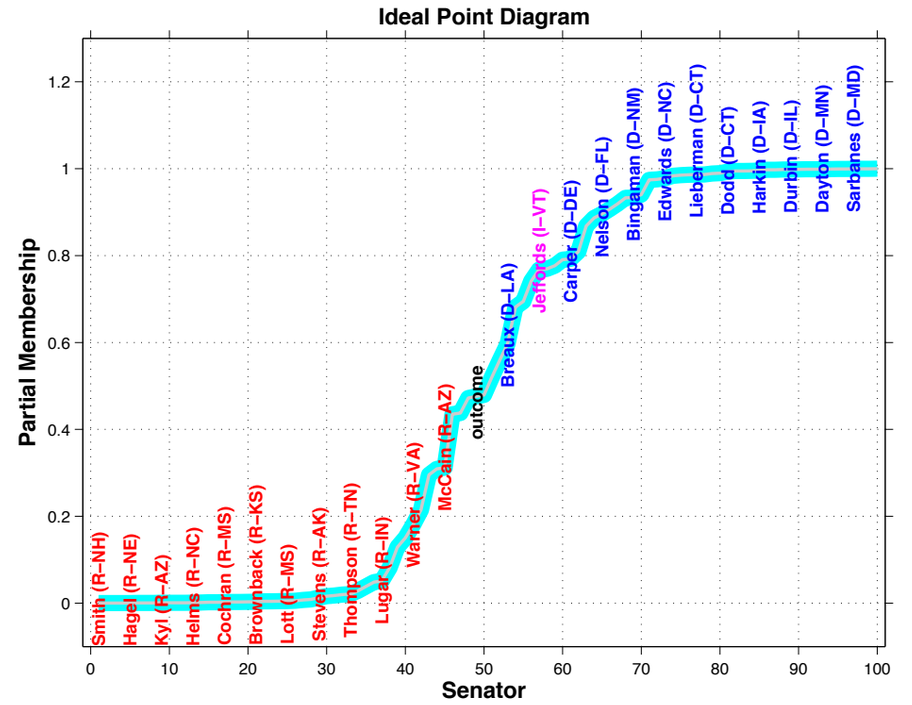
$$\mathcal{F}(y,q) = \mathbb{E}_{q(z)}[\log p(y|f_\theta(z))] - \frac{1}{2}\sum_i (\sigma_i^2 + \mu_i^2 - 1 - \ln \sigma_i^2)$$

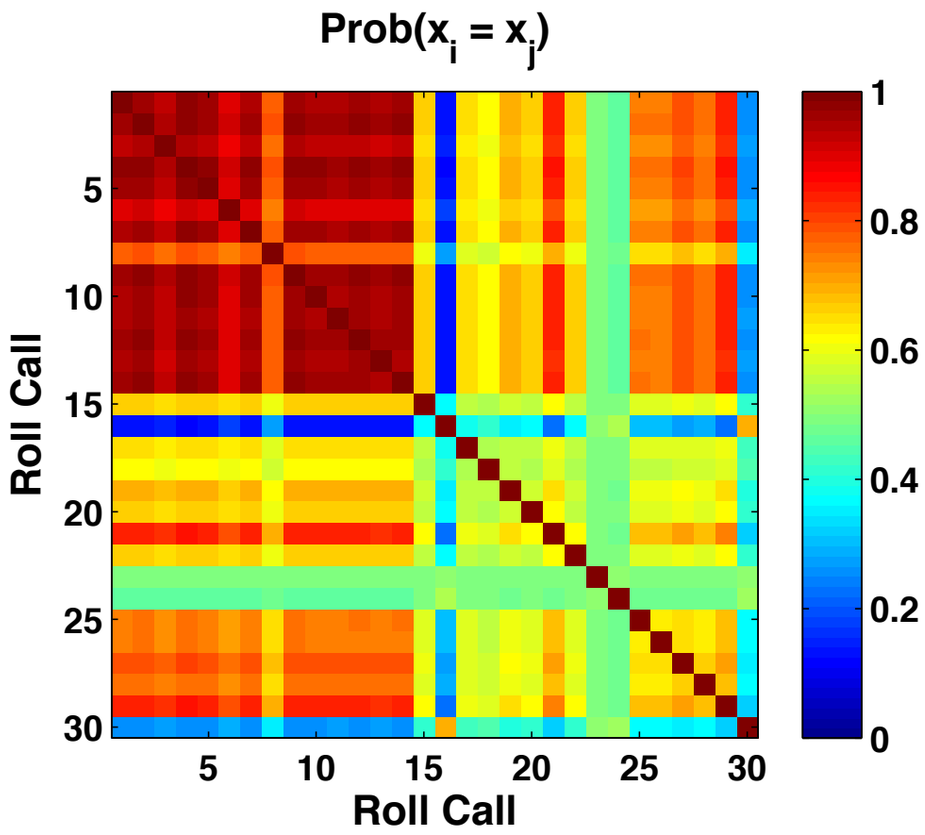
Data Visualisation

Binary data set of votes in the US senate.

Factor Analysis



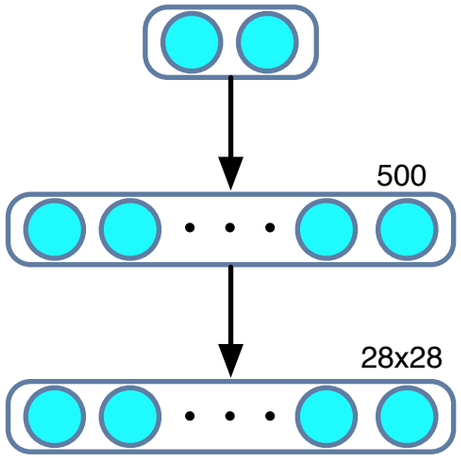


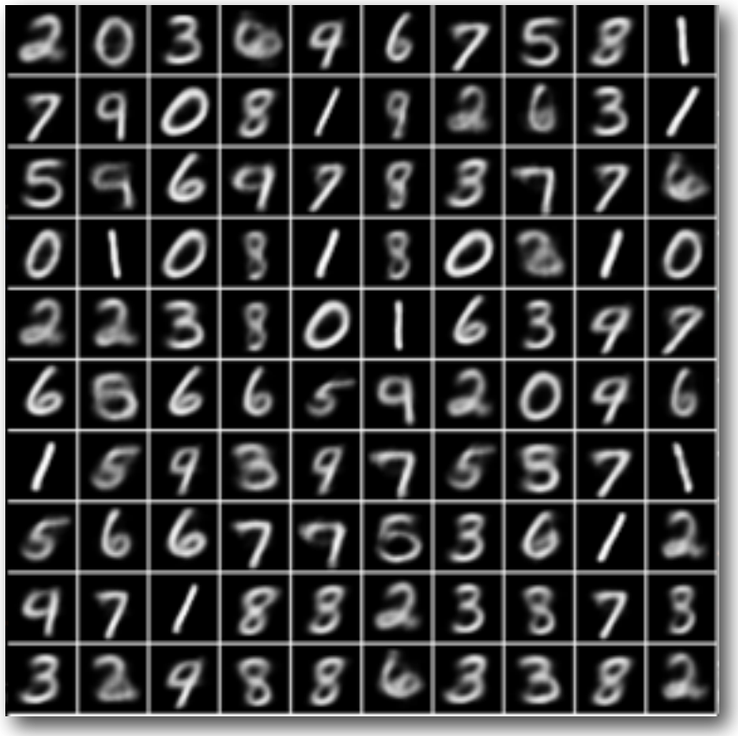


Data Visualisation

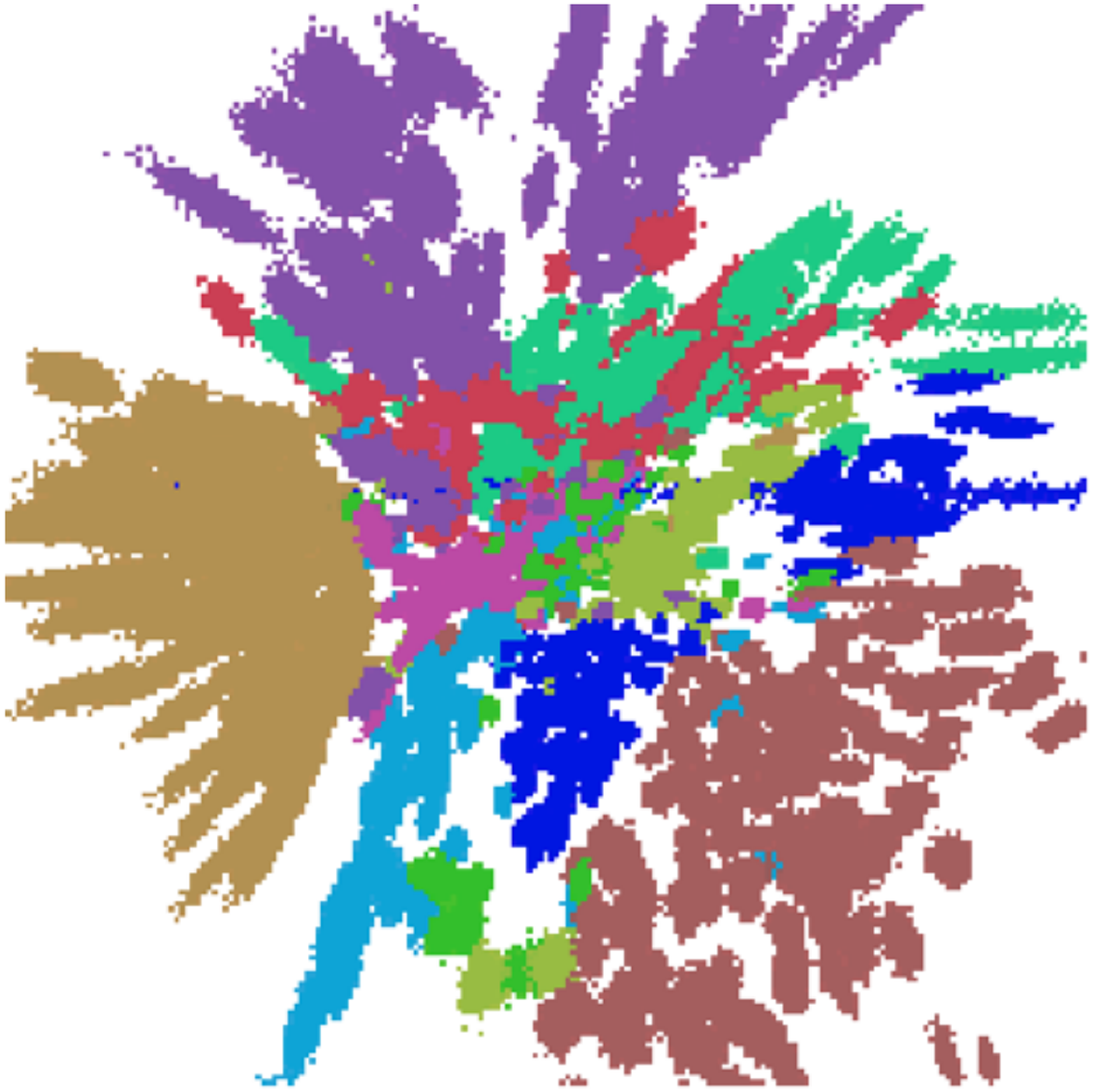
MNIST Handwritten digits

DLGM





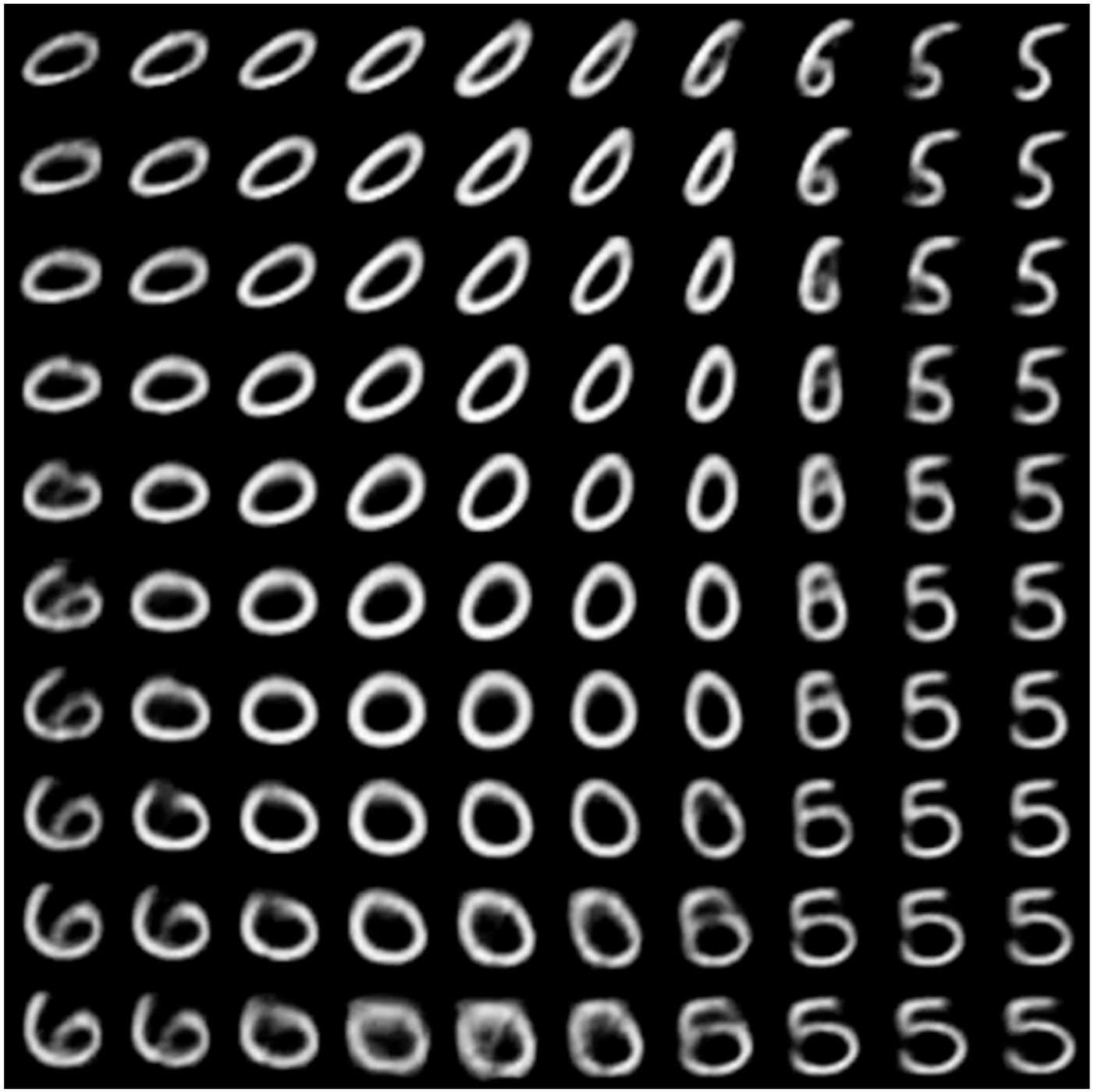
Samples from 2D latent model



Labels in 2D latent space

Visualizing MNIST in 3D

DLGM



Data Simulation

DLGM

Data

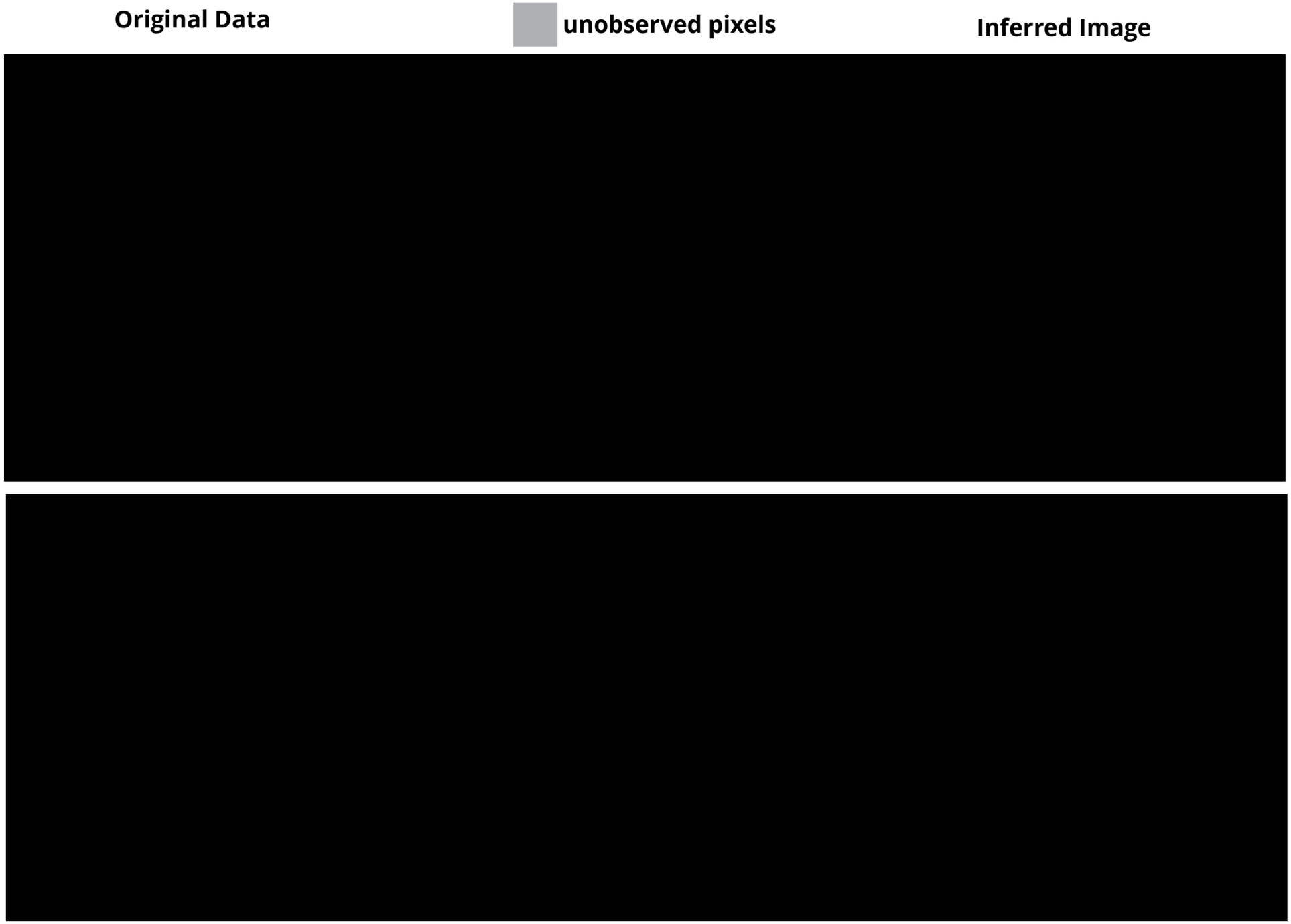
Samples

Missing Data Imputation

DLGM

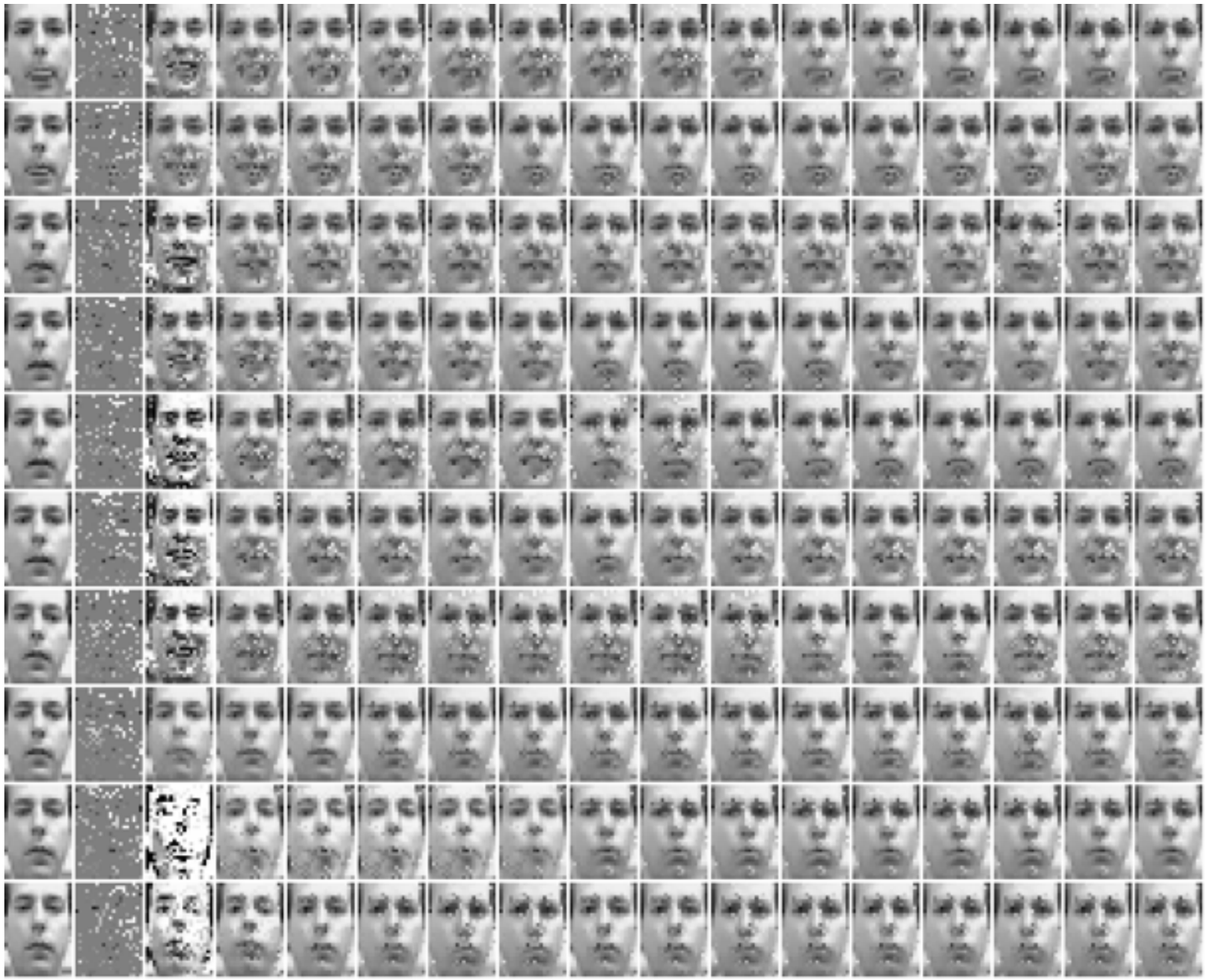
10%
observed

50%
observed



Missing Data Imputation

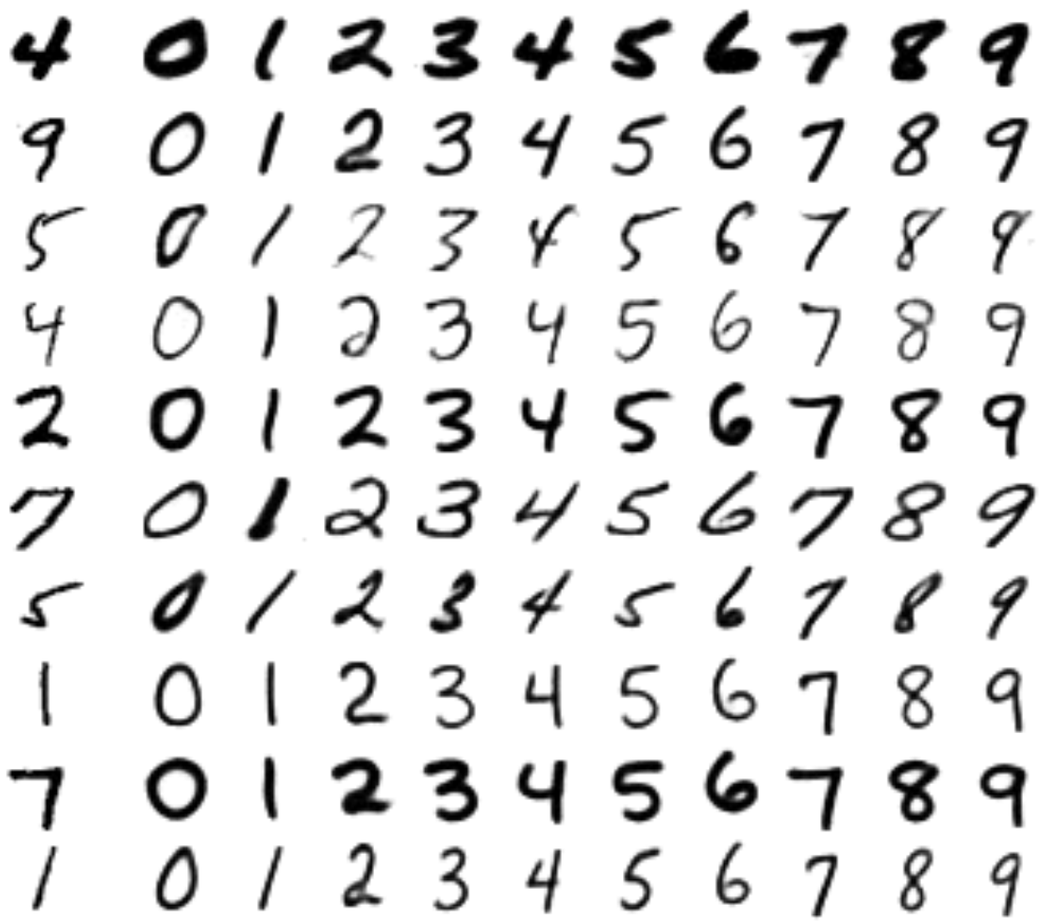
Frey Faces dataset. Completion: **80% missing pixels**

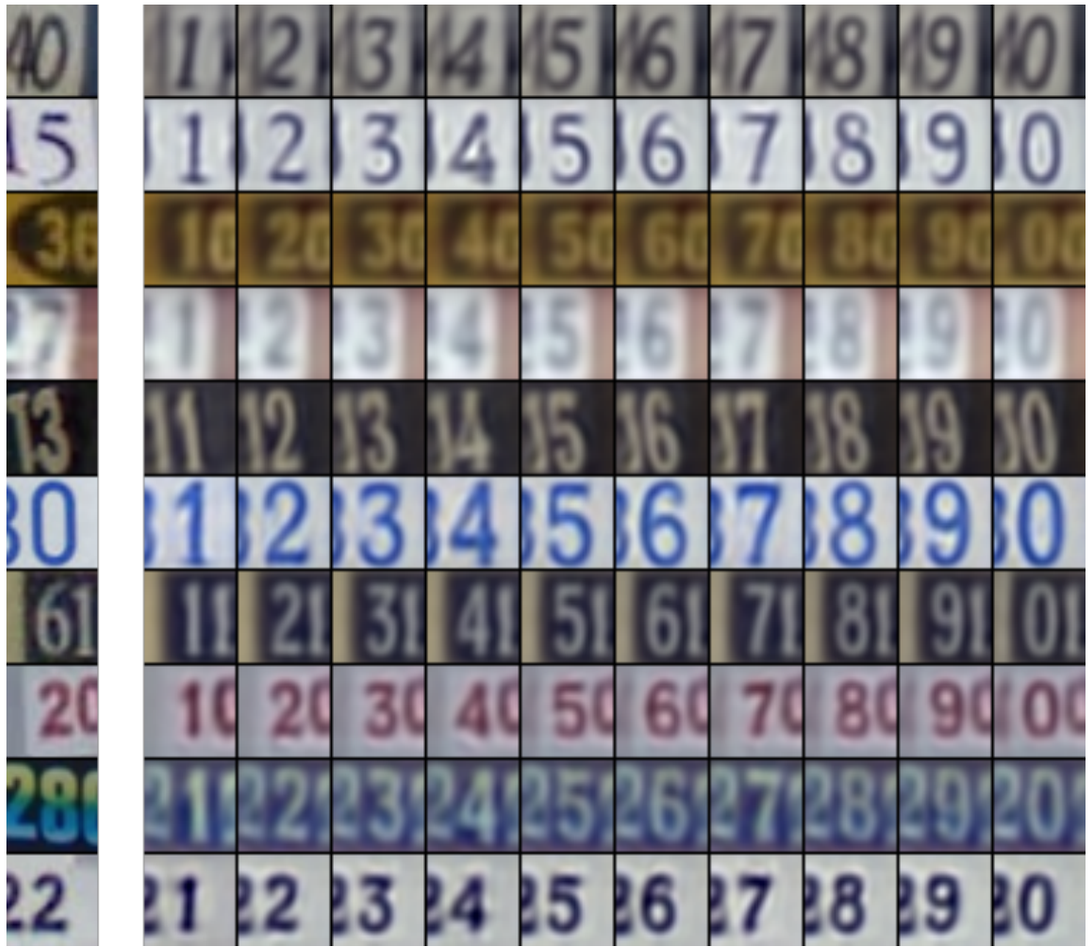


DLGM

Analogical Reasoning

Semi-supervised DLGM

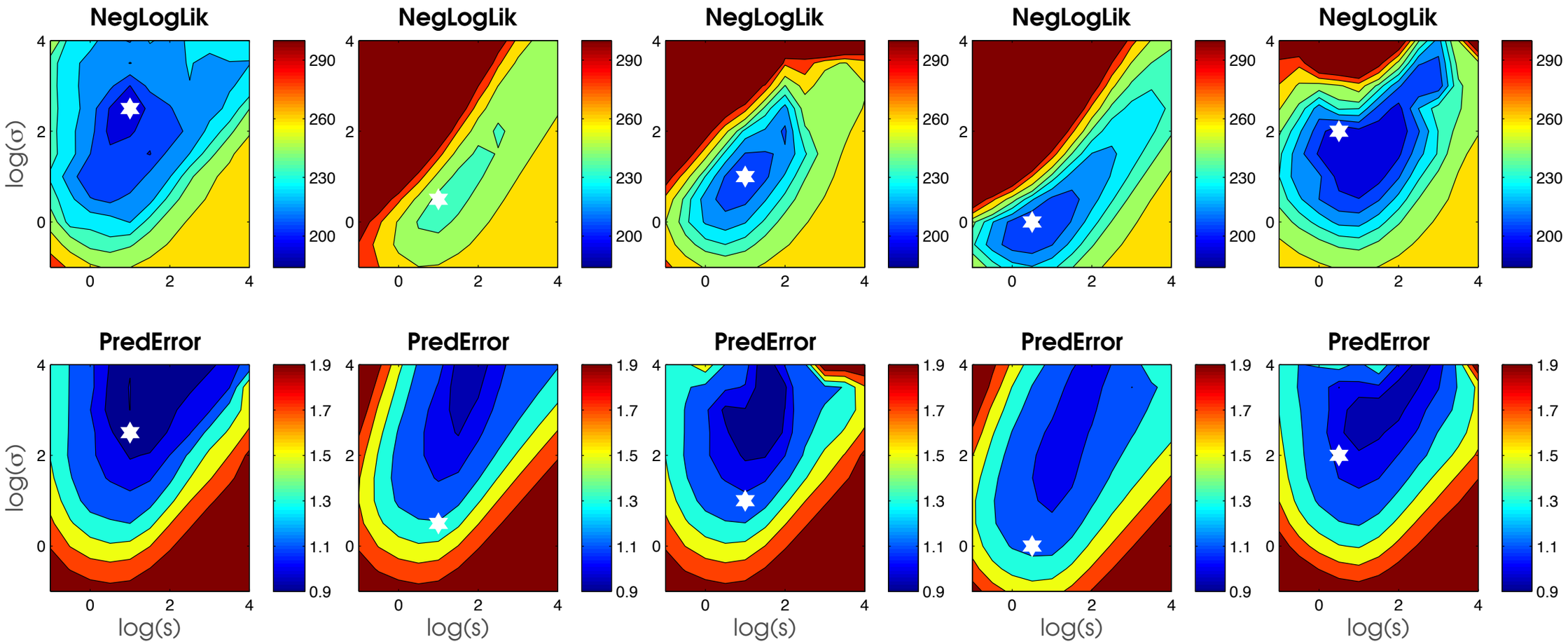




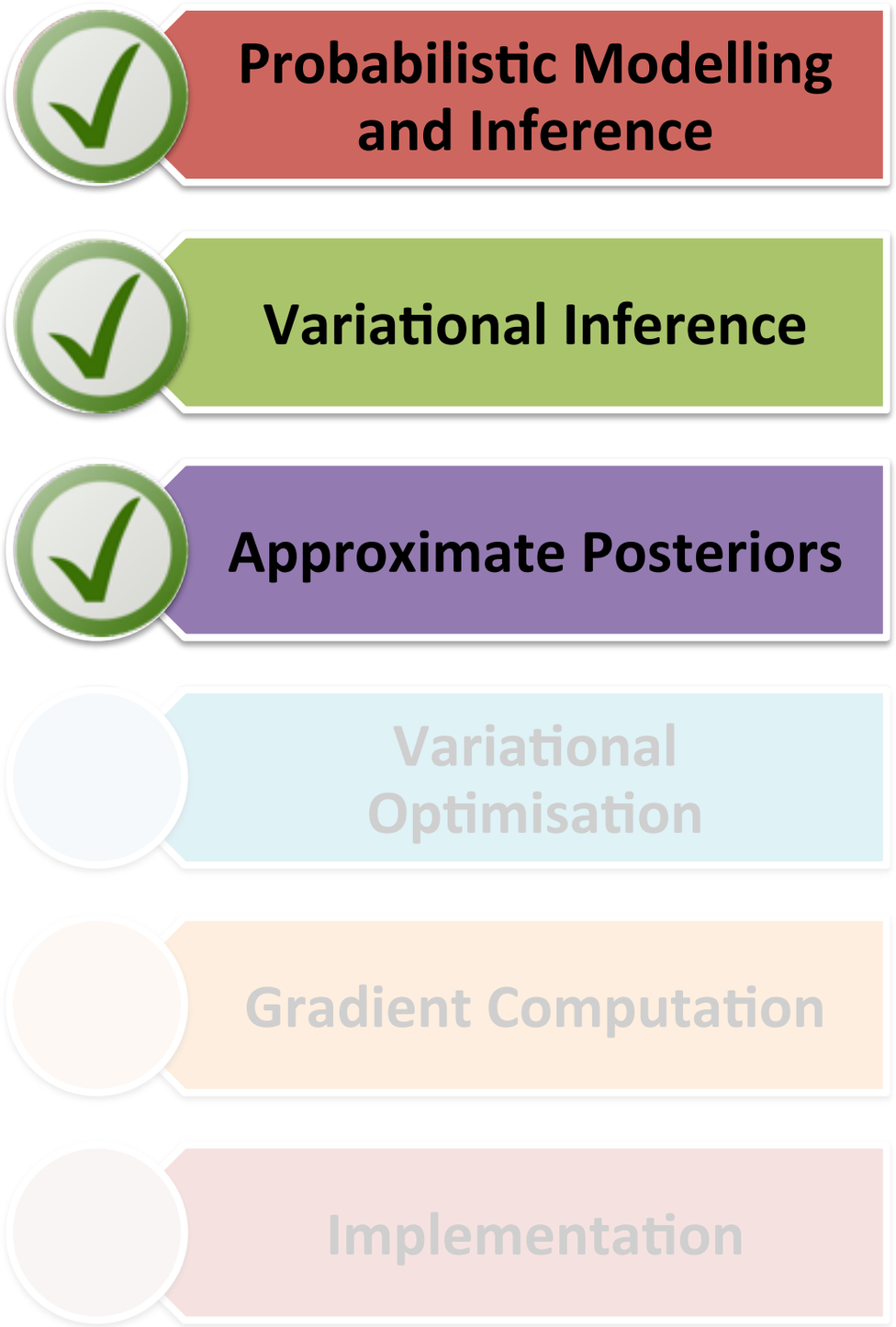
Model Selection

Get marginal likelihood estimates that allow for model selection.

GP Regression



Progress ...


Probabilistic Modelling and Inference
Variational Inference
Approximate Posteriors
Variational Optimisation
Gradient Computation
Implementation

Optimising the Variational Objective

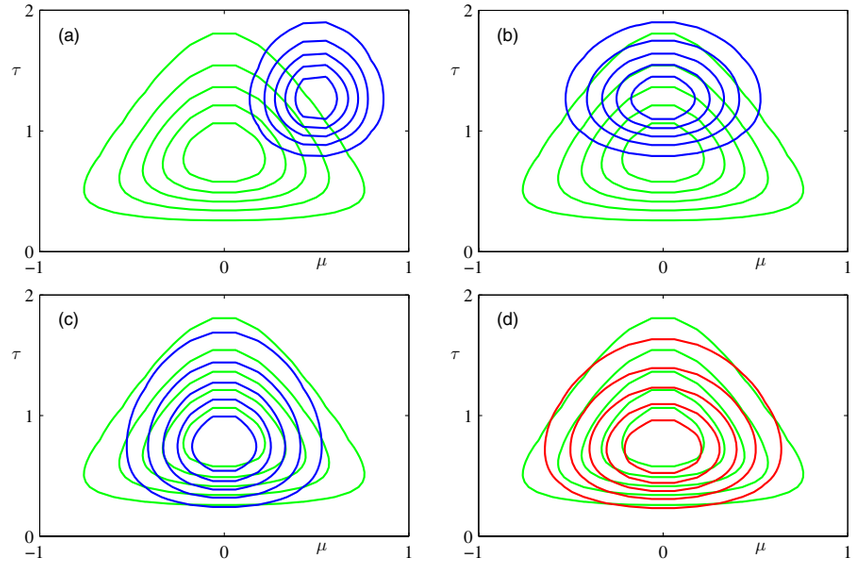
$$\max_{q,\theta} \mathcal{F}(y, q) = \mathbb{E}_{q(z)}[\log p(y|z, \theta)] - KL[q(z)\|p(z)]$$

Approx. Posterior

Reconstruction

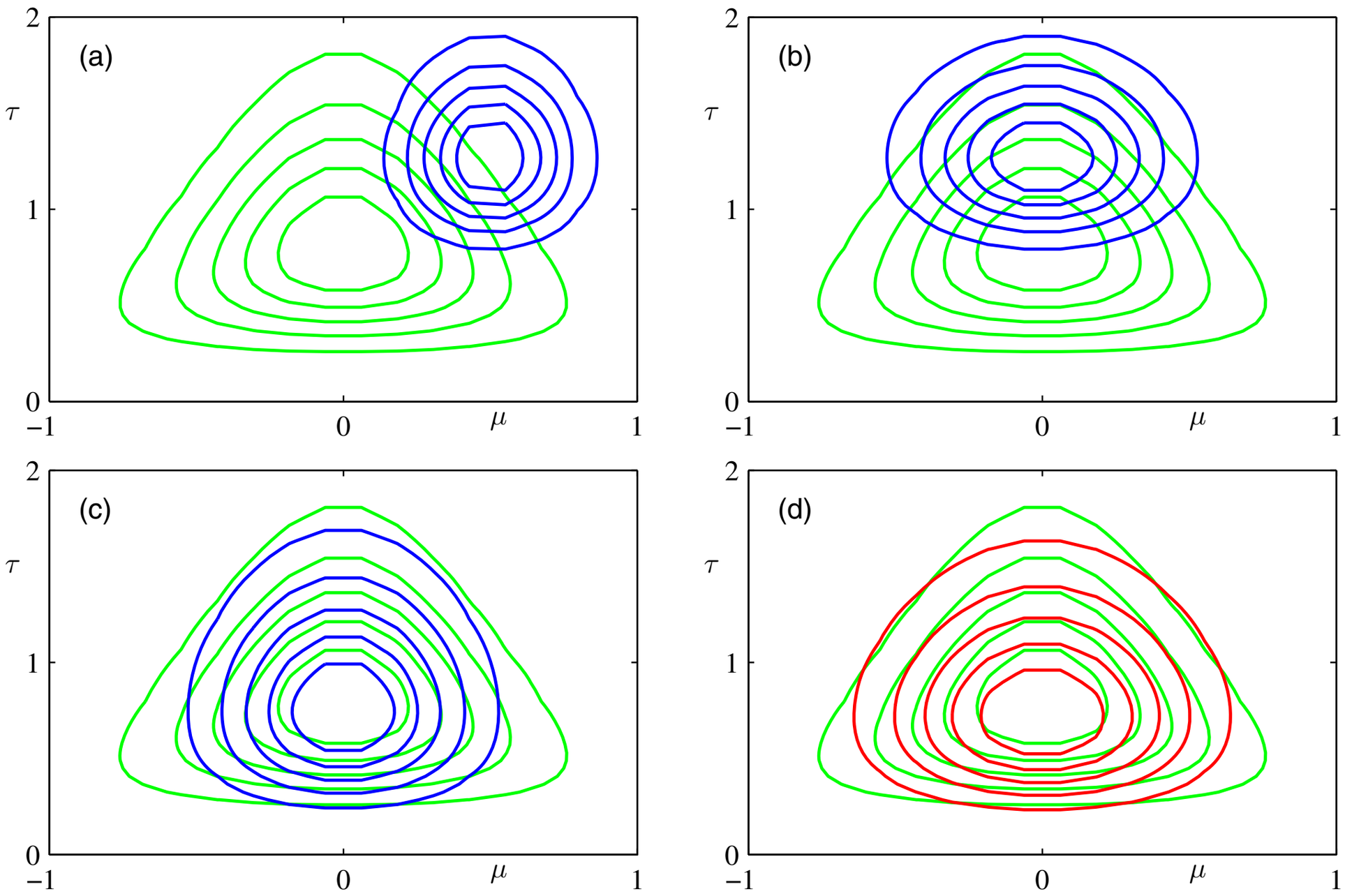
Penalty

- *Variational EM*
- *Stochastic Variational Inference*
- *Doubly Stochastic Variational Inference*
- *Amortised Inference*



Optimising the Variational Objective

Example of variational optimisation for a simple 2D density.



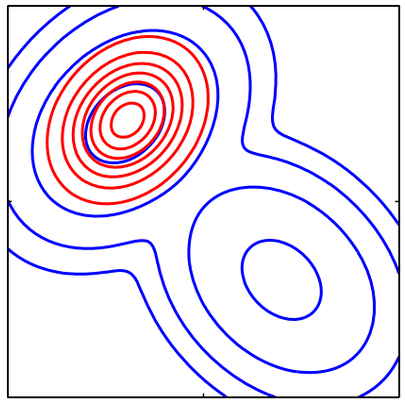
What optimisation schemes can we use to achieve this?

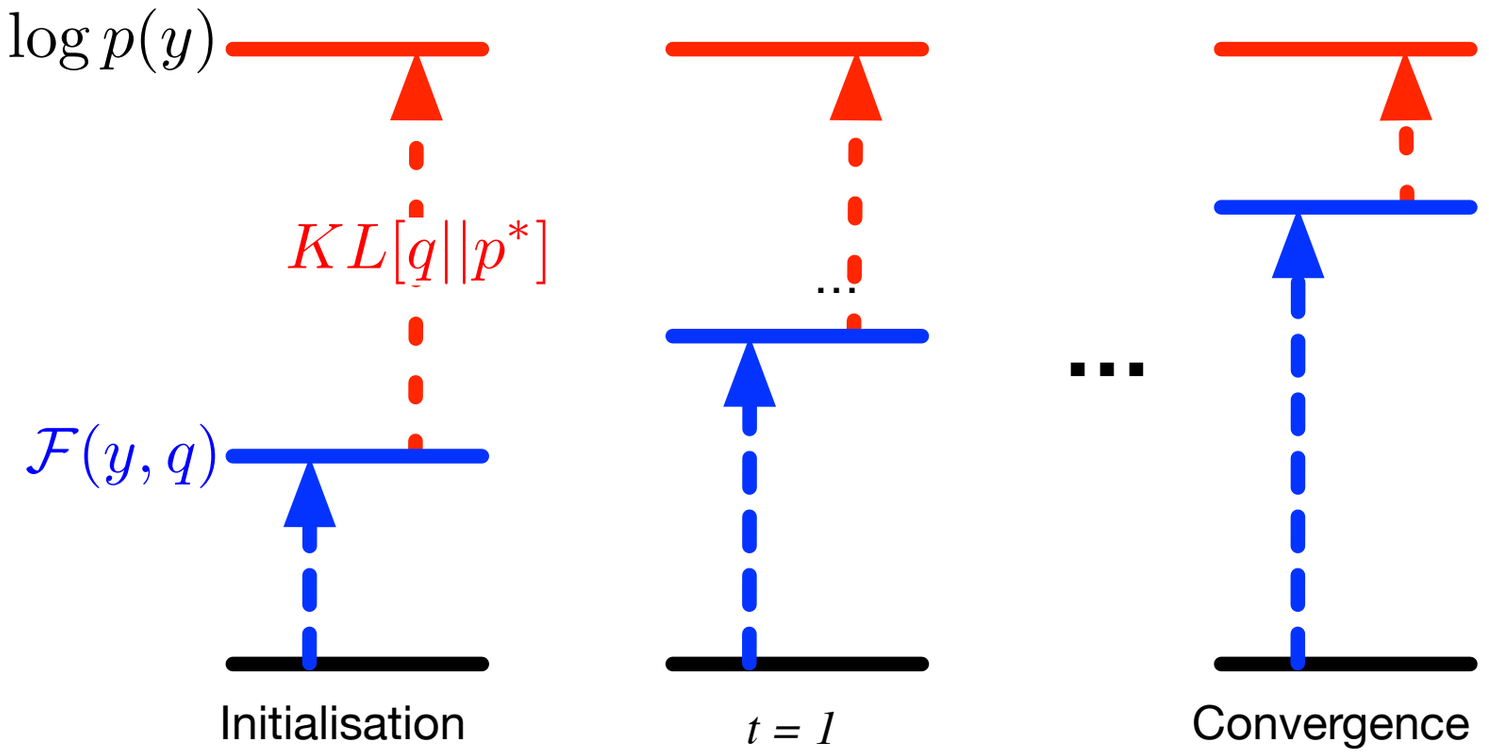
Variational Expectation Maximisation

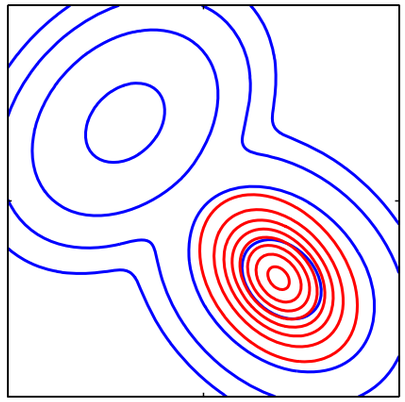
Alternating optimisation for the variational parameters and then model parameters (VEM).

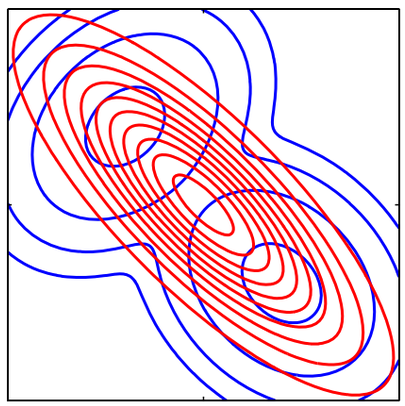
Repeat:

E-step	$\phi \propto \nabla_\phi \mathcal{F}(y, q)$	***Var. params***
M-step	$\theta \propto \nabla_\theta \mathcal{F}(y, q)$	***Model params***









Variational EM

Involves computation over the entire data set.

Repeat:

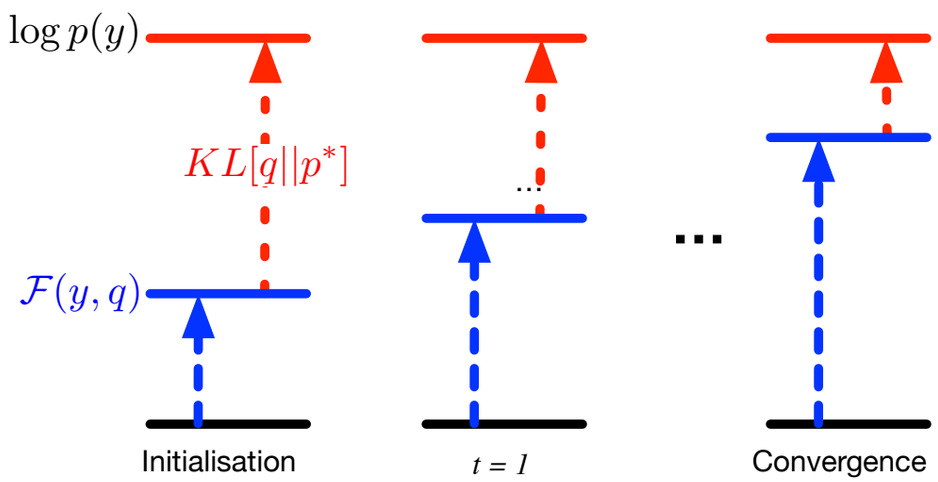
E-step *(Inference)*

For i = 1, ... N

$$\phi_n \propto \nabla_\phi \mathbb{E}_{q_\phi(z)}[\log p_\theta(y_n|z_n)] - \nabla_\phi KL[q(z_n)||p(z_n)]$$

M-step *(Parameter Learning)*

$$\theta \propto \frac{1}{N}\sum_n \mathbb{E}_{q_\phi(z)}[\nabla_\theta \log p_\theta(y_n|z_n)]$$



Stochastic Variational Inference

Instead use a **stochastic gradient based on a mini-batch** of data.

Many names: *online EM, stochastic approximation EM, stochastic variational inference*.

Repeat:

Mini-batch E-step

For i = 1, ... N

$$\phi_n \propto \nabla_\phi \mathbb{E}_{q_\phi(z)}[\log p_\theta(y_n|z_n)] - \nabla_\phi KL[q(z_n)||p(z_n)]$$

M-step

$$\theta \propto \frac{1}{N}\sum_n \mathbb{E}_{q_\phi(z)}[\nabla_\theta \log p_\theta(y_n|z_n)]$$

***N* is a mini-batch:** sampled with replacement from the full data set or received online.

Scalable - only need to operate on a small batch at a time. Can operate on large data sets.

Doubly Stochastic Variational Inference

VEM and SVI assume easy computation of the expected log-likelihood (and KL).

E-step: $$\phi_n \propto \nabla_\phi \mathbb{E}_{q_\phi(z)}[\log p_\theta(y_n|z_n)] - \nabla_\phi KL[q(z_n)||p(z_n)]$$

Instead compute all expectations by Monte Carlo approximation.

Doubly stochastic estimation : one source of stochasticity from the mini-batch, another from the Monte Carlo evaluation of the expectation.

Monte Carlo E-step: $$z_n^{(s)} \sim q(z_n|y_n)$$

$$\phi_n \propto \nabla_\phi \frac{1}{S} \sum_s \left[\log p_\theta(y_n|z_n(\phi)^{(s)}) - \log \frac{q(z_n(\phi)^{(s)})}{p(z)} \right]$$

General idea only. Will make precise when we look at Monte Carlo estimators.

Amortised Variational Inference

Repeat:

E-step

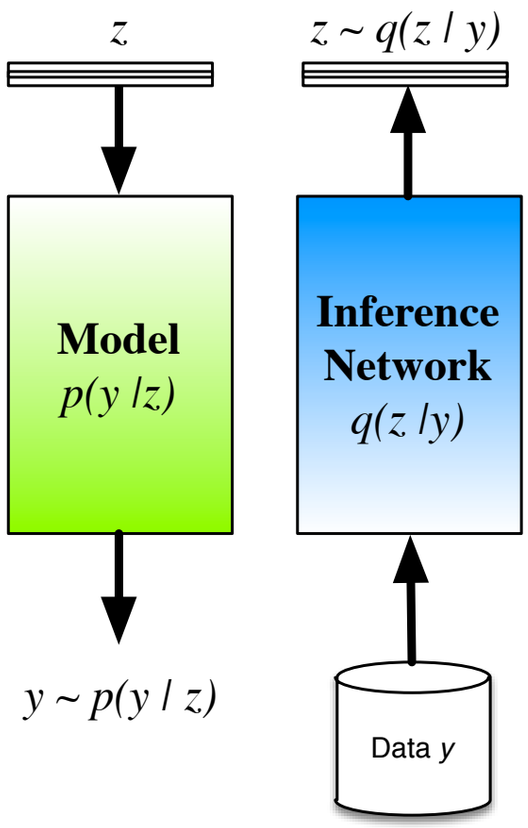
For i = 1, ... N

$$\phi_n \propto \nabla_\phi \mathbb{E}_{q_\phi(z)}[\log p_\theta(y_n|z_n)] - \nabla_\phi KL[q(z_n)\|p(z_n)]$$

Instead of solving this optimisation for every data point *n*, we can instead use a model.

M-step

$$\theta \propto \frac{1}{N}\sum_n \nabla_\theta \log p_\theta(y_n|z_n)$$



Inference network: q is an encoder or inverse model.

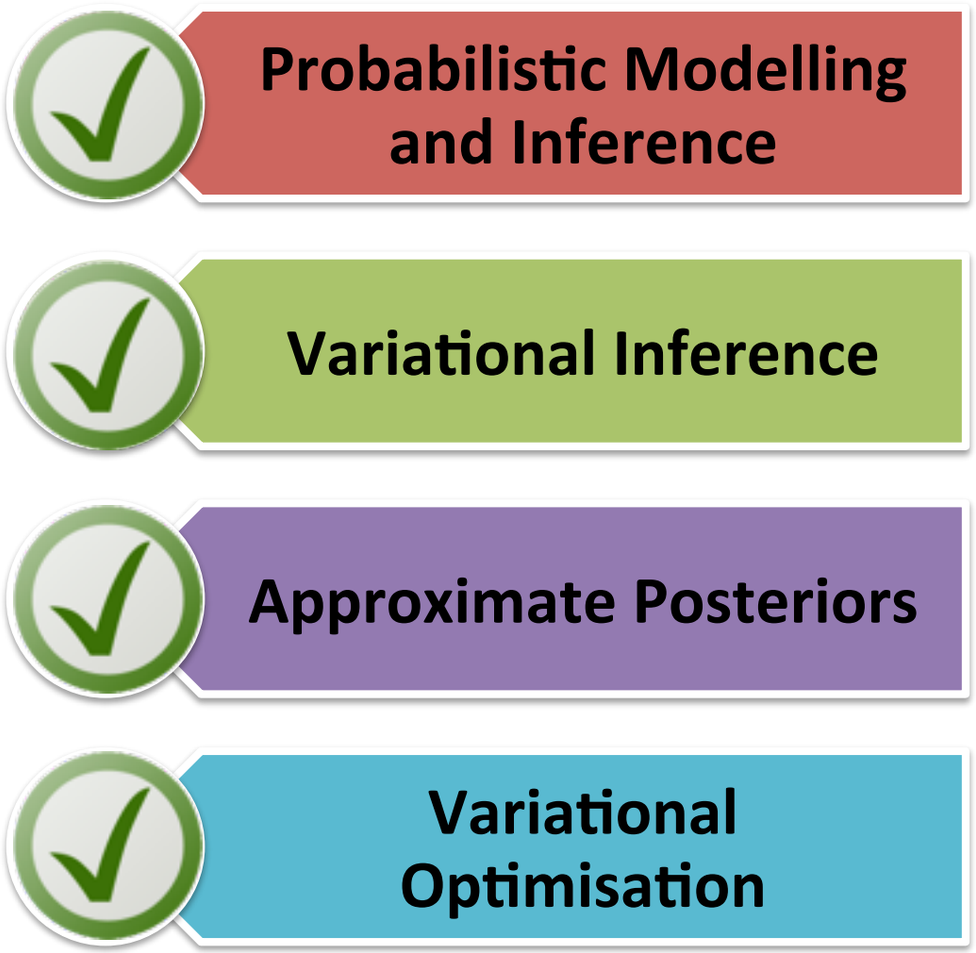
Parameters of q are now a set of global parameters used for inference of all data points - test and train.

Share the cost of inference (amortise) over all data.

Combines easily with mini-batches and Monte Carlo expectations.

Can jointly optimise variational and model parameters: no need for alternating optimisation.

Progress ...


Probabilistic Modelling and Inference
Variational Inference
Approximate Posteriors
Variational Optimisation
Gradient Computation
Implementation

Computing the expected log-likelihood

An outstanding issue in all the optimisation methods is the computation of the expected log-likelihood (and KL term if unknown).

$$\nabla_\xi \mathbb{E}_{q(z)}[\log p_\theta(y_n|z_n)]$$

- We don't know this expectation in general.
- The parameters of the distribution with respect to which the expectation is taken.

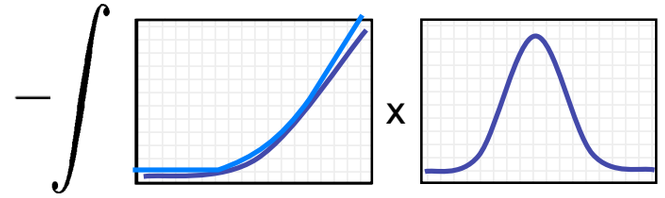
Two general approaches:

- **Deterministic methods:** use additional bounds to make this computation easier - local variational methods.
- **Stochastic methods:** Compute the expectation by Monte Carlo and use properties of the distributions involved to simplify computation.

Local Variational Methods

Replace the likelihood with a simpler form — a lower bound that makes the expectation easy to compute.

$$\mathbb{E}_{q(z)}[\log p_\theta(y_n|z_n)]$$

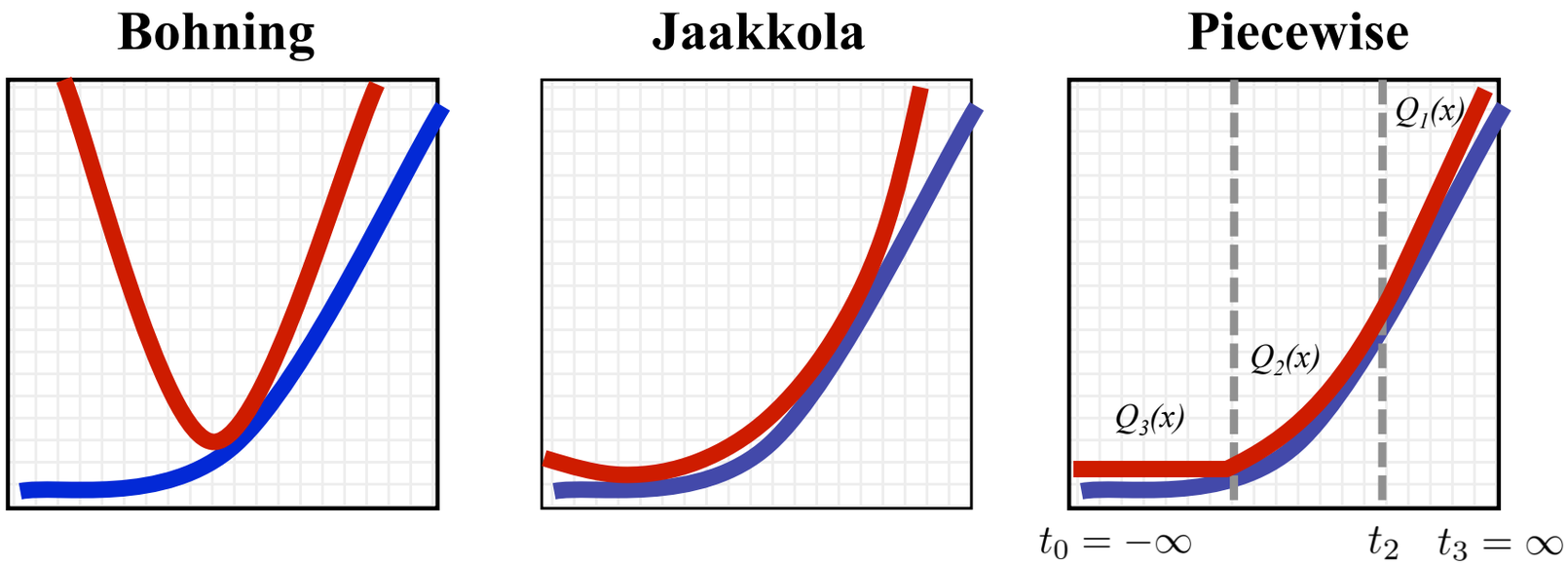


Original problem $$p(y=1|z) = \frac{1}{1+\exp(-z)} = \sigma(z)$$

Local Bound $$\sigma(z) \geq \sigma(\xi)\exp\left(\frac{z-\xi}{2} - \lambda(\xi)(z^2-\xi^2)\right)$$

Additional variational parameters ξ

Bound with only linear or quadratic terms: expectations, especially against a Gaussian, are easy to compute.



Stochastic Backpropagation

A Monte Carlo method that works with continuous latent variables.

Original problem

$$\nabla_\xi \mathbb{E}_{q(z)}[f(z)]$$

Reparameterisation

$$z \sim \mathcal{N}(\mu, \sigma^2)$$
$$z = \mu + \sigma\epsilon \quad \epsilon \sim \mathcal{N}(0,1)$$

Backpropagation with Monte Carlo

$$\nabla_\xi \mathbb{E}_{\mathcal{N}(0,1)}[f(\mu + \sigma\epsilon)]$$
$$\mathbb{E}_{\mathcal{N}(0,1)}[\nabla_{\xi=\{\mu,\sigma\}} f(\mu + \sigma\epsilon)]$$

- Can use *any likelihood function,* avoids the need for additional lower bounds.
- *Low-variance*, unbiased estimator of the gradient.
- Can use just *one sample* from the base distribution.
- Possible for many distributions with location-scale or other known transformations, such as the CDF.

Monte Carlo Control Variate Estimators

More general Monte Carlo approach that can be used with both discrete or continuous latent variables.

Property of the score function: $$\nabla_\xi \log q_\xi(z|x) = \frac{\nabla_\xi q_\xi(z|x)}{q_\xi(z|x)}$$

Original problem $$\nabla_\phi \mathbb{E}_{q_\phi(z)}[\log p_\theta(y|z)]$$

Score ratio $$\mathbb{E}_{q_\phi(z)}[\log p_\theta(y|z) \nabla_\phi \log q(z|y)]$$

MCCV Estimate $$\mathbb{E}_{q_\phi(z)}[(\log p_\theta(y|z) - c) \nabla_\phi \log q(z|y)]$$

c is known as a ***control variate*** and is used to control the variance of the estimator.

Progress ...

- ✓ Probabilistic Modelling and Inference
- ✓ Variational Inference
- ✓ Approximate Posteriors
- ✓ Variational Optimisation
- ✓ Gradient Computation
- Implementation

Implementing your Variational Algorithm

Avoid deriving pages of gradient updates for variational inference.

Variational inference turns integration into optimisation:

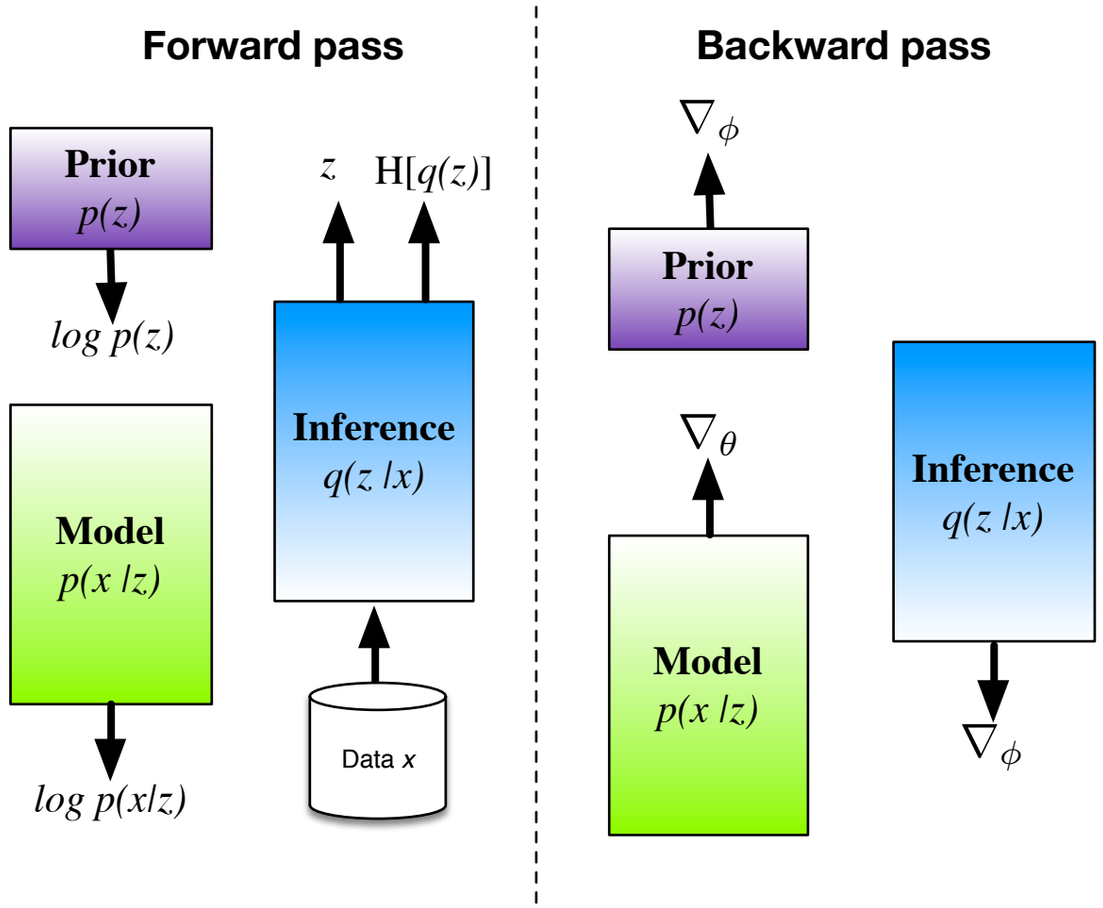
- **Automated Tools:**

  Differentiation: Theano, Torch7.

  Message passing: infer.NET

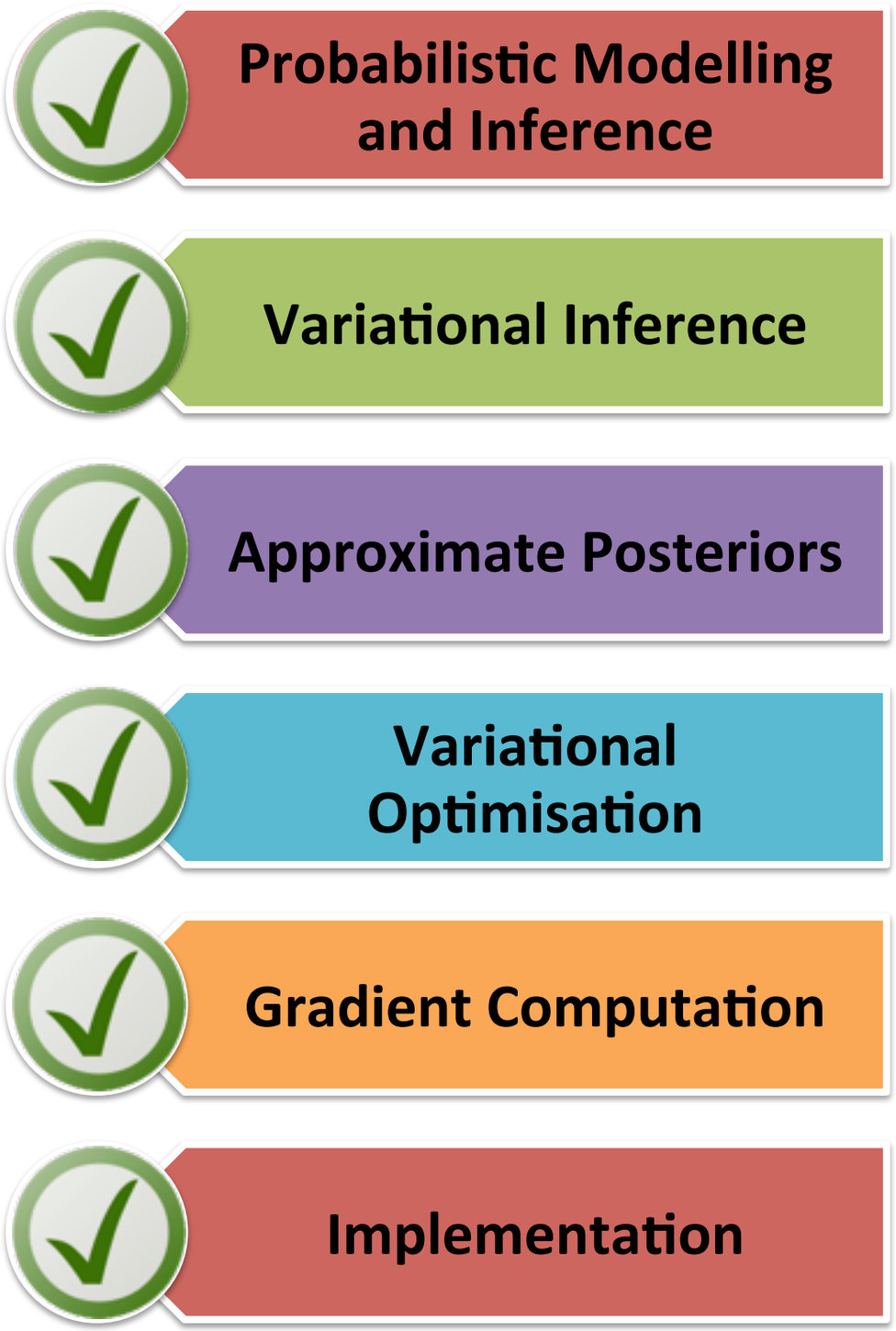
- Stochastic gradient descent and other preconditioned optimisation.
- Same code can run on both GPUs or on distributed clusters.
- Probabilistic models are modular, can easily be combined.

$$\mathbb{E}_q[(-\log p(y|z) + \log q(z) - \log p(z)]$$



Ideally want probabilistic programming using variational inference.

Progress ...


Probabilistic Modelling and Inference
Variational Inference
Approximate Posteriors
Variational Optimisation
Gradient Computation
Implementation

Variational Inference Theory

- **Tightness of the bound:**
  - The bound is exact is if q is the true posterior.
  - For certain classes of q-distributions, we can show that the class of distributions is rich enough to include the true posterior distributions.
- **Convexity and duality**
  - For Latent Gaussian models, and many others, we can show convexity of the variational objective. This allows us to exploit other optimisation approaches such as dual decomposition.
- **Bound correction**
  - We can obtain a tighter bound in a number of settings using perturbation analysis.

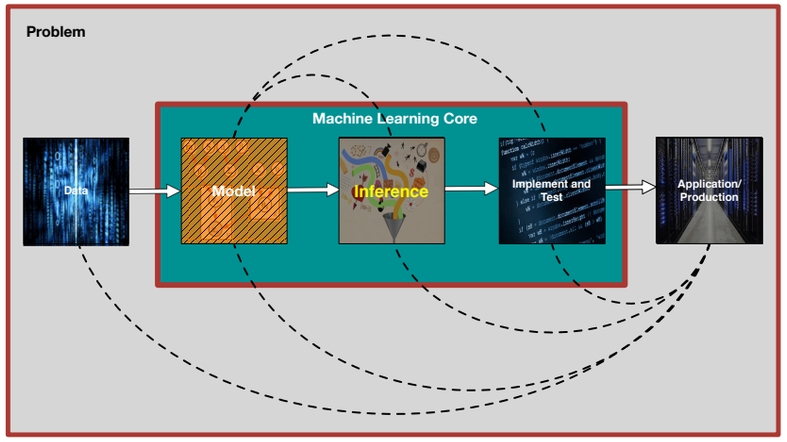
Variational Inference Theory

- **Convergence**
  - Based on VEM, we can show convergence to local minima.
  - We can also show theoretically for certain models that we have local convergence to the optimum in asymptotic settings.
- **Consistency**
  - We can show consistency of the mean of maximum likelihood parameter estimates, for some types of latent variable models using properties of the functional derivative. In other cases, we can show that we get inconsistent estimators.
- **Asymptotic normality**
  - We can use the theory for asymptotic normality of Laplace approximations to show asymptotic normality for certain classes of models using variational inference.

Other Variational Problems

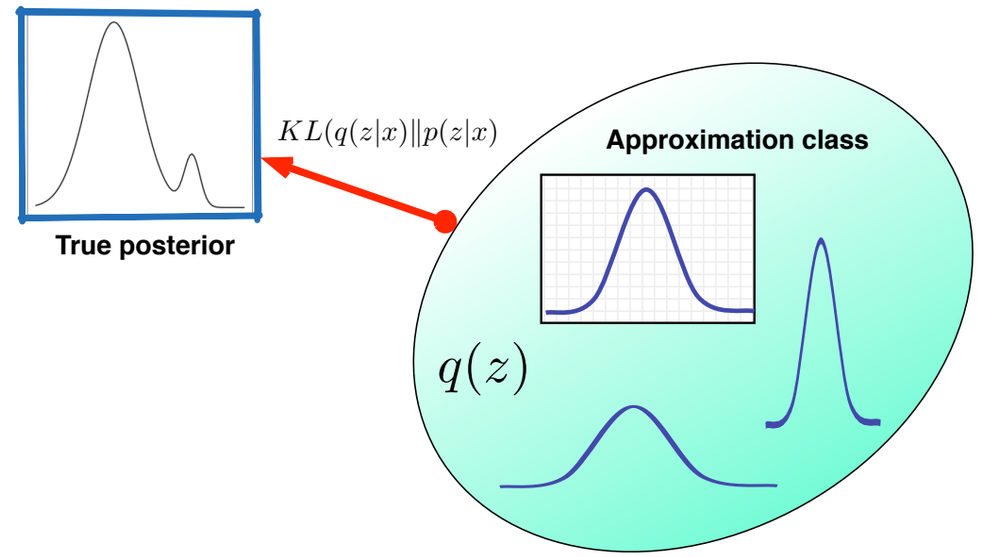
- *Belief propagation*
- *Expectation propagation*
- *Mutual Information maximisation*
- *Rate distortion theory*
- *Information bottleneck*
- *Policy search methods*

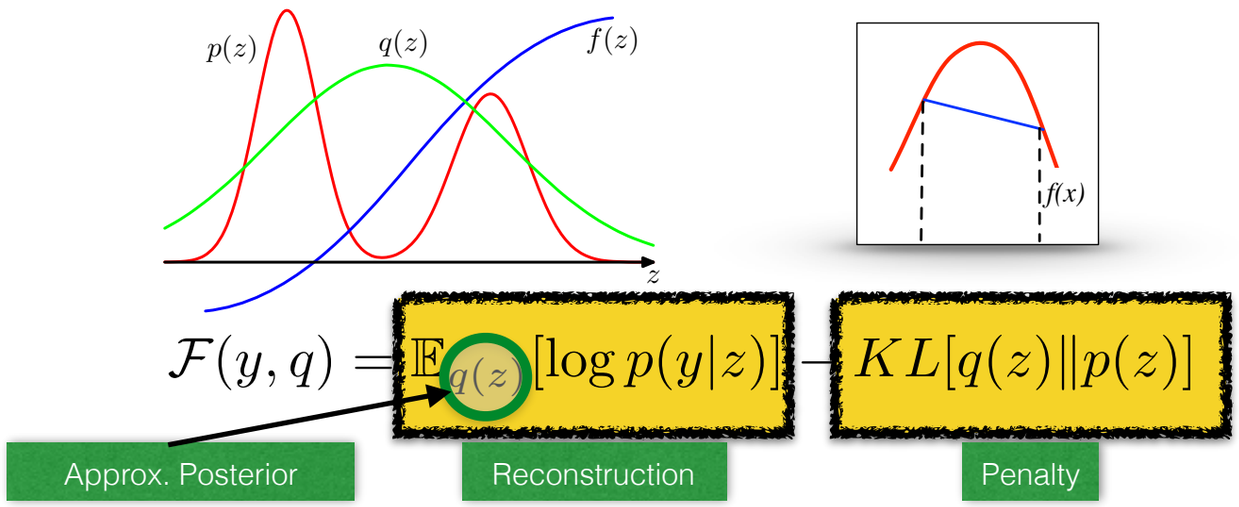
In Review …



Explored the **central role of statistical inference** in Machine Learning and data science.

Looked at the **variational approach** as one powerful and compelling method for inference.

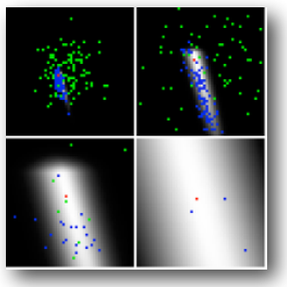




Moved from importance sampling to variational inference by applying the variational principle giving us the **variational lower bound.**

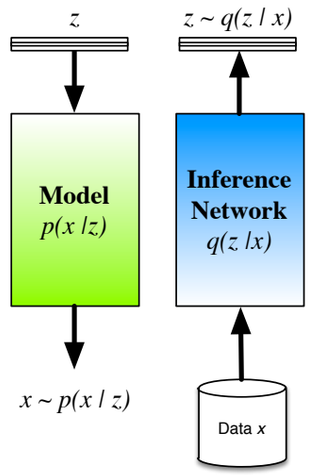
In Review …

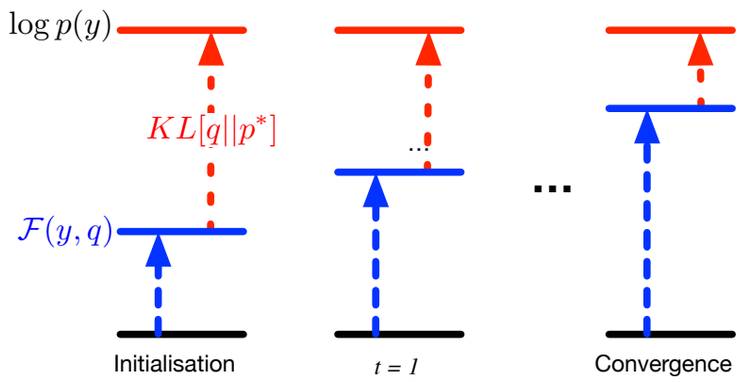
$q_\phi(z) = f(z;\phi)$

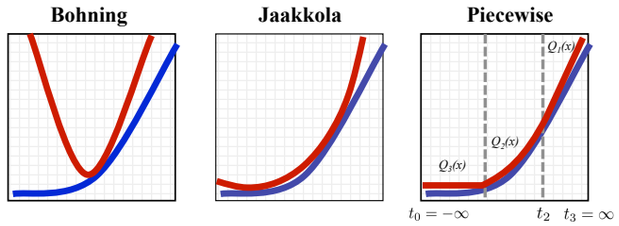


Fixed-form variational inference specifies the class of posterior approximations.

Many different ways to optimise the variational objective. Most commonly use **stochastic and amortised inference**.



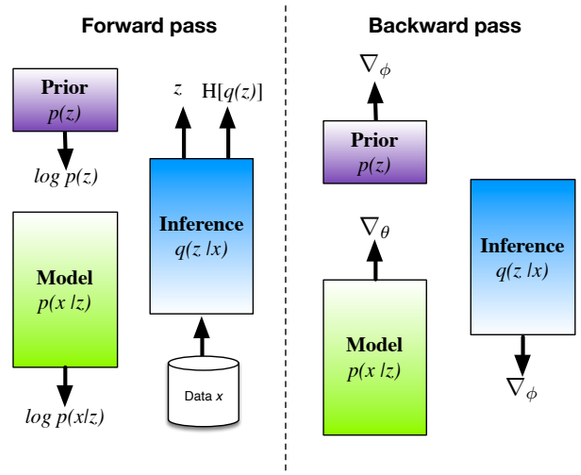




$$\mathbb{E}_{\mathcal{N}(0,1)}[\nabla_{\xi=\{\mu,\sigma\}} f(\mu + \sigma\epsilon)]$$

Gradients can be computed in many ways. **Monte Carlo gradients** are most generally applicable.

Automate as much as possible when you implement your variational algorithm.



Not mentioned

- **Posterior approximation:**
  - Mixture models
  - Non-parametric approaches
  - Hamiltonian variational approximation

- **Optimisation:**
  - Variational message passing
  - Memoised inference

- **Gradient computations:**
  - Delta method and Laplace approaches.
  - Natural gradients

- **Implementation**
  - VB building blocks

Thanks to many people:

Danilo Rezende, Charles Blundell, Theophane Weber, Andriy Mnih, Karol Gregor, Daan Wierstra (*Google DeepMind*).

Durk Kingma, Max Welling (*U. Amsterdam*)

Emtiyaz Khan (*EPFL*), Kevin Murphy (*Google*)

Thank You.

Some References

Textbooks

- Bishop, Christopher M. ***Pattern recognition and machine learning***. New York: springer, 2006.
- Murphy, Kevin P. ***Machine learning: a probabilistic perspective***. MIT press, 2012.
- Barber, David. ***Bayesian reasoning and machine learning***. Cambridge University Press, 2012.

Overview

- Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., & 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.
- Meng, Anders. "***An introduction to Variational calculus in Machine Learning***." (2004).

Modern Variational Inference

- Marlin, Benjamin M., Mohammad Emtiyaz Khan, and Kevin P. Murphy. "***Piecewise Bounds for Estimating Bernoulli-Logistic Latent Gaussian Models.***" ICML. 2011.
- Hoffman, M. D., Blei, D. M., Wang, C., & Paisley, J. (2013). ***Stochastic variational inference***. The Journal of Machine Learning Research, 14(1), 1303-1347.
- Wingate, D. and Weber, T. ***Automated variational inference in probabilistic programming***. 2013
- Ranganath, Rajesh, Sean Gerrish, and David M. Blei. "***Black box variational inference.***" AISTATS (2014).
- Kingma, Diederik P., and Max Welling. "***Auto-encoding variational Bayes***." ICLR 2014.
- Rezende, Danilo Jimenez, Shakir Mohamed, and Daan Wierstra. "***Stochastic backpropagation and approximate inference in deep generative models***." ICML (2014).
- Mnih, Andriy, and Karol Gregor. "***Neural variational inference and learning in belief networks***." ICML (2014).
- Gregor, Karol, et al. "***Deep autoregressive networks***." ICML (2014).
- Titsias, Michalis, and Miguel Lázaro-Gredilla. "***Doubly Stochastic Variational Bayes for non-Conjugate Inference.***" Proceedings of the 31st International Conference on Machine Learning (ICML-14). 2014.
- Kingma, D. P., Mohamed, S., Rezende, D. J., & Welling, M. (2014). ***Semi-supervised learning with deep generative models***. NIPS (pp. 3581-3589).