Variational Inference

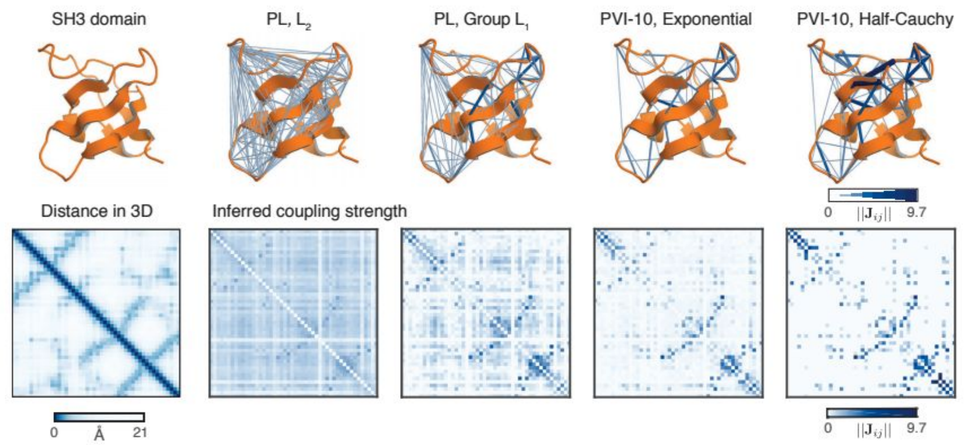
Introduction & ELBO-Derivations

Motivation, & Bayesian Modelling

Probabilistic modelling (PM) is widely used in ML …

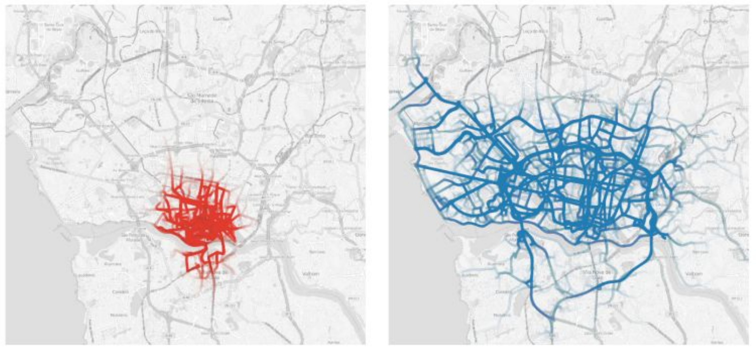
[Radford et.a. 2019]

Image generation

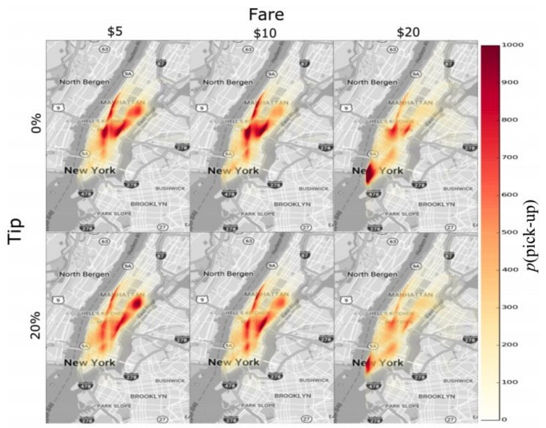


Protein Contact Predictions

[Kucukelbir et. al. 2016]

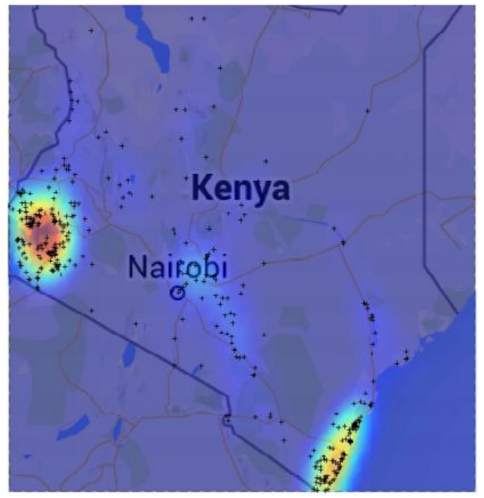


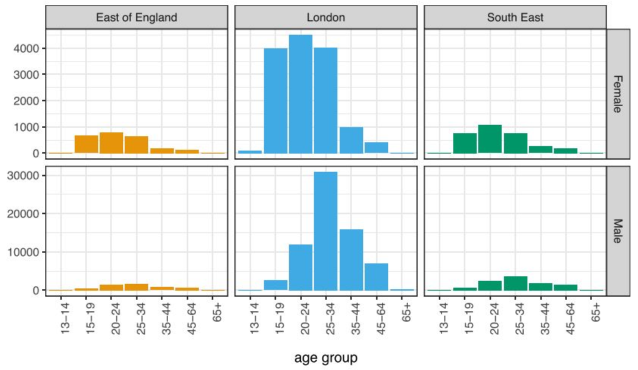
[Trippe et. al. 2018]

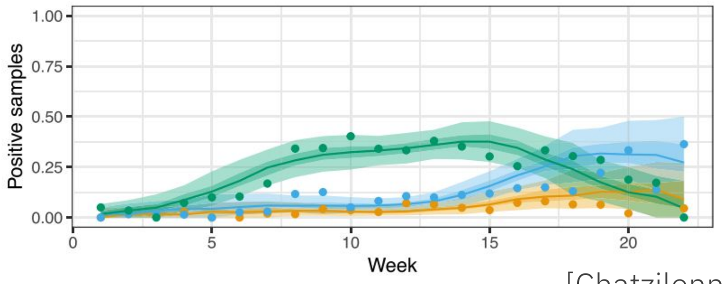


Taxi and Fare Analysis

[Lloyd et. al. 2015]

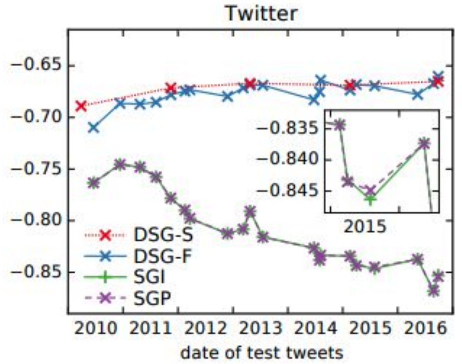






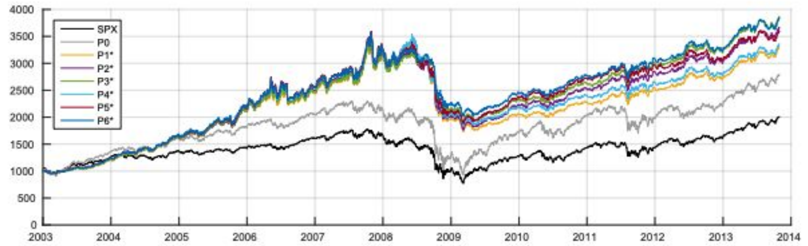
[Chatzilenna et. al. 2019]

Epidemics & Infectious diseases

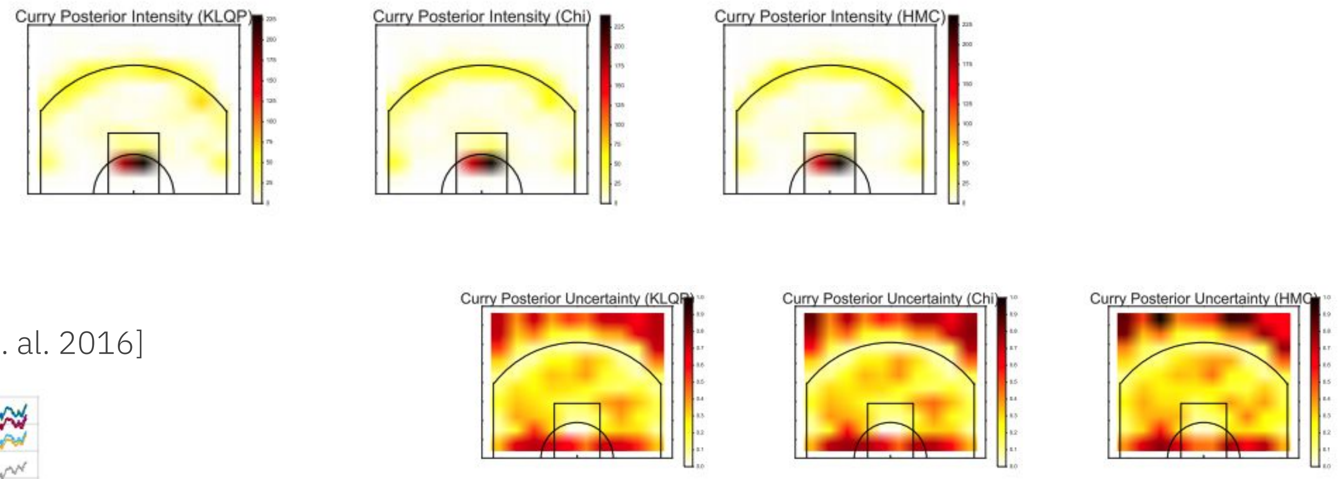


[Bamler et. al. 2017]

[Gruber et. al. 2016]



Stock Markets and Time-Series



[Dieng et. al. 2017]

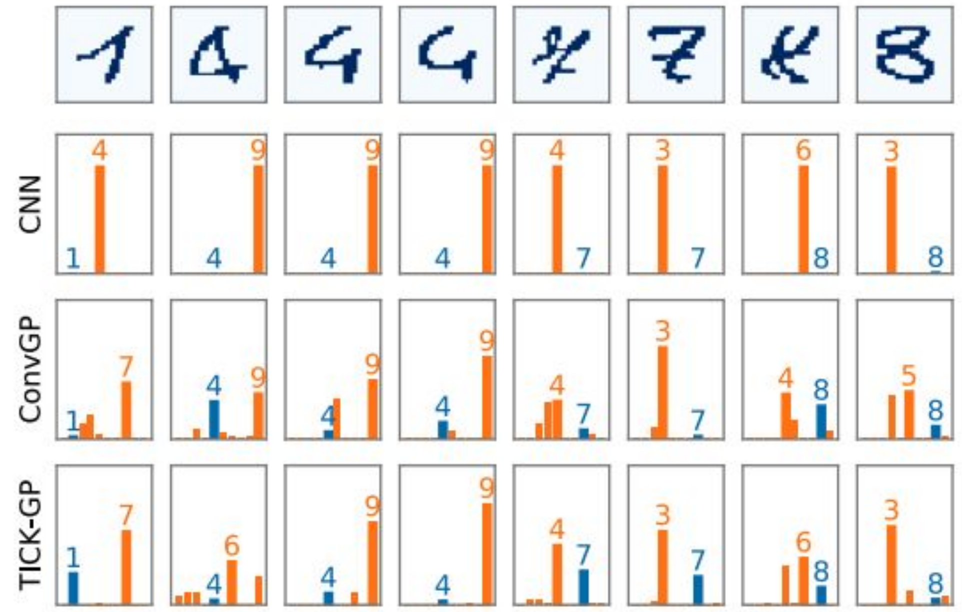
Sport analysis

... importantly, PM allows uncertainty estimation ...

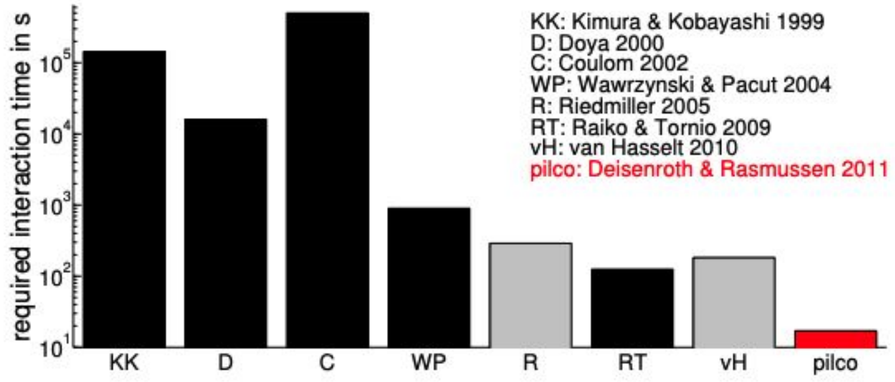
... conv network predictions

... bayesian model predictions

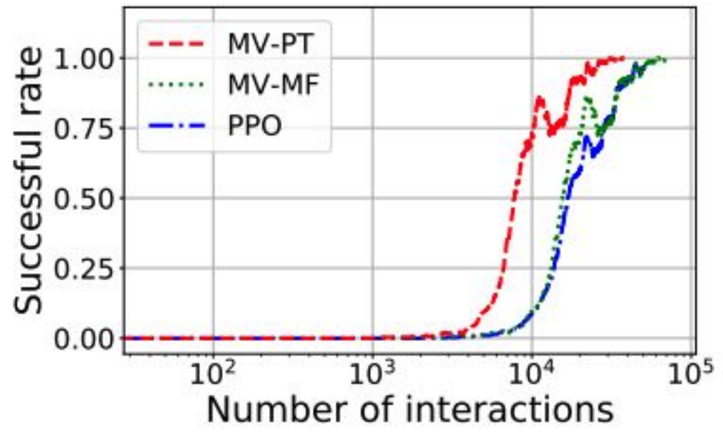
[Dutordoir et. al. 2019]

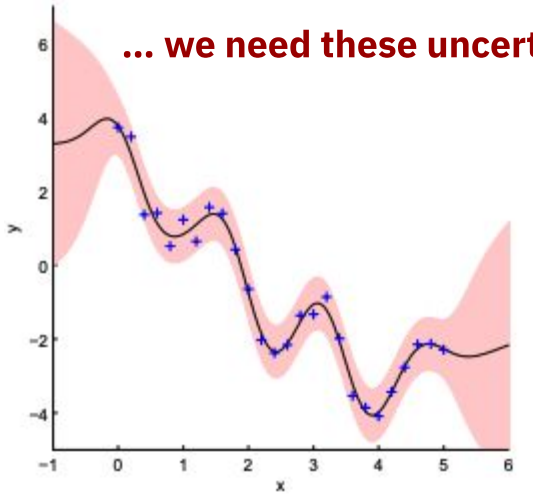


[Deisenroth et.al. 2011]



[Li et.al. 2019]





... we need these uncertainties ...

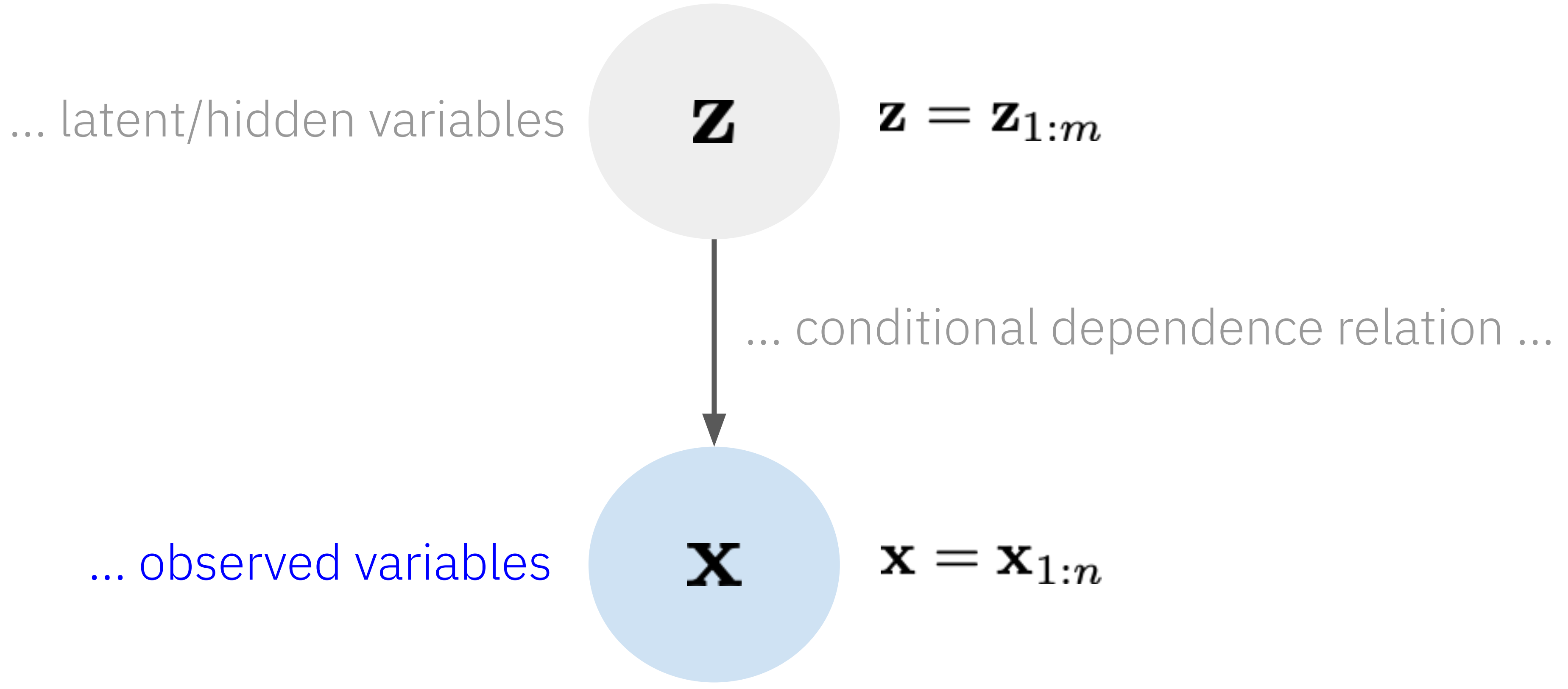
... I will be careful to using an RNN to invest my money ...

... can be useful for reinforcement learning as well

... we represent random variables using graphical models ...

Latent Variable Models allow:

- inferring about hidden information not observed in the data
- principled way to introduce our beliefs and priors on the phenomena we are trying to estimate



Our Goal:

$$\overbrace{\max \log p(\mathbf{x})}^{\text{... max marginal ...}} = \max \log \underbrace{\int p(\mathbf{x}, \mathbf{z}) d\mathbf{z}}_{\text{... marginalise latents ...}} = \max \log \boxed{\int p(\mathbf{x}|\mathbf{z}) p(\mathbf{z}) d\mathbf{z}}$$

... how to compute this integral ...

... let’s illustrate the difficulty ...

<u>Our Goal:</u>

... max marginal ...

$$\max \log p(\mathbf{x}) = \max \log \int p(\mathbf{x}, \mathbf{z}) d\mathbf{z} = \max \log \boxed{\int p(\mathbf{x}|\mathbf{z}) p(\mathbf{z}) d\mathbf{z}}$$

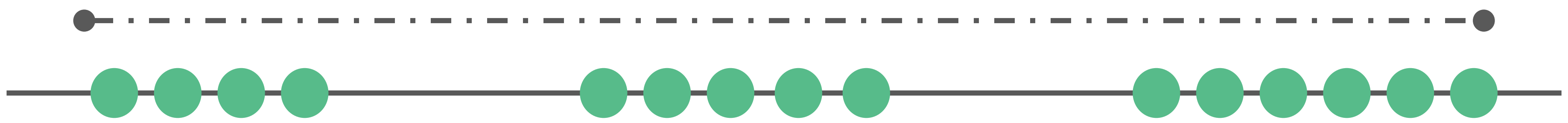
... how to compute this integral ...

... marginalise latents ...

<u>Let’s take an example of this, where we need to cluster some given data-set, we assume:</u>

- We know the total number of cluster
- We know that one data point belongs to one cluster

... we are given n data points ...



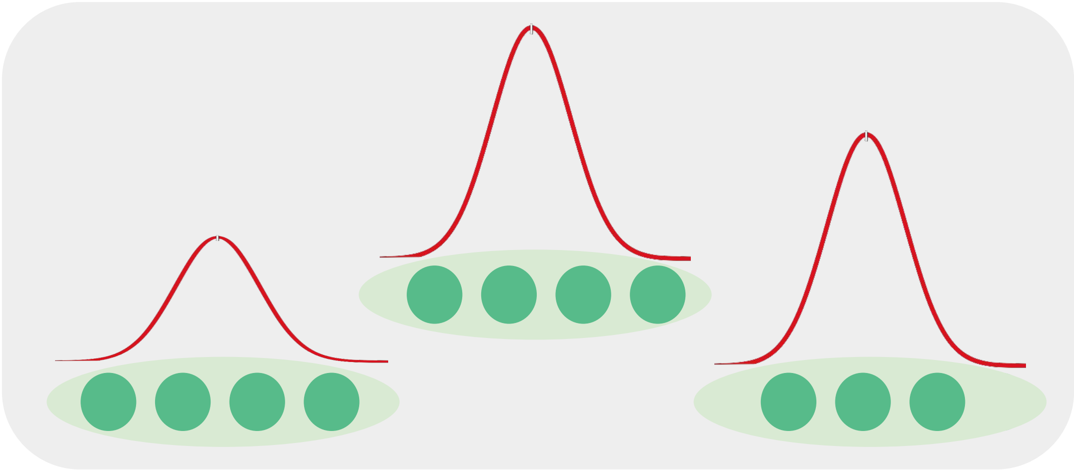
... have K clusters ...

... we want to think of **clustering** from a **<u>probabilistic modelling perspective</u>** ...

Bayesian mixture of Gaussians ...

To do so, we assume that there is some hidden/latent process by which this data has been generated, e.g.,

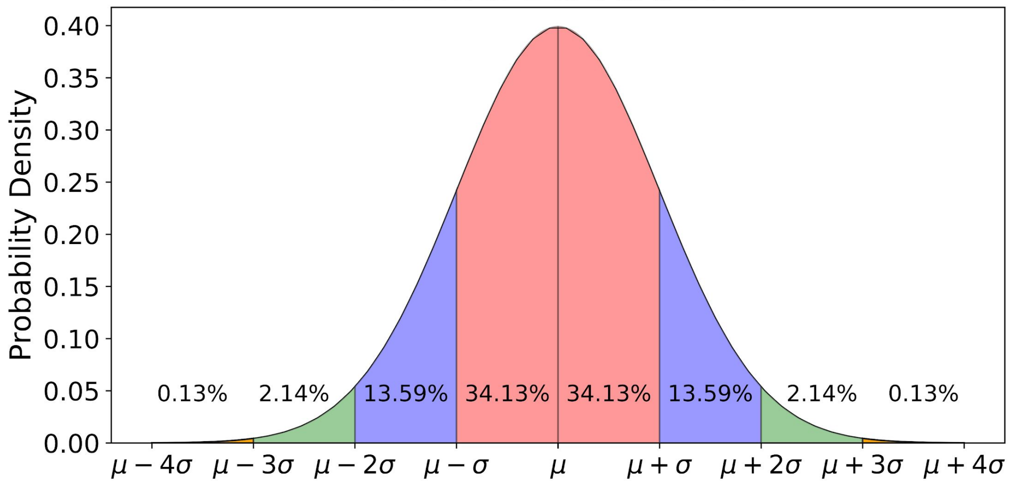
- Nature has access to the finite number of clusters
- Each cluster data has some distribution
- Nature samples a cluster and from that distribution samples a data-point that we saw in the data-set



Nature's hidden process

... **as such, to model nature's process, we need to be able to deal with two types of random variables:**

- ❏ Continuous Random Variables -- Gaussians (e.g., point locations)
- ❏ Discrete Random Variables -- Multinomials (e.g., clusters chosen)



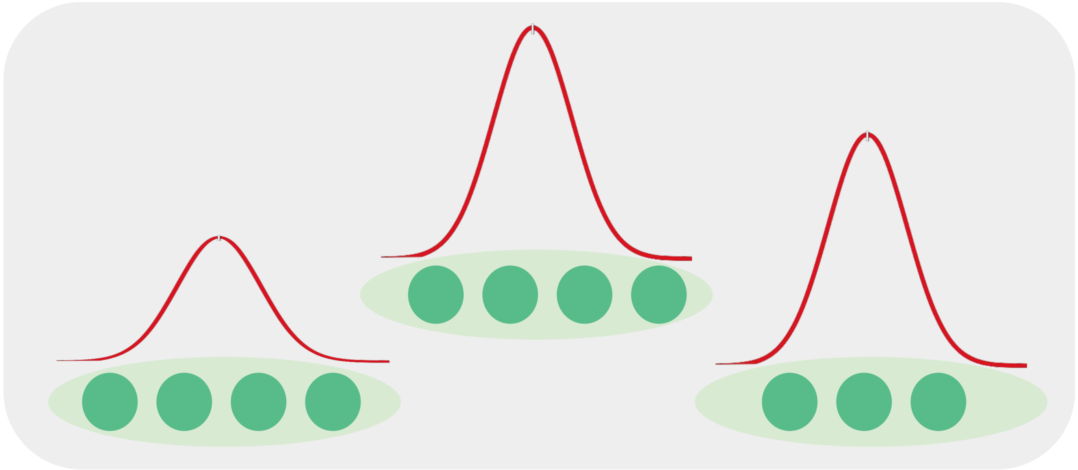
Gaussian Distribution

Categorical Distribution

... Bayesian mixture of Gaussians ...

To do so, we assume that there is some hidden/latent process by which this data has been generated, e.g.,

- Nature has access to the finite number of clusters
- Each cluster data has some distribution
- Nature samples a cluster and from that distribution samples a data-point that we saw in the data-set



Nature's hidden process

$$\mu_k \sim \mathcal{N}(0, \sigma^2) \quad k = 1, \ldots, K$$

... we think of cluster centers as coming from some Gaussian (real-valued random variable)

$$\mathbf{c}_i \sim \mathrm{Cat}\,(K) \quad \text{for } i = 1, \ldots, n$$

... we think of cluster choice as a one-hot vector

$$\boldsymbol{\mu} = [\mu_1, \ldots, \mu_K]^\mathsf{T}$$

... vector of all means

$$\mathbf{c}_i^\mathsf{T} \mathbf{x}_i = [0, 0, 1, \ldots, 0] \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_K \end{bmatrix} = \mu_3$$

... data is from a gaussian with some mean ...

$$\mathbf{x}_i | \mathbf{c}_i, \boldsymbol{\mu} \sim \mathcal{N}\left(\mathbf{c}_i^\mathsf{T} \boldsymbol{\mu}, \sigma_2^2\right) \quad i = 1, \ldots, n$$

... each data point is assigned to a cluster through an inner product

... one-hot vector

$$\mathbf{c}_i^\mathsf{T} = \underbrace{[0, 0, 1, \ldots 0]}_{K}$$

... Bayesian mixture of Gaussians ...

Our Goal:

$$\overbrace{\max \log p(\mathbf{x})}^{\text{... max marginal ...}} = \max \log \underbrace{\int p(\mathbf{x}, \mathbf{z}) d\mathbf{z}}_{\text{... marginalise latents ...}} = \max \log \boxed{\int p(\mathbf{x}|\mathbf{z}) p(\mathbf{z}) d\mathbf{z}}$$

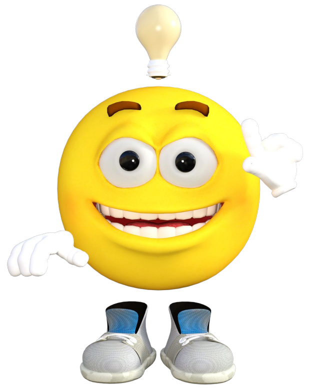
... how to compute this integral ...

Modelling Assumptions:

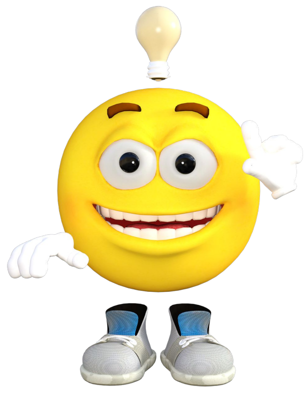
$$\mu_k \sim \mathcal{N}(0, \sigma^2) \quad k = 1, \ldots, K$$
$$\mathbf{c}_i \sim \text{Cat}\,(K) \quad \text{for } i = 1, \ldots, n$$
$$\mathbf{x}_i | \mathbf{c}_i, \boldsymbol{\mu} \sim \mathcal{N}\left(\mathbf{c}_i^\mathsf{T} \boldsymbol{\mu}, \sigma_2^2\right) \quad i = 1, \ldots, n$$

Our Goal is to Maximise:

$$\log p(\mathbf{x}) = \log \int_{\boldsymbol{\mu}} \sum_{\mathbf{c}} p(\mathbf{x}, \mathbf{c}, \boldsymbol{\mu}) d\boldsymbol{\mu}$$



... but what is this crazy joint distribution ...



... Bayesian mixture of Gaussians ...

Modelling Assumptions:

$$\mu_k \sim \mathcal{N}(0, \sigma^2) \quad k = 1, \dots, K$$

$$\mathbf{c}_i \sim \text{Cat}\,(K) \quad \text{for } i = 1, \dots, n$$

$$\mathbf{x}_i | \mathbf{c}_i, \boldsymbol{\mu} \sim \mathcal{N}\left(\mathbf{c}_i^\top \boldsymbol{\mu}, \sigma_2^2\right) \quad i = 1, \dots, n$$

Our Goal is to Maximise:

$$\log p(\mathbf{x}) = \log \int_{\boldsymbol{\mu}} \sum_{\mathbf{c}} p(\mathbf{x}, \mathbf{c}, \boldsymbol{\mu}) d\boldsymbol{\mu}$$

$$\log p(\mathbf{x}) = \log \int_{\boldsymbol{\mu}} \sum_{\mathbf{c}} p(\mathbf{x}, \mathbf{c}, \boldsymbol{\mu}) d\boldsymbol{\mu}$$

Chain Rule of Probability $p(A_n, \dots, A_1) = p(A_n | A_{n-1}, \dots, A_1) p(A_{n-1}, \dots, A_1)$

$$p(\mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{c}_1, \dots, \mathbf{c}_n, \mu_1, \dots, \mu_K) = p(\mathbf{x}_n | \mathbf{c}_n, \boldsymbol{\mu}) \dots p(\mathbf{x}_1 | \mathbf{c}_1, \boldsymbol{\mu}) p(\mathbf{c}_n) \dots p(\mathbf{c}_1) p(\boldsymbol{\mu}) = p(\boldsymbol{\mu}) \prod_{i=1}^{n} p(\mathbf{c}_i) p(\mathbf{x}_i | \mathbf{c}_i, \boldsymbol{\mu})$$

$$\max \log \boxed{\int p(\mathbf{x}|\mathbf{z}) p(\mathbf{z}) d\mathbf{z}}$$

... hard to compute $\log \int_{\boldsymbol{\mu}} p(\boldsymbol{\mu}) \prod_{i=1}^{n} \sum_{c_i} p(c_i) p(x_i | c_i, \boldsymbol{\mu}) d\boldsymbol{\mu}$

... variational inference helps us deal with these probs ...

... variational inference **transforms computing integrals to an optimisation problem** which we can solve using standard optimisation techniques, e.g., ADAM...

$$\log \int p(\mathbf{x},\mathbf{z})d\mathbf{z} = \log \int \frac{q(\mathbf{z})}{q(\mathbf{z})} p(\mathbf{x},\mathbf{z})d\mathbf{z} \;\; = \log \mathbb{E}_{q(\mathbf{z})}\left[\frac{p(\mathbf{x},\mathbf{z})}{q(\mathbf{z})}\right] \geq \mathbb{E}_{q(\mathbf{z})}\left[\log\left[\frac{p(\mathbf{x},\mathbf{z})}{q(\mathbf{z})}\right]\right]$$

$$= \mathbb{E}_{q(\mathbf{z})}\left[\log\left[\frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{q(\mathbf{z})}\right]\right]$$

$$= \mathbb{E}_{q(\mathbf{z})}\left[\log p(\mathbf{x}|\mathbf{z}) + \log \frac{p(\mathbf{z})}{q(\mathbf{z})}\right]$$

$$= \mathbb{E}_{q(\mathbf{z})}\left[\log p(\mathbf{x}|\mathbf{z})\right] + \mathbb{E}_{q(\mathbf{z})}\left[\log \frac{p(\mathbf{z})}{q(\mathbf{z})}\right]$$

$$= \mathbb{E}_{q(\mathbf{z})}\left[\log p(\mathbf{x}|\mathbf{z})\right] - \mathbb{E}_{q(\mathbf{z})}\left[\log \frac{q(\mathbf{z})}{p(\mathbf{z})}\right]$$

Evidence Lower-Bound (ELBO)

$$= \mathbb{E}_{q(\mathbf{z})}\underbrace{\left[\log p(\mathbf{x}|\mathbf{z})\right]}_{\text{likelihood}} - \underbrace{\mathrm{KL}\left(q(\mathbf{z})||p(\mathbf{z})\right)}_{\text{regulariser}}$$

... we can also write our ELBO as ...

.. another way to rewrite the ELBO is simply using the Bayes rule as ...

$$
\begin{aligned}
\mathbb{E}_{q(\mathbf{z})}\left[\log p(\mathbf{x}|\mathbf{z})\right] - \mathrm{KL}\left(q(\mathbf{z})||p(\mathbf{z})\right) &= \mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x}|\mathbf{z})] - \int q(\mathbf{z}) \log \frac{q(\mathbf{z})}{p(\mathbf{z})} d\mathbf{z} \\
&= \mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x}|\mathbf{z})] - \int q(\mathbf{z}) \log q(\mathbf{z}) d\mathbf{z} + \int q(\mathbf{z}) \log p(\mathbf{z}) d\mathbf{z} \\
&= \int q(\mathbf{z})[\log p(\mathbf{x}|\mathbf{z}) + \log p(\mathbf{z})] d\mathbf{z} - \int q(\mathbf{z}) \log q(\mathbf{z}) d\mathbf{z} \\
&= \int q(\mathbf{z}) \log p(\mathbf{x}|\mathbf{z}) p(\mathbf{z}) d\mathbf{z} - \int q(\mathbf{z}) \log q(\mathbf{z}) d\mathbf{z} \\
&= \int q(\mathbf{z}) \log p(\mathbf{x}, \mathbf{z}) d\mathbf{z} - \int q(\mathbf{z}) \log q(\mathbf{z}) d\mathbf{z} = \mathbb{E}_{q(\mathbf{z})}\left[\log p(\mathbf{x}, \mathbf{z})\right] + \mathcal{H}\left(q(\mathbf{z})\right)
\end{aligned}
$$

... applying Bayes Rule

$$p(A|B) = \frac{p(A, B)}{p(B)} \implies p(A, B) = p(A|B)p(B)$$

.. another way to write the ELBO

… to finalise the problem, we introduce family of dist’s …

… to fully specify the problem, we need an additional ingredient of the type of the variational distribution. For now, we assume a **mean-field variational distribution** that we write as …

$$q(\mathbf{z}) = \prod_{j=1}^{m} q_j(\mathbf{z}_j)$$

… each latent is covered by its own variational factor …

… a distribution over latent variables, trained to maximise ELBO …

Evidence Lower-Bound (ELBO)

$$\mathbb{E}_{q(\mathbf{z})}\left[\log p(\mathbf{x}|\mathbf{z})\right] - \mathrm{KL}\left(q(\mathbf{z})||p(\mathbf{z})\right)$$

… we use this distribution in the ELBO which we need to maximise …

… remember in our example, we have: $\mathbf{z} = \{\boldsymbol{\mu}, \mathbf{c}\} \Longrightarrow q(\mathbf{z}) = q(\boldsymbol{\mu}, \mathbf{c}) = q(\boldsymbol{\mu})q(\mathbf{c}) = q(\mu_1, \dots, \mu_k)q(\mathbf{c}_1, \dots, \mathbf{c}_n)$

… mean-field variational family … $= \underbrace{\prod_{k=1}^{K} q(\mu_k; \phi_k^{(\text{var-}\mu)})}_{\text{parameterised Gaussian}} \underbrace{\prod_{i=1}^{n} q(\mathbf{c}_i; \phi_i^{(\text{var-}c)})}_{\text{parameterised multinomial}}$

... back to our example ...

... let’s write the overall problem that we need to solve for our Bayesian mixture of Gaussians, when using variational inference ...

$$\int q(\mathbf{z}) \log p(\mathbf{x}, \mathbf{z}) d\mathbf{z} - \int q(\mathbf{z}) \log q(\mathbf{z}) d\mathbf{z}$$

... we need to understand what happens with this optimisation problem ...

$$\begin{aligned}
\int q(\mathbf{z}) \log p(\mathbf{x}, \mathbf{z}) d\mathbf{z} &= \int \sum_{\mathbf{c}} q(\boldsymbol{\mu}, \mathbf{c}) \log p(\mathbf{x}, \mathbf{c}, \boldsymbol{\mu}) d\boldsymbol{\mu} \\
&= \int \sum_{\mathbf{c}} q(\boldsymbol{\mu}) q(\mathbf{c}) \log p(\mathbf{x}, \mathbf{c}, \boldsymbol{\mu}) d\boldsymbol{\mu} \\
&= \int \sum_{\mathbf{c}} q(\boldsymbol{\mu}) q(\mathbf{c}) \log p(\boldsymbol{\mu}) \prod_i p(\mathbf{c}_i) p(\mathbf{x}_i | \mathbf{c}_i \boldsymbol{\mu}) d\boldsymbol{\mu} \\
&= \mathbb{E}_{q(\boldsymbol{\mu})}[\log p(\boldsymbol{\mu})] + \mathbb{E}_{q(\mathbf{c})} \left[\sum_i \log p(\mathbf{c}_i) \right] + \mathbb{E}_{q(\boldsymbol{\mu}) q(\mathbf{c})} \left[\sum_i \log p(\mathbf{x}_i | \mathbf{c}_i, \boldsymbol{\mu}) \right]
\end{aligned}$$

... back to our example ...

$$\mathbb{E}_{q(\boldsymbol{\mu})}[\log p(\boldsymbol{\mu})] + \mathbb{E}_{q(\mathbf{c})}\left[\sum_i \log p(\mathbf{c}_i)\right] + \mathbb{E}_{q(\boldsymbol{\mu})q(\mathbf{c})}\left[\sum_i \log p(\mathbf{x}_i|\mathbf{c}_i, \boldsymbol{\mu})\right]$$

$$p(\mathbf{c}_i) = \frac{1}{K}$$

prior assumption

$$\log p(\mathbf{x}_i|\mathbf{c}_i, \boldsymbol{\mu}) = \log \frac{1}{\sqrt{2\pi\sigma_2^2}} - \frac{1}{2\sigma_2^2}\left(\mathbf{x}_i - \mathbf{c}_i^\top \boldsymbol{\mu}\right)^2$$

$$\begin{aligned}\mathbf{x}_i^2 - 2\mathbf{c}_i^\top \boldsymbol{\mu}\mathbf{x}_i + (\mathbf{c}_i^\top \boldsymbol{\mu})^2 &= \mathbf{x}_i^2 - 2\sum_{j=1}^{K} \mathbf{c}_{i,j}\boldsymbol{\mu}_j \mathbf{x}_i + (\mathbf{c}_i^\top \boldsymbol{\mu})(\mathbf{c}_i^\top \boldsymbol{\mu}) \\ &= \mathbf{x}_i^2 - 2\sum_{j=1}^{K} \mathbf{c}_{i,j}\boldsymbol{\mu}_j \mathbf{x}_i + \sum_{j=1}^{K} \mathbf{c}_{i,j}\mu_j^2 \\ &= \sum_{j=1}^{K} \mathbf{c}_{i,j}(\mathbf{x}_i - \mu_j)^2\end{aligned}$$

$$= \log \frac{1}{\sqrt{2\pi\sigma_2^2}} - \frac{1}{2\sigma_2^2}\sum_{j=1}^{K} \mathbf{c}_{i,j}(\mathbf{x}_i - \mu_j)^2$$

$$\log p(\mu_1, \dots, \mu_K) = \log \prod_{j=1}^{K} p(\mu_j) = \sum_{j=1}^{K} \log p(\mu_j) = \sum_{j=1}^{K} \log\left[\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}\mu_j^2\right)\right] = \sum_{j=1}^{K} \log \frac{1}{\sqrt{2\pi\sigma^2}} - \sum_{j=1}^{K} \frac{1}{2\sigma^2}\mu_j^2 = -\frac{K}{2}\log 2\pi\sigma^2 - \sum_{j=1}^{K} \frac{1}{2\sigma^2}\mu_j^2$$

... back to our ELBO ...

$$\int q(\mathbf{z}) \log p(\mathbf{x}, \mathbf{z}) d\mathbf{z} - \int q(\mathbf{z}) \log q(\mathbf{z}) d\mathbf{z}$$

... now, this term ...

$$\mathbb{E}_{q(\boldsymbol{\mu})}[\log p(\boldsymbol{\mu})] + \mathbb{E}_{q(\mathbf{c})}\left[\sum_i \log p(\mathbf{c}_i)\right] + \mathbb{E}_{q(\boldsymbol{\mu})q(\mathbf{c})}\left[\sum_i \log p(\mathbf{x}_i | \mathbf{c}_i, \boldsymbol{\mu})\right]$$

$$= \mathbb{E}_{q(\boldsymbol{\mu})}\left[-\frac{K}{2}\log 2\pi\sigma^2 - \sum_{j=1}^{K} \frac{1}{2\sigma^2}\mu_j^2\right] - N \log K + \mathbb{E}_{q(\boldsymbol{\mu})q(\mathbf{c})}\left[-\frac{N}{2}\log 2\pi\sigma_2^2 - \frac{1}{2\sigma_2^2}\sum_i \sum_{j=1}^{K} c_{i,j}\left(\boldsymbol{x}_i - \mu_j\right)^2\right]$$

$$\begin{aligned}
\int q(\mathbf{z}) \log q(\mathbf{z}) d\mathbf{z} = \int \sum_{\mathbf{c}} q(\boldsymbol{\mu}) q(\mathbf{c}) \log q(\boldsymbol{\mu}) q(\mathbf{c}) &= \int \sum_{\mathbf{c}} q(\boldsymbol{\mu}) q(\mathbf{c}) \left[\log q(\boldsymbol{\mu}) + \log q(\mathbf{c})\right] d\boldsymbol{\mu} \\
&= \int q(\boldsymbol{\mu}) \log q(\boldsymbol{\mu}) d\boldsymbol{\mu} + \sum_{\mathbf{c}} q(\mathbf{c}) \log q(\mathbf{c}) \\
&= \mathbb{E}_{q(\boldsymbol{\mu})}\left[\log q(\boldsymbol{\mu})\right] + \mathbb{E}_{q(\mathbf{c})}\left[\log q(\mathbf{c})\right] \\
&= \mathbb{E}_{q(\boldsymbol{\mu})}\left[\sum_{j=1}^{K} \log q_j(\mu_j)\right] + \mathbb{E}_{q(\mathbf{c})}\left[\sum_{i=1}^{n} \log q_i(\mathbf{c}_i)\right]
\end{aligned}$$

... back to our ELBO ...

$$\text{ELBO}(\cdot) = \mathbb{E}_{q(\boldsymbol{\mu})}\left[-\frac{K}{2}\log 2\pi\sigma^2 - \sum_{j=1}^{K}\frac{1}{2\sigma^2}\mu_j^2\right] - N\log K + \mathbb{E}_{q(\boldsymbol{\mu})q(\boldsymbol{c})}\left[-\frac{N}{2}\log 2\pi\sigma_2^2 - \frac{1}{2\sigma_2^2}\sum_i\sum_{j=1}^{K}c_{i,j}\left(\boldsymbol{x}_i-\mu_j\right)^2\right]$$

$$-\mathbb{E}_{q(\boldsymbol{\mu})}\left[\sum_{j=1}^{K}\log q_j(\mu_j)\right] - \mathbb{E}_{q(\boldsymbol{c})}\left[\sum_{i=1}^{n}\log q_i(\boldsymbol{c}_i)\right]$$

... remember our mean-field approximation: $q(\boldsymbol{\mu})q(\mathbf{c}) = \prod_{j=1}^{K}q_j(\mu_j)\prod_{i=1}^{n}q_i(\mathbf{c}_i)$

$$= -\frac{K}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}\sum_{j=1}^{K}\mathbb{E}_{q_j(\mu_j)}\left[\mu_j^2\right] - n\log K - \frac{n}{2}\log 2\pi\sigma_2^2 - \frac{1}{2\sigma_2^2}\sum_{i=1}^{n}\sum_{j=1}^{K}\mathbb{E}_{q_i(\mathbf{c}_i)}\left[c_{i,j}\right]\mathbb{E}_{q_j(\mu_j)}\left[(\mathbf{x}_i-\mu_j)^2\right]$$

$$-\sum_{j=1}^{K}\mathbb{E}_{q_j(\mu_j)}\left[\log q_j(\mu_j)\right] - \sum_{i=1}^{n}\mathbb{E}_{q_i(\boldsymbol{c}_i)}\left[\log q_i(\boldsymbol{c}_i)\right]$$

Variational Assumptions:

$$q_j(\mu_j) = \mathcal{N}\left(\phi_{j,1}^{(\text{var}-\mu)}, \phi_{j,2}^{(\text{var}-\sigma)}\right), \quad j = 1,\dots,K$$

$$q_i\left(\boldsymbol{c}_i = \underbrace{[0,0,\dots,1,\dots,0]}_{\text{one at } j^{th} \text{ position}}\right) = \phi_{i,j}^{(\text{var-c})}, \quad i = 1,\dots,n.$$

... back to our ELBO ...

$$\mathrm{ELBO}(\cdot) = -\frac{K}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}\sum_{j=1}^{K}\mathbb{E}_{q_j(\mu_j)}\left[\mu_j^2\right] - n\log K - \frac{n}{2}\log 2\pi\sigma_2^2 - \frac{1}{2\sigma_2^2}\sum_{i=1}^{n}\sum_{j=1}^{K}\mathbb{E}_{q_i(\mathbf{c}_i)}\left[c_{i,j}\right]\mathbb{E}_{q_j(\mu_j)}\left[(\mathbf{x}_i-\mu_j)^2\right]$$

$$-\sum_{j=1}^{K}\mathbb{E}_{q_j(\mu_j)}\left[\log q_j(\mu_j)\right] - \sum_{i=1}^{n}\mathbb{E}_{q_i(\mathbf{c}_i)}\left[\log q_i(\mathbf{c}_i)\right]$$

Variational Assumptions:

$$q_j(\mu_j) = \mathcal{N}\left(\phi_{j,1}^{(\text{var}-\mu)}, \phi_{j,2}^{(\text{var}-\sigma)}\right), \qquad j = 1,\dots,K$$

$$q_i\left(\boldsymbol{c}_i = \underbrace{[0,0,\dots,1,\dots,0]}_{\text{one at } j^{th} \text{ position}}\right) = \phi_{i,j}^{(\text{var-c})}, \qquad i = 1,\dots,n.$$

$$\mathbb{E}_{q_j(\mu_j)}[\mu_j^2] = \phi_{j,2}^{(\text{var - }\sigma)} + \left(\phi_{j,1}^{(\text{var-}\mu)}\right)^2$$

$$\mathbb{E}_{q_i(\mathbf{c}_i)}\left[c_{i,j}\right]\mathbb{E}_{q_j(\mu_j)}\left[(\mathbf{x}_i-\mu_j)^2\right] = \phi_{i,j}^{(\text{var-c})}\,\mathbb{E}_{q_j(\mu_j)}\left[(\mathbf{x}_i-\mu_j)^2\right] = \phi_{i,j}^{(\text{var -c})}\left(\mathbf{x}_i^2 - 2\mathbf{x}_i\phi_{j,1}^{(\text{var-}\mu)} + \phi_{j,2}^{(\text{var- }\sigma)} + \left(\phi_{j,1}^{(\text{var- }\sigma)}\right)^2\right)$$

$$-\mathbb{E}_{q_j(\mu_j)}\left[\log q_j(\mu_j)\right] = \mathcal{H}(q_j(\mu_j)) = \log\left(\sqrt{\phi_{j,2}^{(\text{var - }\sigma)}2\pi e}\right)$$

$$\mathbb{E}_{q_i(\mathbf{c}_i)}\left[\log q_i(\mathbf{c}_i)\right] = \sum_{j=1}^{K}\phi_{i,j}^{(\text{var-}\sigma)}\log\phi_{i,j}^{(\text{var-}\sigma)}$$

... back to our ELBO ...

$$\mathrm{ELBO}(\cdot) = -\frac{K}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}\sum_{j=1}^{K}\mathbb{E}_{q_j(\mu_j)}\left[\mu_j^2\right] - n\log K - \frac{n}{2}\log 2\pi\sigma_2^2 - \frac{1}{2\sigma_2^2}\sum_{i=1}^{n}\sum_{j=1}^{K}\mathbb{E}_{q_i(\mathbf{c}_i)}\left[c_{i,j}\right]\mathbb{E}_{q_j(\mu_j)}\left[(\mathbf{x}_i-\mu_j)^2\right]$$

$$-\sum_{j=1}^{K}\mathbb{E}_{q_j(\mu_j)}\left[\log q_j(\mu_j)\right] - \sum_{i=1}^{n}\mathbb{E}_{q_i(\boldsymbol{c}_i)}\left[\log q_i(\boldsymbol{c}_i)\right]$$

$$= -\frac{K}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}\sum_{j=1}^{K}\ \phi_{j,2}^{(\text{var - }\sigma)} + \left(\phi_{j,1}^{(\text{var-}\mu)}\right)^2 - n\log K - \frac{n}{2}\log 2\pi\sigma_2^2 - \frac{1}{2\sigma_2^2}\sum_{i=1}^{n}\sum_{j=1}^{K}\ \phi_{i,j}^{(\text{var -c})}\left(\mathbf{x}_i^2 - 2\mathbf{x}_i\ \phi_{j,1}^{(\text{var-}\mu)} + \left(\phi_{j,1}^{(\text{var-}\mu)}\right)^2\right)$$

$$+\sum_{j=1}^{K}\log\left(\sqrt{\phi_{j,2}^{(\text{var - }\sigma)}2\pi e}\right) - \sum_{j=1}^{K}\phi_{i,j}^{(\text{var-}\sigma)}\log\phi_{i,j}^{(\text{var-}\sigma)}$$

... take the gradients ...

$$\mathrm{ELBO}(\cdot) = -\frac{K}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}\sum_{j=1}^{K}\phi_{j,2}^{(\text{var - }\sigma)} + \left(\phi_{j,1}^{(\text{var-}\mu)}\right)^2 - n\log K - \frac{n}{2}\log 2\pi\sigma_2^2 - \frac{1}{2\sigma_2^2}\sum_{i=1}^{n}\sum_{j=1}^{K}\phi_{i,j}^{(\text{var -c})}\left(\mathbf{x}_i^2 - 2\mathbf{x}_i\phi_{j,1}^{(\text{var-}\mu)} + \phi_{j,2}^{(\text{var- }\sigma)} + \left(\phi_{j,1}^{(\text{var-}\mu)}\right)^2\right)$$

$$+\sum_{j=1}^{K}\log\left(\sqrt{\phi_{j,2}^{(\text{var - }\sigma)}2\pi e}\right) - \sum_{j=1}^{K}\phi_{i,j}^{(\text{var-}\sigma)}\log\phi_{i,j}^{(\text{var-}\sigma)}$$

$$\nabla_{\phi_{i,j}^{(\text{var - c})}}\mathrm{ELBO}(\cdot) = -\frac{1}{2\sigma_2^2}\left(\mathbf{x}_i^2 - 2\mathbf{x}_i\phi_{j,1}^{(\text{var-}\mu)} + \phi_{j,2}^{(\text{var- }\sigma)} + \left(\phi_{j,1}^{(\text{var-}\mu)}\right)^2\right) - 1 - \log\phi_{i,j}^{(\text{var-}\sigma)} = 0$$

... we still need to normalise across j ... $\phi_{i,j}^{(\text{var - }\sigma)} \propto \exp\left(-\frac{1}{2\sigma_2^2}\left(\mathbf{x}_i^2 - 2\mathbf{x}_i\phi_{j,1}^{(\text{var-}\mu)} + \phi_{j,2}^{(\text{var- }\sigma)} + \left(\phi_{j,1}^{(\text{var-}\mu)}\right)^2\right)\right)$

$$\nabla_{\phi_{j,1}^{(\text{var-}\mu)}}\mathrm{ELBO}() = -\frac{1}{\sigma^2}\phi_{1,j}^{(\text{var-}\mu)} + \frac{1}{\sigma_2^2}\sum_{i=1}^{n}\phi_{i,j}^{(\text{var-c})}\left(\boldsymbol{x}_i - \phi_{j,1}^{(\text{var-}\mu)}\right) = 0$$

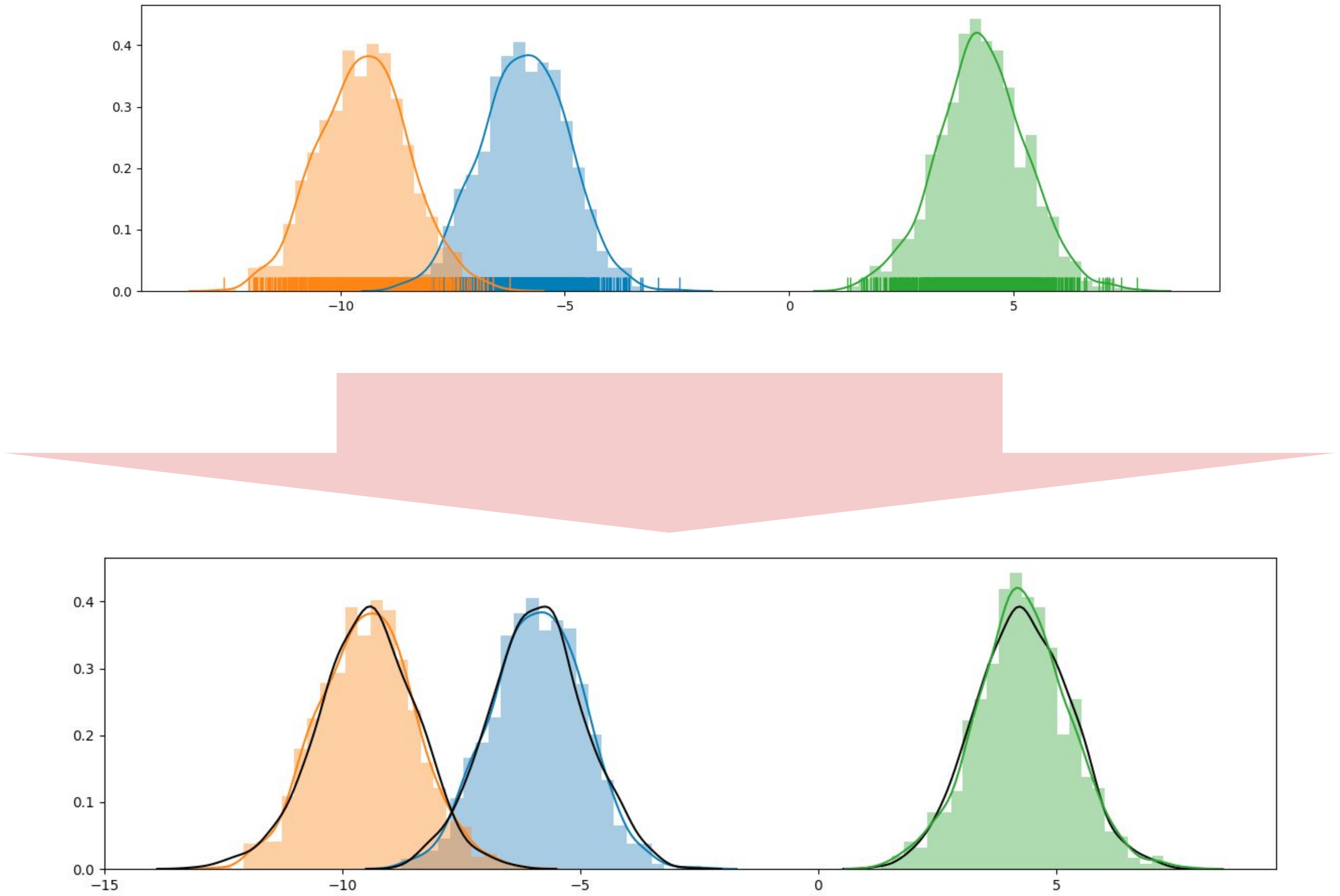
$$\phi_{j,1}^{(\text{var-}\mu)} = \frac{\frac{1}{\sigma_2^2}\sum_{i=1}^{n}\phi_{i,j}^{(\text{var-c})}\boldsymbol{x}_i}{\frac{1}{\sigma^2} + \frac{1}{\sigma_2^2}\sum_{i=1}^{n}\phi_{i,j}^{(\text{var-c})}}$$

$$\nabla_{\phi_{j,2}^{(\text{var-}\sigma)}}\mathrm{ELBO}(\cdot) = -\frac{1}{2\sigma^2} - \frac{1}{2\sigma_2^2}\sum_{i=1}^{n}\phi_{i,j}^{(\text{var-c})} + \frac{1}{4\phi_{i,j}^{(\text{var-}\sigma)}\pi e} = 0$$

$$\phi_{i,j}^{(\text{var-}\sigma)} = \frac{1}{2\pi e\left[\frac{1}{\sigma^2} + \frac{1}{\sigma_2^2}\sum_{i=1}^{n}\phi_{i,j}^{(\text{var-c})}\right]}$$

… example …

```python
def get_elbo(self):
    t1 = np.log(self.s2) - self.m/self.sigma2
    t1 = t1.sum()
    t2 = -0.5*np.add.outer(self.X**2, self.s2+self.m**2)
    t2 += np.outer(self.X, self.m)
    t2 -= np.log(self.phi)
    t2 *= self.phi
    t2 = t2.sum()
    return t1 + t2

def fit(self, max_iter=4000, tol=1e-10):
    self._init()
    self.elbo_values = [self.get_elbo()]
    self.m_history = [self.m]
    self.s2_history = [self.s2]
    for iter_ in range(1, max_iter+1):
        self._cavi()
        self.m_history.append(self.m)
        self.s2_history.append(self.s2)
        self.elbo_values.append(self.get_elbo())
        if iter_ % 5 == 0:
            print(iter_, self.m_history[iter_])
        if np.abs(self.elbo_values[-2] - self.elbo_values[-1]) <= tol:
            print('ELBO converged with ll %.3f at iteration %d'%(self.elbo_values[-1],
                                                                 iter_))
            break

    if iter_ == max_iter:
        print('ELBO ended with ll %.3f'%(self.elbo_values[-1]))


def _cavi(self):
    self._update_phi()
    self._update_mu()

def _update_phi(self):
    t1 = np.outer(self.X, self.m)
    t2 = -(0.5*self.m**2 + 0.5*self.s2)
    exponent = t1 + t2[np.newaxis, :]
    self.phi = np.exp(exponent)
    self.phi = self.phi / self.phi.sum(1)[:, np.newaxis]

def _update_mu(self):
    self.m = (self.phi*self.X[:, np.newaxis]).sum(0) * (1/self.sigma2 + self.phi.sum(0))**(-1)
    assert self.m.size == self.K
    #print(self.m)
    self.s2 = (1/self.sigma2 + self.phi.sum(0))**(-1)
    assert self.s2.size == self.K
```

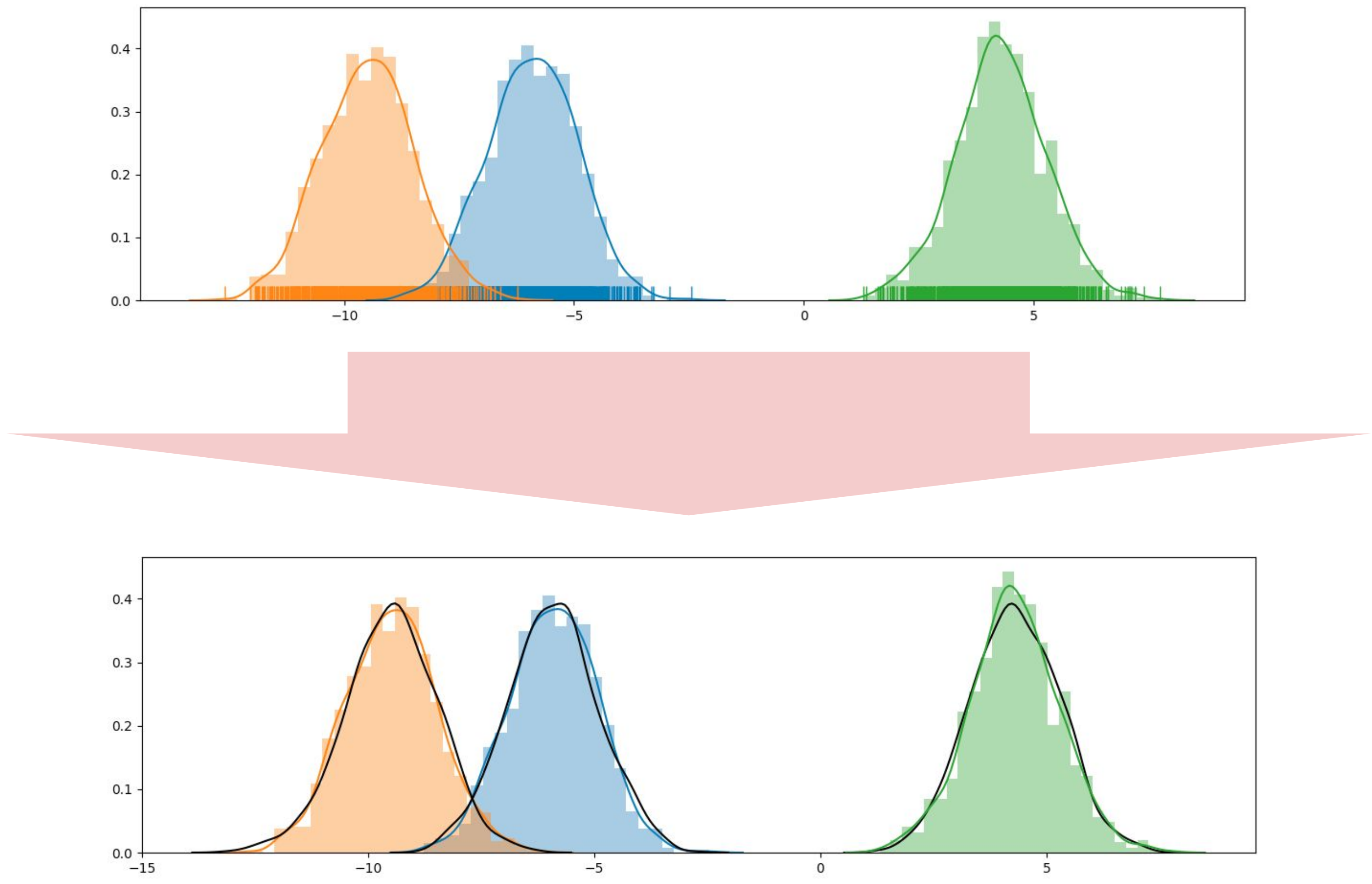


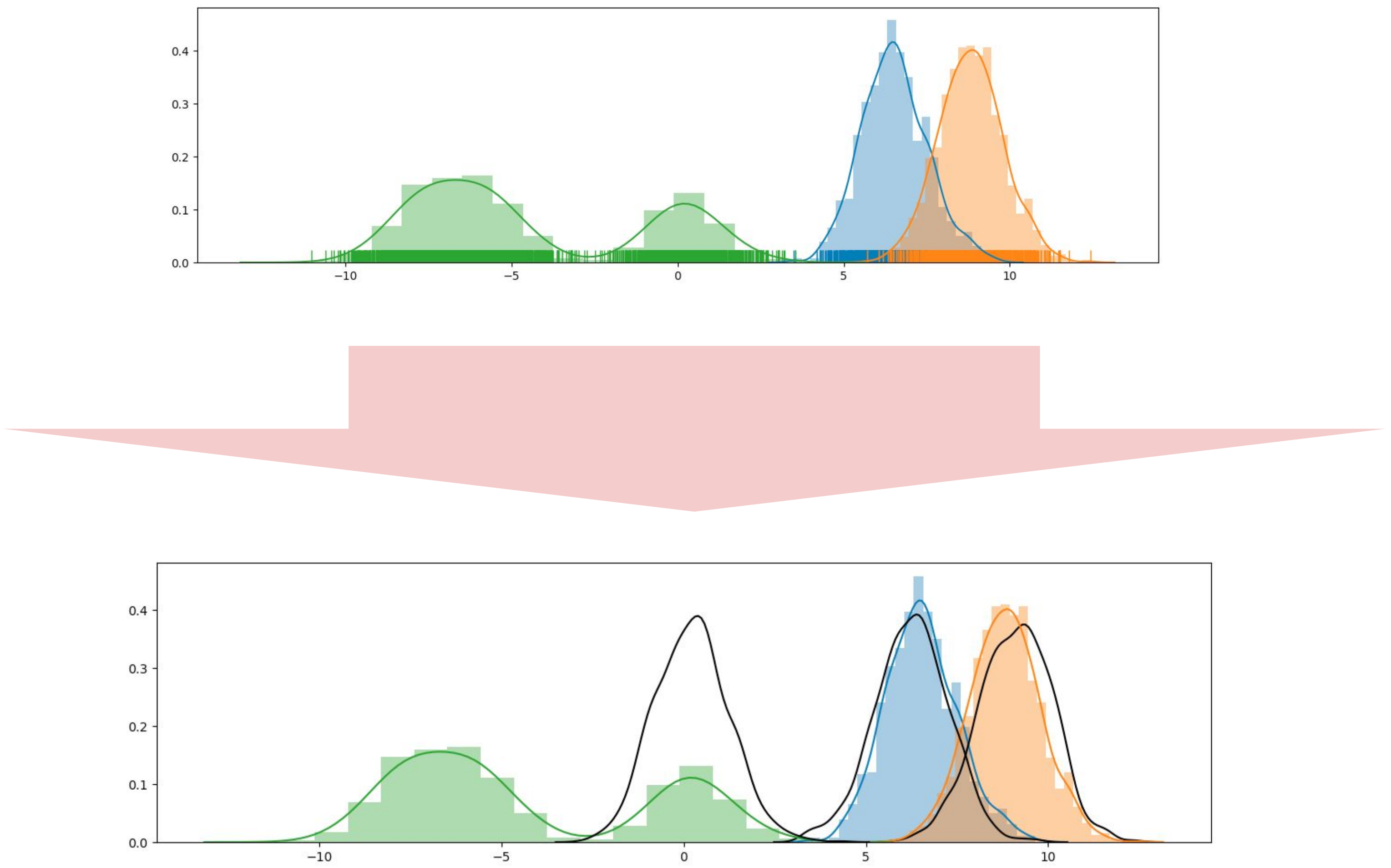
https://zhiyzuo.github.io/VI/#coordinate-ascent-vi-cavi

Zhiya Zuo
Filet-O-Fish 🍔 is the BEST!

Tweet Share

… example …





https://zhiyzuo.github.io/VI/#coordinate-ascent-vi-cavi

Zhiya Zuo

Filet-O-Fish 🍔 is the BEST!

Tweet Share

… code & resources …

Variational Inference: A Review for Statisticians

David M. Blei
Department of Computer Science and Statistics
Columbia University

Alp Kucukelbir
Department of Computer Science
Columbia University

Jon D. McAuliffe
Department of Statistics
University of California, Berkeley

May 11, 2018

17 commits | 1 branch | 0 packages | 0 releases | 3 contributors | MIT

Branch: master | New pull request | Find file | Clone or download

ldeecke Merge pull request #9 from madrugado/patch-1 … Latest commit a114780 10 days ago

LICENSE.md	Add license	4 months ago
README.md	added example	2 years ago
example.png	fixed example	2 years ago
example.py	Typos	13 months ago
gmm.py	ENH: predict_proba method	24 days ago
test.py	separated tests for CPU/GPU, formatting	2 years ago

https://github.com/ldeecke/gmm-torch

scikit learn Install User Guide API Examples More Go

Prev Up Next

scikit-learn 0.22.1
Other versions

Please **cite us** if you use the software.

2.1. Gaussian mixture models

`sklearn.mixture` is a package which enables one to learn Gaussian Mixture Models (diagonal, spherical, tied and full covariance matrices supported), sample them, and estimate them from data. Facilities to help determine the appropriate number of components are also provided.

https://scikit-learn.org/stable/modules/mixture.html

https://zhiyzuo.github.io/VI/#coordinate-ascent-vi-cavi

Zhiya Zuo
Filet-O-Fish 🍔 is the BEST!

Tweet Share

Thanks!