

Machine Learning and the Physical World

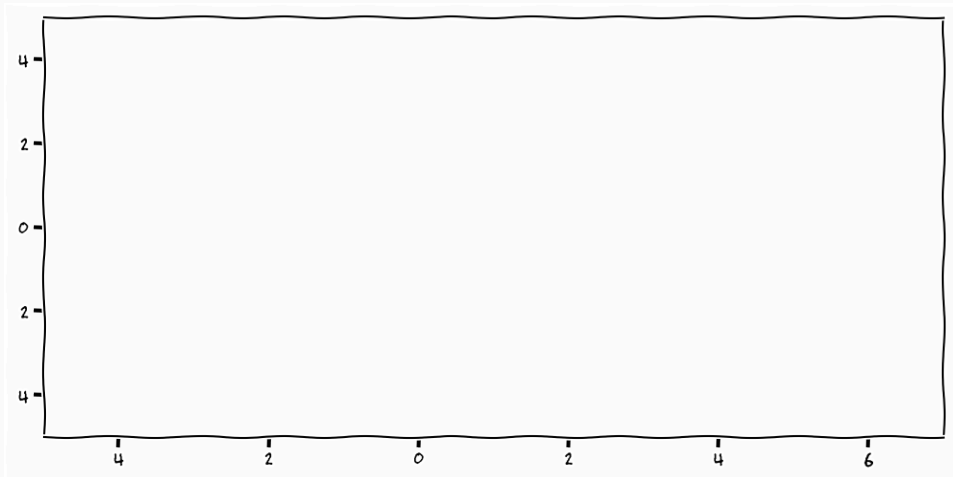
Lecture 5 : Practical Gaussian Processes

Carl Henrik Ek - che29@cam.ac.uk

26th of October, 2026

<http://carlhenrik.com>

Functions

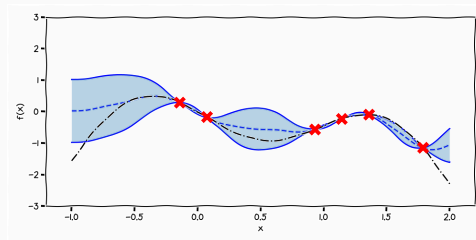
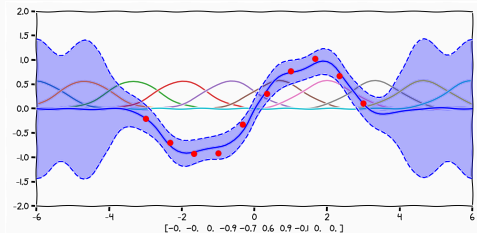


Parametrics vs. Non-parametrics

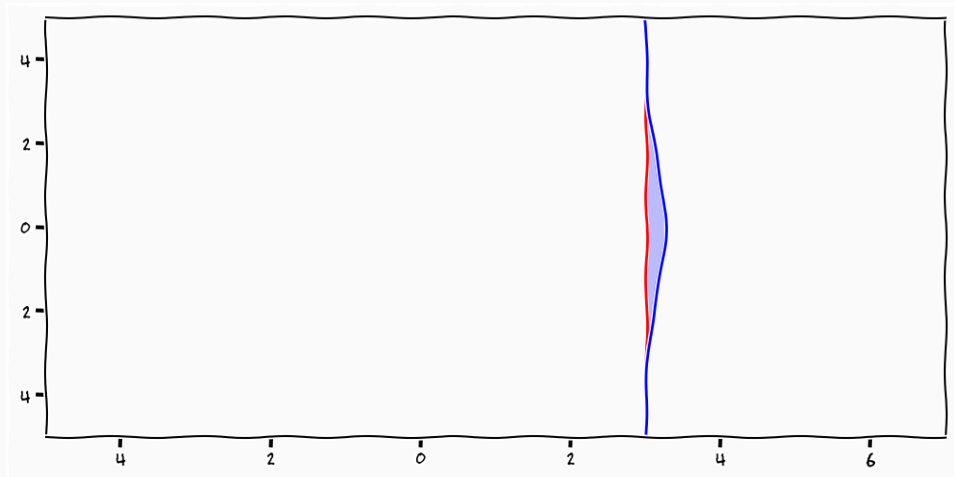
$$y_i = \underbrace{\mathbf{w}^T}_{\text{random}} \phi(\mathbf{x}_i) + \epsilon_i$$

$$y_i = \underbrace{f_i}_{\text{random}} + \epsilon_i$$

Non-parametrics vs Parametrics



Gaussian Processes



$$p(\mathbf{f}) = \mathcal{N} \left(\begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \\ \vdots \end{bmatrix} \middle| \begin{bmatrix} \mu(x_1) \\ \mu(x_2) \\ \vdots \\ \mu(x_N) \\ \vdots \end{bmatrix}, \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_N) & \dots \\ k(x_2, x_1) & k(x_2, x_2) & \dots & k(x_2, x_N) & \dots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ k(x_N, x_1) & k(x_N, x_2) & \dots & k(x_N, x_N) & \dots \\ \vdots & \vdots & \dots & \vdots & \ddots \end{bmatrix} \right)$$

- Co-variance and mean-function both have parameters ¹

¹I will usually just refer to them as θ

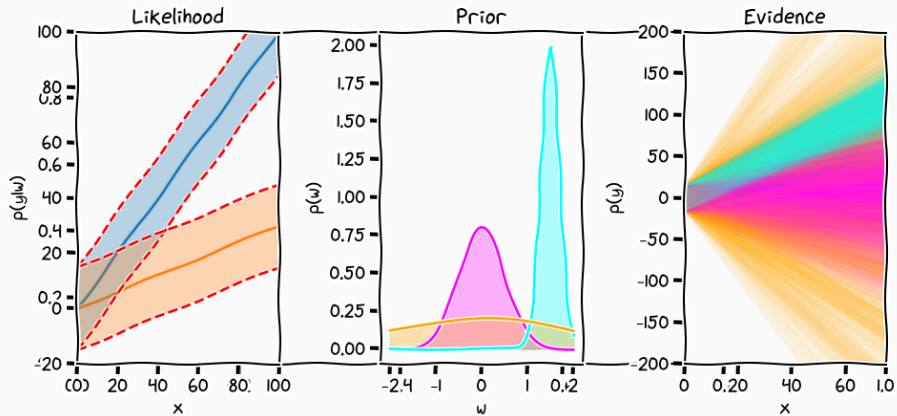
$$p(f \mid y) = \frac{p(y \mid f)p(f)}{\underbrace{\int p(y \mid f)p(f)df}_{p(y)}}$$

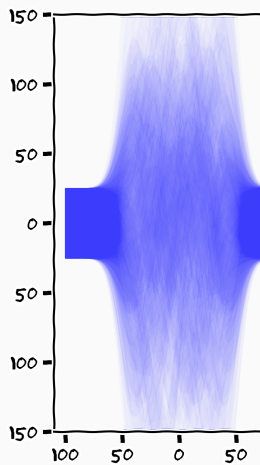
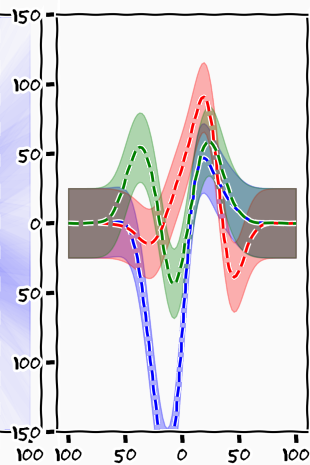
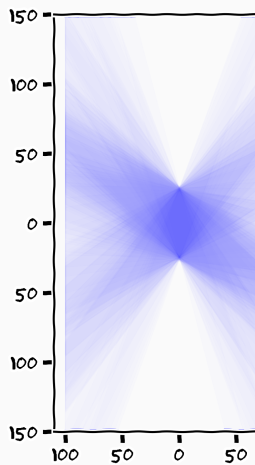
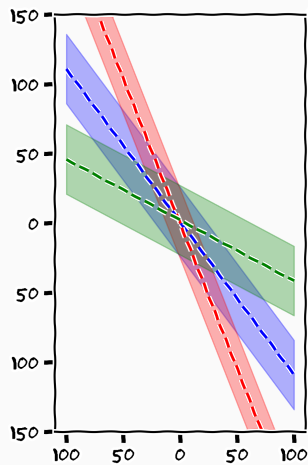
Likelihood How much **evidence** is there in the data for a specific hypothesis

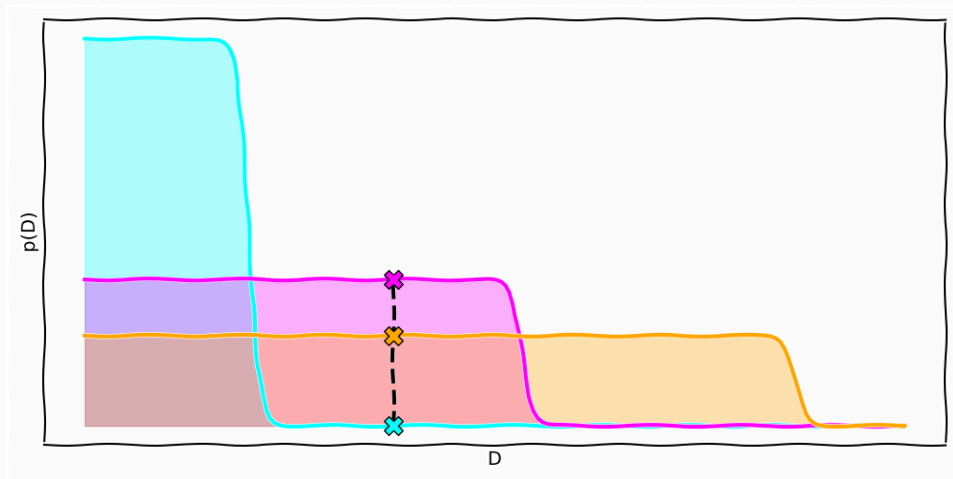
Prior What are my beliefs about different hypothesis

Posterior What is my **updated** belief after having seen data

Evidence What is my belief about the data







$$\boldsymbol{\theta}^* = \operatorname{argmax}_{\boldsymbol{\theta}} p_{\boldsymbol{\theta}}(\mathcal{D})$$

$$\begin{aligned}\log p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta}) &= \log \int p(\mathbf{y} \mid \mathbf{f}) p(\mathbf{f} \mid \mathbf{X}, \boldsymbol{\theta}) d\mathbf{f} \\ &= -\frac{1}{2} \mathbf{y}^T (k(\mathbf{X}, \mathbf{X}) + \beta^{-1} \mathbf{I})^{-1} \mathbf{y} - \frac{1}{2} \log \det (k(\mathbf{X}, \mathbf{X}) + \beta^{-1} \mathbf{I}) \\ &\quad - \frac{N}{2} \log 2\pi\end{aligned}$$

$$p(y) = \int p(y | f)p(f)df$$

Code

```
import numpy as np
```

```
...
```

- GPy
- GPFlow
- GPTorch
- GPJax

Numerical Stability

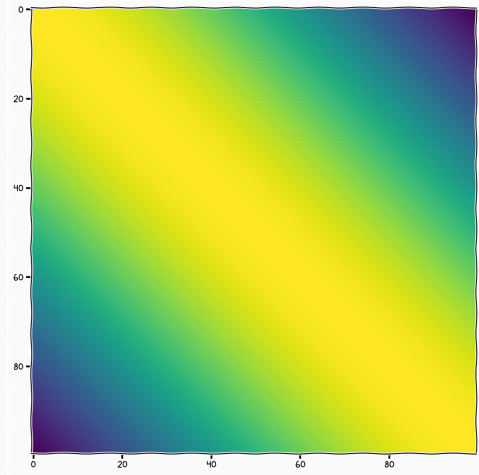
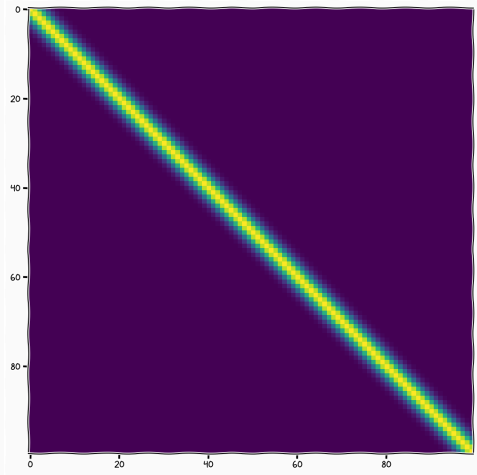
$$\boldsymbol{\theta}^* = \operatorname{argmax}_{\boldsymbol{\theta}} -\mathbf{y}^T \left(k(\mathbf{X}, \mathbf{X}) + \beta^{-1} \mathbf{I} \right)^{-1} \mathbf{y} - \log \det \left(k(\mathbf{X}, \mathbf{X}) + \beta^{-1} \mathbf{I} \right)$$

$$\log \det (k(\mathbf{X}, \mathbf{X}) + \beta^{-1} \mathbf{I})$$

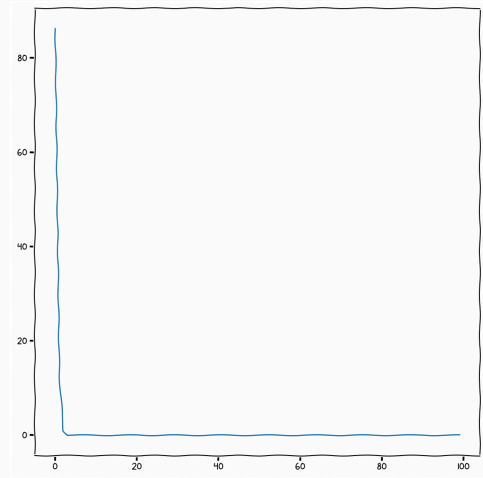
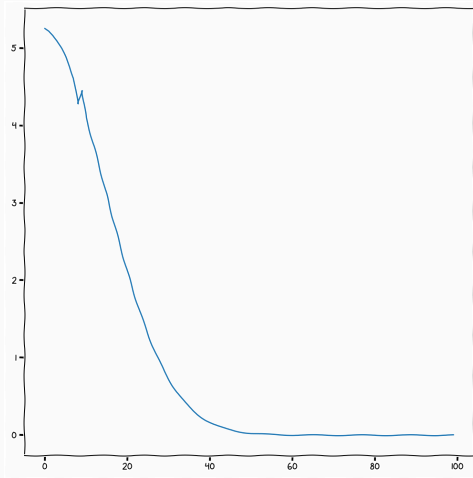
Code

```
np.log(np.linalg.det(K+1/beta*np.eye(K.shape[0])))
```

Co-Variance Matrices



Eigen-decomposition



$$\mathbf{K} = \mathbf{L}\mathbf{L}^T$$

- Factorisation of a *Hermitian* and *Positive-Semi-Definite* Matrix into the product of two lower-triangular matrices

$$\begin{aligned}\log \det \mathbf{K} &= \log \det (\mathbf{L}\mathbf{L}^T) = \\ &\log (\det \mathbf{L})^2 = \\ &\log \left(\prod_i^N \ell_{ii} \right)^2 = 2 \sum_i^N \ell_{ii}\end{aligned}$$

$$\mathbf{y}^T \mathbf{K}^{-1} \mathbf{y}$$

- Matrix inverse have cubic complexity $\mathcal{O}(n^3)$
- Finding the general inverse is numerically tricky
- The matrix is structured and we do not need the explicit matrix

$$\mathbf{y}^T \mathbf{K}^{-1} \mathbf{y} = \mathbf{y}^T (\mathbf{L} \mathbf{L}^T)^{-1} \mathbf{y}$$

$$\begin{aligned}\mathbf{y}^T \mathbf{K}^{-1} \mathbf{y} &= \mathbf{y}^T (\mathbf{L} \mathbf{L}^T)^{-1} \mathbf{y} \\ &= \mathbf{y}^T (\mathbf{L}^{-1})^T \mathbf{L}^{-1} \mathbf{y}\end{aligned}$$

$$\begin{aligned}\mathbf{y}^T \mathbf{K}^{-1} \mathbf{y} &= \mathbf{y}^T (\mathbf{L} \mathbf{L}^T)^{-1} \mathbf{y} \\ &= \mathbf{y}^T (\mathbf{L}^{-1})^T \mathbf{L}^{-1} \mathbf{y} \\ &= (\mathbf{L}^{-1} \mathbf{y})^T \mathbf{L}^{-1} \mathbf{y}\end{aligned}$$

$$\begin{aligned}\mathbf{y}^T \mathbf{K}^{-1} \mathbf{y} &= \mathbf{y}^T (\mathbf{L} \mathbf{L}^T)^{-1} \mathbf{y} \\ &= \mathbf{y}^T (\mathbf{L}^{-1})^T \mathbf{L}^{-1} \mathbf{y} \\ &= (\mathbf{L}^{-1} \mathbf{y})^T \mathbf{L}^{-1} \mathbf{y} \\ &= \mathbf{z}^T \mathbf{z}.\end{aligned}$$

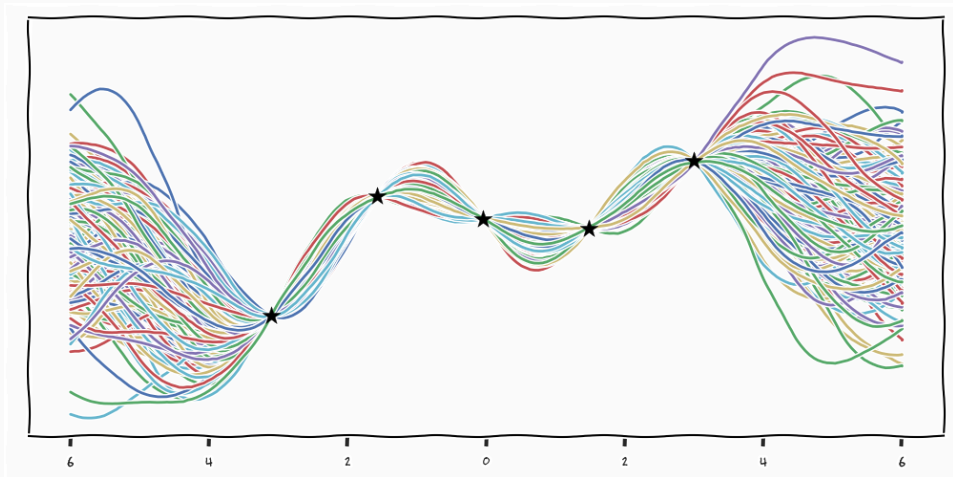
$$\begin{aligned}\mathbf{y}^T \mathbf{K}^{-1} \mathbf{y} &= \mathbf{y}^T (\mathbf{L} \mathbf{L}^T)^{-1} \mathbf{y} \\ &= \mathbf{y}^T (\mathbf{L}^{-1})^T \mathbf{L}^{-1} \mathbf{y} \\ &= (\mathbf{L}^{-1} \mathbf{y})^T \mathbf{L}^{-1} \mathbf{y} \\ &= \mathbf{z}^T \mathbf{z}.\end{aligned}$$

$$\mathbf{L} \mathbf{z} = \mathbf{y}$$

$$\begin{aligned}\ell_{1,1}z_1 &= y_1 \\ \ell_{2,1}z_1 + \ell_{2,2}z_2 &= y_2 \\ \vdots & \\ \ell_{n,1}z_1 + \ell_{n,2}z_2 + \dots + \ell_{n,n}z_n &= y_n,\end{aligned}$$

- we can easily solve $z_1 = \frac{y_1}{\ell_{1,1}}$ and $z_2 = \frac{y_1 - \ell_{2,1}z_1}{\ell_{2,2}}$, etc.
- `scipy.linalg.cho_solve`

Sampling from a Gaussian



$$\int_a^b f(x)dx = \int_{y(a)}^{y(b)} f(x(y)) \frac{dx}{dy} dy$$

Transformation function $y : \mathcal{X} \rightarrow \mathcal{Y}$

Inverse $x : \mathcal{Y} \rightarrow \mathcal{X}$

$$P(a \leq x < b) = \int_a^b p(x) dx$$

$$P(y(a) \leq y < y(b)) = \int_{y(a)}^{y(b)} p(x(y)) \frac{dx}{dy} dy$$

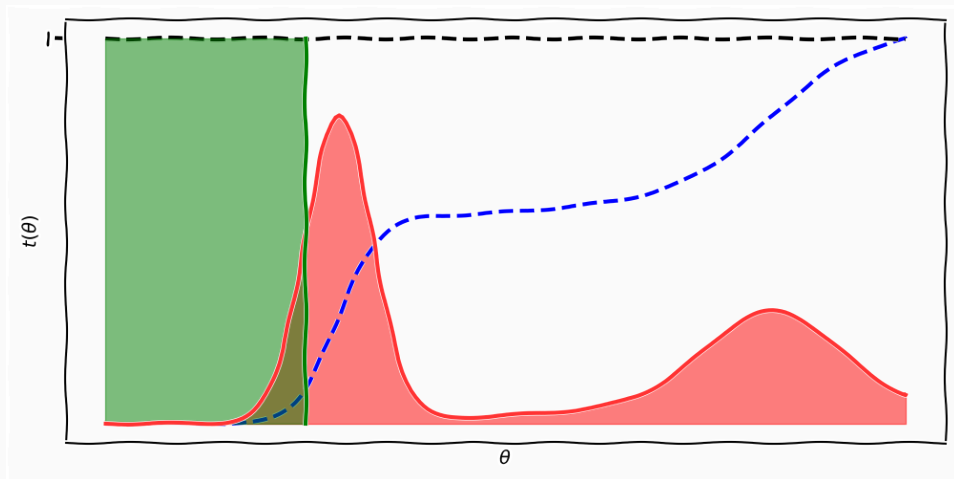
$$\Rightarrow p(y) = p(x(y)) \frac{dx}{dy}$$

$p(x)$ is a density function

- positive and integrates to one $\frac{dx}{dy} \geq 0$ or $\frac{dx}{dy} \leq 0$

$$p(y) = p(x(y)) \left| \frac{dx}{dy} \right|$$

Change of Variable



$$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

$$y = \mathbf{L}\mathbf{z} + \boldsymbol{\mu}$$

$$\mathbf{L}\mathbf{L}^T = \mathbf{K}$$

- Mean

$$\mathbb{E}[\mathbf{y}] = \boldsymbol{\mu}$$

- Covariance

$$\text{Covar}(\mathbf{y}) = \mathbb{E}[\mathbf{y}^T \mathbf{y}] - (\mathbb{E}[\mathbf{y}])^2 = \mathbf{L}\mathbf{L}^T$$

- A numerical method is an "approximation"

- A numerical method is an "approximation"
- Our computers have finite precision

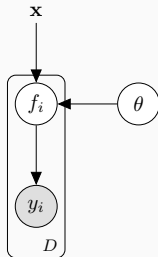
- A numerical method is an "approximation"
- Our computers have finite precision
- Even "worse" they have floating finite precision

- A numerical method is an "approximation"
- Our computers have finite precision
- Even "worse" they have floating finite precision
- Keep the computer in mind when formulating your problem

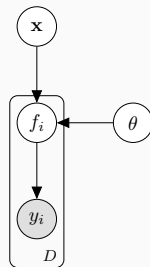
- A numerical method is an "approximation"
- Our computers have finite precision
- Even "worse" they have floating finite precision
- Keep the computer in mind when formulating your problem
- There is a "big" forgotten step going from math to code, don't forget your numerical analysis

Intractabilities

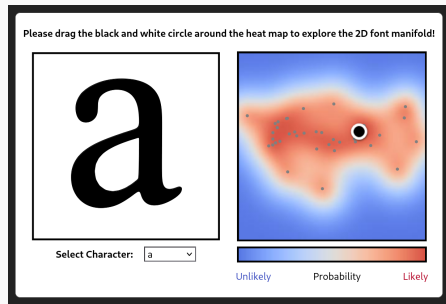
Unsupervised Learning



$$p(y|x) = \int p(y | f)p(f)df$$

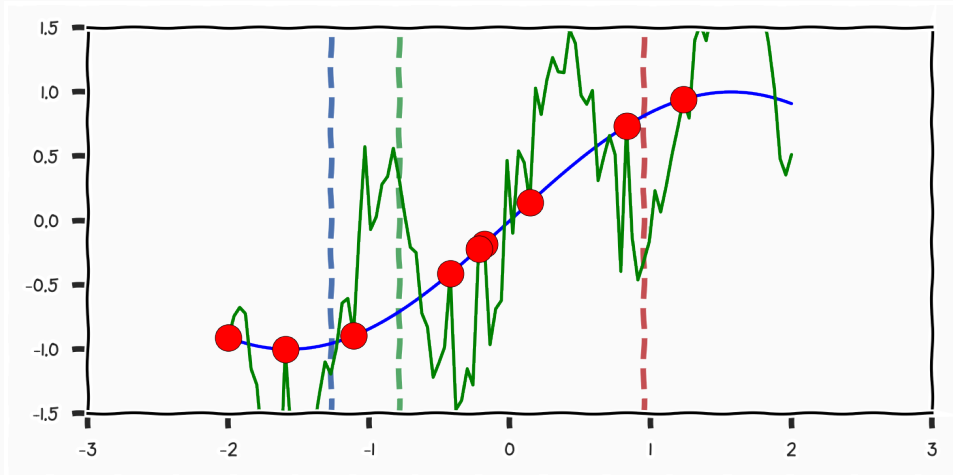


$$p(y) = \int p(y | f, x)p(f | x)p(x)dfdx$$

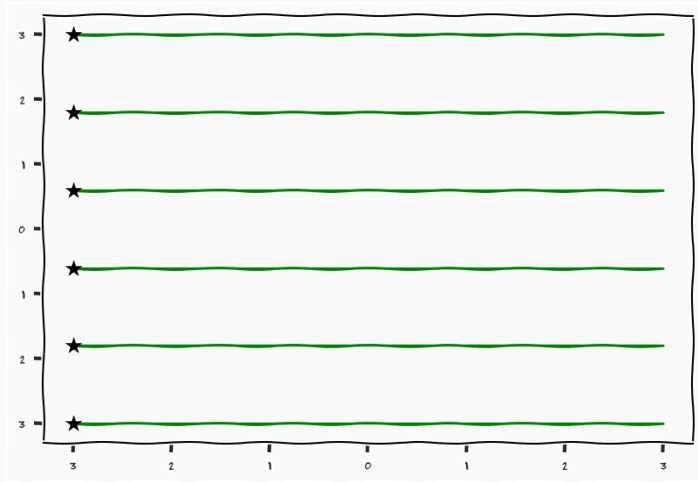


NDF Campbell et al. (July 2014). “Learning a manifold of fonts”. In: *ACM Transactions on Graphics (TOG)* 33.4, p. 91

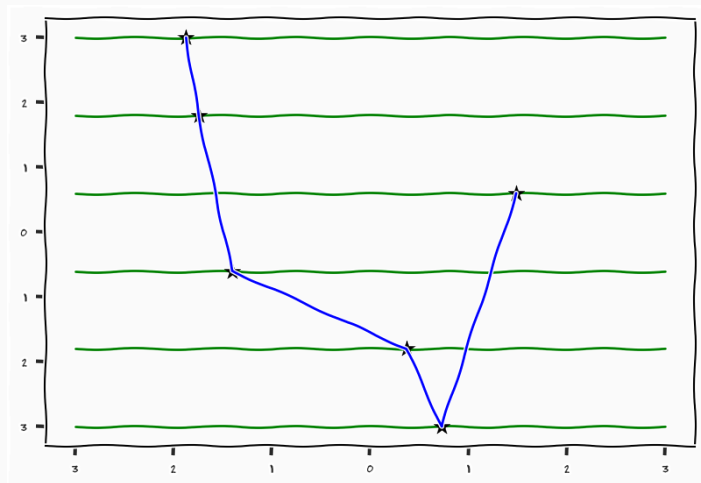
Functions



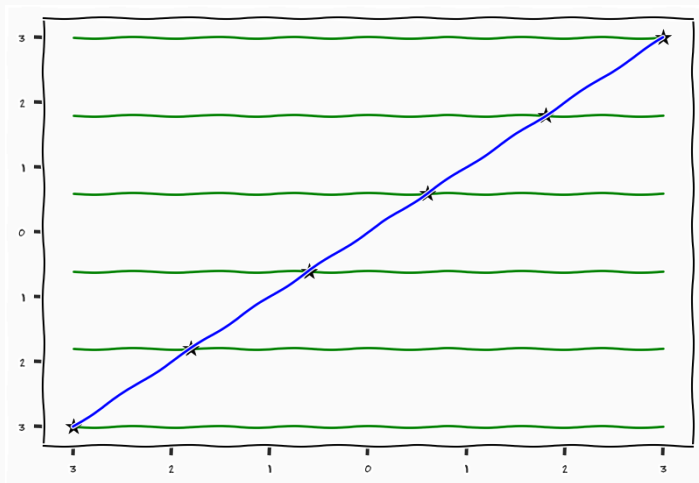
Unsupervised Learning



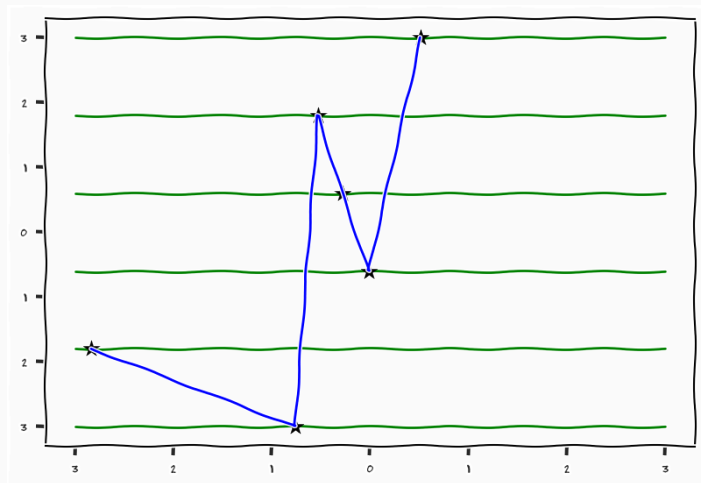
Unsupervised Learning



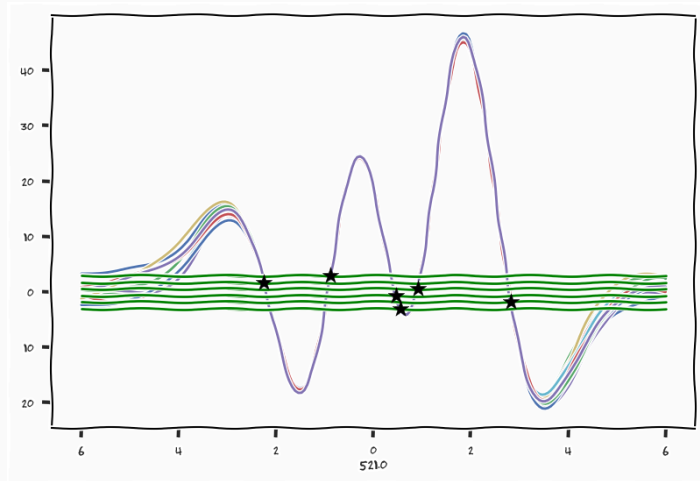
Unsupervised Learning



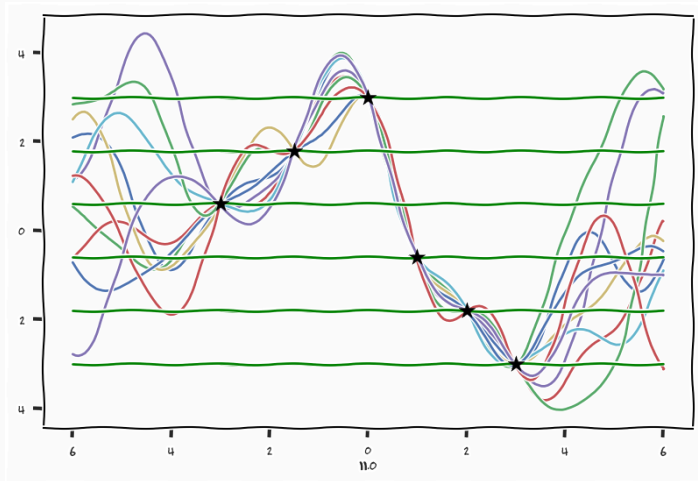
Unsupervised Learning



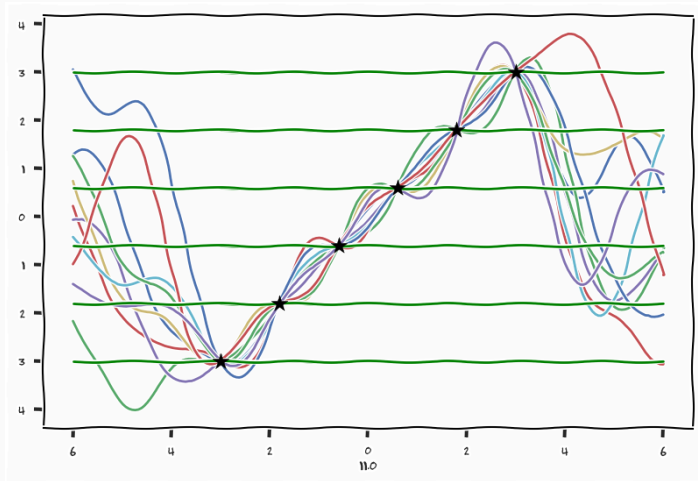
Unsupervised Learning



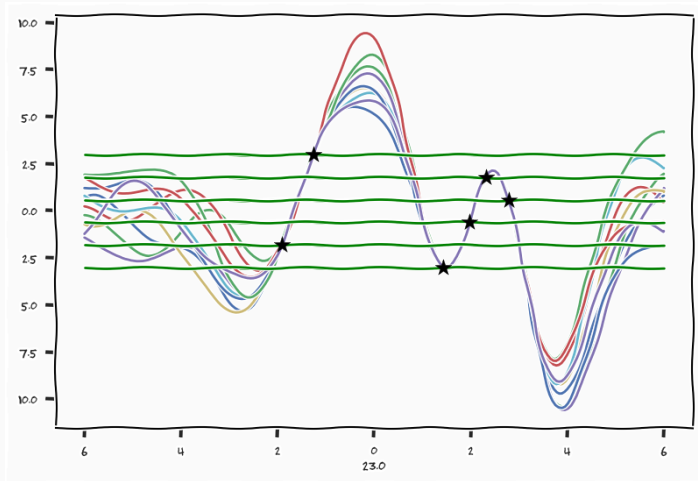
Unsupervised Learning



Unsupervised Learning



Unsupervised Learning



Regression there are infinite number of possible functions that connects the data equally well. A GP provides a measure over these solutions that makes the problem "well-posed".

Regression there are infinite number of possible functions that connects the data equally well. A GP provides a measure over these solutions that makes the problem "well-posed".

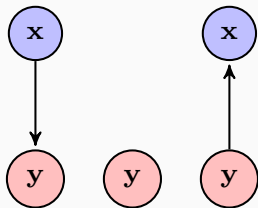
Unsupervised Learning there are infinite number of possible combinations of input locations and functions that generate the data equally well. A GP and a latent space prior jointly provides a measure over these solutions to make the problem "well-posed"

$$p(y) = \int p(y | f)p(f | x)p(x)dfdx$$

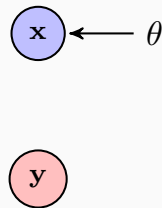
- This integral is analytically intractable

Approximate Inference

$$p(y) = \int p(y \mid x)p(x)dx$$



$$p(y) = \int_x p(y|x)p(x) = \frac{p(y|x)p(x)}{p(x|y)}$$



$$q_{\theta}(x) \approx p(x|y)$$

$$p(y)$$

$$\log p(y)$$

$$\log p(y) = \log p(y) + \int \log \frac{p(x|y)}{p(x|y)} dx$$

$$\begin{aligned}\log p(y) &= \log p(y) + \int \log \frac{p(x|y)}{p(x|y)} dx \\ &= \int q(x) \log p(y) dx + \int q(x) \log \frac{p(x|y)}{p(x|y)} dx\end{aligned}$$

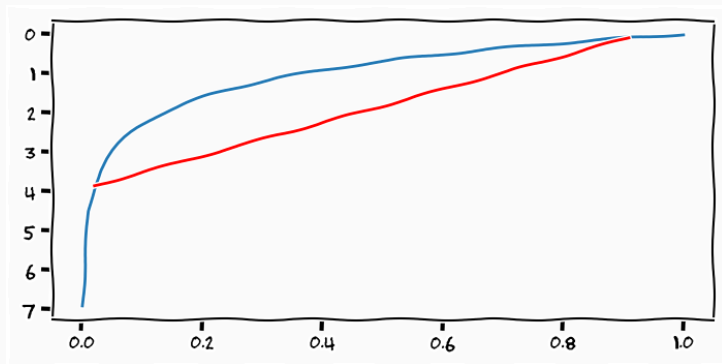
$$\begin{aligned}\log p(y) &= \log p(y) + \int \log \frac{p(x|y)}{p(x|y)} dx \\ &= \int q(x) \log p(y) dx + \int q(x) \log \frac{p(x|y)}{p(x|y)} dx \\ &= \int q(x) \log \frac{p(x|y)p(y)}{p(x|y)} dx\end{aligned}$$

$$\begin{aligned}\log p(y) &= \log p(y) + \int \log \frac{p(x|y)}{p(x|y)} dx \\ &= \int q(x) \log p(y) dx + \int q(x) \log \frac{p(x|y)}{p(x|y)} dx \\ &= \int q(x) \log \frac{p(x|y)p(y)}{p(x|y)} dx = \int q(x) \log \frac{p(x, y)}{p(x|y)} dx\end{aligned}$$

$$\begin{aligned}\log p(y) &= \log p(y) + \int \log \frac{p(x|y)}{p(x|y)} dx \\&= \int q(x) \log p(y) dx + \int q(x) \log \frac{p(x|y)}{p(x|y)} dx \\&= \int q(x) \log \frac{p(x|y)p(y)}{p(x|y)} dx = \int q(x) \log \frac{p(x, y)}{p(x|y)} dx \\&= \int q(x) \log \frac{q(x)}{q(x)} dx + \int q(x) \log p(x, y) dx + \int q(x) \log \frac{1}{p(x|y)} dx\end{aligned}$$

$$\begin{aligned}\log p(y) &= \log p(y) + \int \log \frac{p(x|y)}{p(x|y)} dx \\&= \int q(x) \log p(y) dx + \int q(x) \log \frac{p(x|y)}{p(x|y)} dx \\&= \int q(x) \log \frac{p(x|y)p(y)}{p(x|y)} dx = \int q(x) \log \frac{p(x, y)}{p(x|y)} dx \\&= \int q(x) \log \frac{q(x)}{q(x)} dx + \int q(x) \log p(x, y) dx + \int q(x) \log \frac{1}{p(x|y)} dx \\&= \int q(x) \log \frac{1}{q(x)} dx + \int q(x) \log p(x, y) dx + \int q(x) \log \frac{q(x)}{p(x|y)} dx\end{aligned}$$

Jensen Inequality



$$f\left(\int g \, dx\right) \leq \int f \circ g \, dx,$$

The "posterior" term

$$\int q(x) \log \frac{q(x)}{p(x|y)} dx$$

The "posterior" term

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

The "posterior" term

$$\begin{aligned}\int q(x) \log \frac{q(x)}{p(x|y)} dx &= - \int q(x) \log \frac{p(x|y)}{q(x)} dx \\ &\geq \log \int p(x|y) dx \\ &= \log 1 = 0\end{aligned}$$

The "posterior" term

$$\int q(x) \log \frac{q(x)}{p(x|y)} dx$$

The "posterior" term

$$\int q(x) \log \frac{q(x)}{p(x|y)} dx = \{\text{Lets assume that } q(x) = p(x|y)\}$$

The "posterior" term

$$\begin{aligned}\int q(x) \log \frac{q(x)}{p(x|y)} dx &= \{\text{Lets assume that } q(x) = p(x|y)\} \\ &= \int p(x|y) \log \underbrace{\frac{p(x|y)}{p(x|y)}}_{=1} dx\end{aligned}$$

The "posterior" term

$$\begin{aligned}\int q(x) \log \frac{q(x)}{p(x|y)} dx &= \{\text{Lets assume that } q(x) = p(x|y)\} \\ &= \int p(x|y) \log \underbrace{\frac{p(x|y)}{p(x|y)}}_{=1} dx \\ &= 0\end{aligned}$$

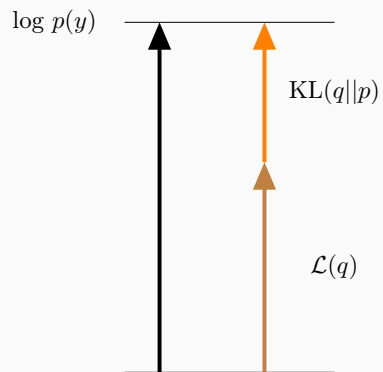
$$\text{KL}(q(x)||p(x|y)) = \int q(x) \log \frac{q(x)}{p(x|y)} dx$$

- $\text{KL}(q(x)||p(x|y)) \geq 0$
- $\text{KL}(q(x)||p(x|y)) = 0 \Leftrightarrow q(x) = p(x|y)$
- Measure of divergence between distributions
- Not a metric (not symmetric)

$$\begin{aligned}\log p(y) &= \int q(x) \log \frac{1}{q(x)} dx + \int q(x) \log p(x, y) dx + \int q(x) \log \frac{q(x)}{p(x|y)} dx \\ &\geq - \int q(x) \log q(x) dx + \int q(x) \log p(x, y) dx\end{aligned}$$

- The Evidence Lower BOnd
- Tight if $q(x) = p(x|y)$

Deterministic Approximation



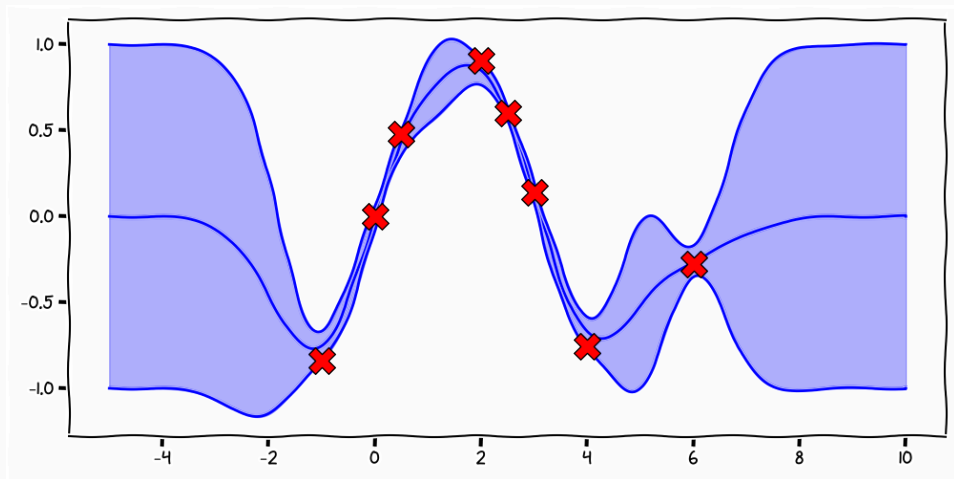
$$\begin{aligned}\log p(y) &\geq - \int q(x) \log q(x) dx + \int q(x) \log p(x, y) dx \\ &= \mathbb{E}_{q(x)} [\log p(x, y)] - H(q(x)) = \mathcal{L}(q(x))\end{aligned}$$

- if we maximise the ELBO we,
 - find an approximate posterior
 - lower bound the marginal likelihood
- *maximising* $p(y)$ **is** learning
- finding $q(x) \approx p(x|y)$ **is** prediction

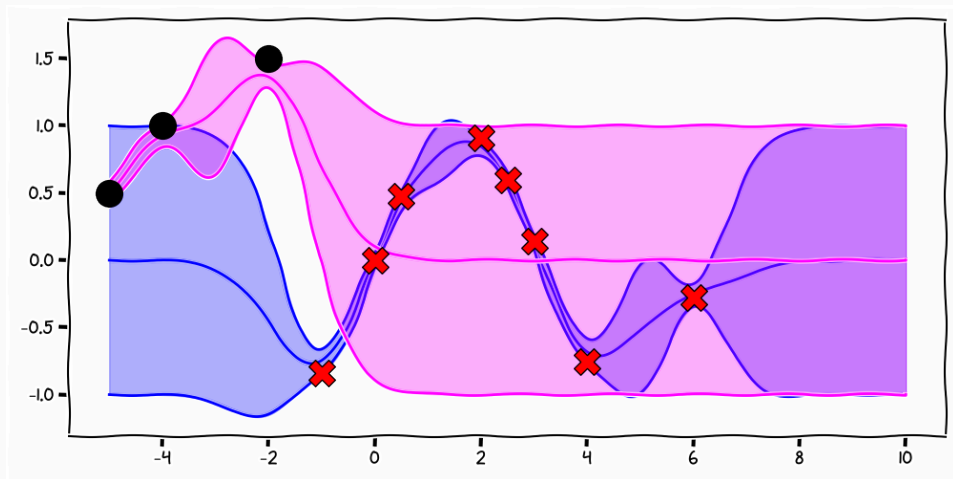
$$\mathcal{L}(q(x)) = \mathbb{E}_{q(x)} [\log p(x, y)] - H(q(x))$$

- We have to be able to compute an expectation over the joint distribution
- The second term should be trivial

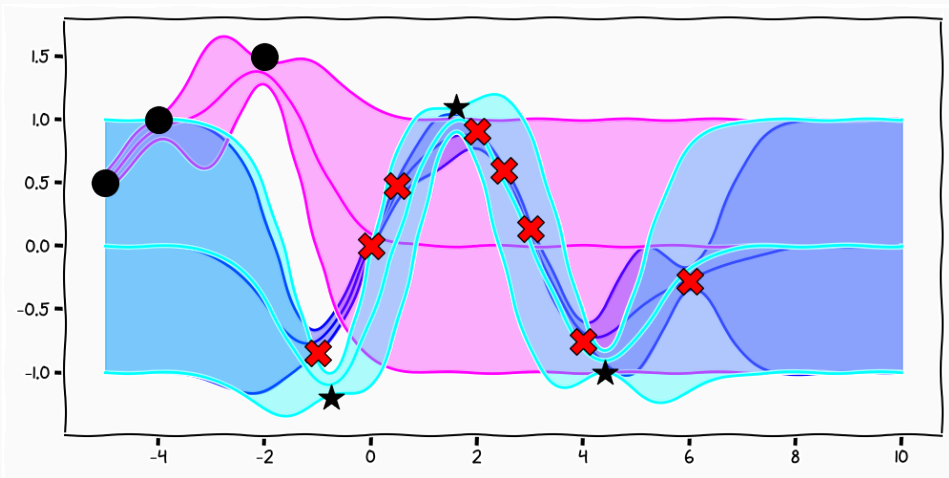
Gaussian Processes Q



Gaussian Processes Q



Gaussian Processes Q



$$p(f, u \mid x, z)$$

- Add another set of samples from the same prior
- Conditional distribution

²Titsias et al., [2010](#)

$$p(f, u \mid x, z) = p(f \mid u, x, z)p(u \mid z)$$

- Add another set of samples from the same prior
- Conditional distribution

²Titsias et al., [2010](#)

$$\begin{aligned} p(f, u \mid x, z) &= p(f \mid u, x, z) p(u \mid z) \\ &= \mathcal{N}(f \mid K_{fu} K_{uu}^{-1} u, K_{ff} - K_{fu} K_{uu}^{-1} K_{uf}) \mathcal{N}(u \mid \mathbf{0}, K_{uu}) \end{aligned}$$

- Add another set of samples from the same prior
- Conditional distribution

²Titsias et al., [2010](#)

$$p(y, f, u, x \mid z) = p(y \mid f)p(f \mid u, x)p(u \mid z)p(x)$$

- we have done nothing to the model, just project an additional set of marginals from the GP
- *however* we will now **interpret** u and z not as **random** variables but **variational** parameters
- i.e. the variational distribution $q(\cdot)$ is parametrised by these

- Variational distributions are approximations to intractable posteriors,

$$q(u) \approx p(u \mid y, x, z, f)$$

$$q(f) \approx p(f \mid u, x, z, y)$$

$$q(x) \approx p(x \mid y)$$

- Variational distributions are approximations to intractable posteriors,

$$q(u) \approx p(u \mid y, x, z, f)$$

$$q(f) \approx p(f \mid u, x, z, y)$$

$$q(x) \approx p(x \mid y)$$

- Bound is **tight** if u completely represents f i.e. u is sufficient statistics for f

$$q(f) \approx p(f \mid u, x, z, y) = p(f \mid u, x, z)$$

$$\mathcal{L} = \int_{x,f,u} q(f) q(u) q(x) \log \frac{p(y, f, u \mid x, z) p(x)}{q(f) q(u) q(x)}$$

$$\begin{aligned}\mathcal{L} &= \int_{x,f,u} q(f)q(u)q(x) \log \frac{p(y, f, u \mid x, z)p(x)}{q(f)q(u)q(x)} \\ &= \int_{x,f,u} q(f)q(u)q(x) \log \frac{p(y \mid f)p(f \mid u, x, z)p(u \mid z)p(x)}{q(f)q(u)q(x)}\end{aligned}$$

- Assume that u is sufficient statistics of f

$$q(f) = p(f \mid u, x, z)$$

$$\tilde{\mathcal{L}} = \int_{x,f,u} q(f) q(u) q(x) \log \frac{p(y | f) p(f | u, x, z) p(u | z) p(x)}{q(f) q(u) q(x)}$$

$$\begin{aligned}\tilde{\mathcal{L}} &= \int_{x,f,u} q(f)q(u)q(x) \log \frac{p(y|f)p(f|u,x,z)p(u|z)p(x)}{q(f)q(u)q(x)} \\ &= \int_{x,f,u} p(f|u,x,z)q(u)q(x) \log \frac{p(y|f)p(f|u,x,z)p(u|z)p(x)}{p(f|u,x,z)q(u)q(x)}\end{aligned}$$

$$\begin{aligned}\tilde{\mathcal{L}} &= \int_{x,f,u} q(f)q(u)q(x) \log \frac{p(y|f)p(f|u,x,z)p(u|z)p(x)}{q(f)q(u)q(x)} \\ &= \int_{x,f,u} p(f|u,x,z)q(u)q(x) \log \frac{p(y|f)p(f|u,x,z)p(u|z)p(x)}{p(f|u,x,z)q(u)q(x)}\end{aligned}$$

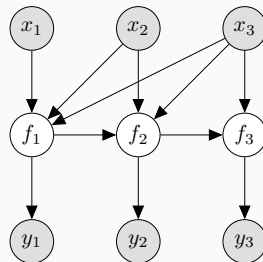
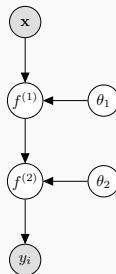
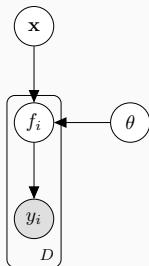
$$\begin{aligned}\tilde{\mathcal{L}} &= \int_{x,f,u} q(f)q(u)q(x) \log \frac{p(y|f)p(f|u,x,z)p(u|z)p(x)}{q(f)q(u)q(x)} \\ &= \int_{x,f,u} p(f|u,x,z)q(u)q(x) \log \frac{p(y|f)\textcolor{red}{p(f|u,x,z)}p(u|z)p(x)}{\textcolor{red}{p(f|u,x,z)}q(u)q(x)} \\ &= \int_{x,f,u} p(f|u,x,z)q(u)q(x) \log \frac{p(y|f)p(u|z)p(x)}{q(u)q(x)}\end{aligned}$$

$$\begin{aligned}\tilde{\mathcal{L}} &= \int_{x,f,u} q(f)q(u)q(x) \log \frac{p(y|f)p(f|u,x,z)p(u|z)p(x)}{q(f)q(u)q(x)} \\&= \int_{x,f,u} p(f|u,x,z)q(u)q(x) \log \frac{p(y|f)\textcolor{red}{p(f|u,x,z)}p(u|z)p(x)}{\textcolor{red}{p(f|u,x,z)}q(u)q(x)} \\&= \int_{x,f,u} p(f|u,x,z)q(u)q(x) \log \frac{p(y|f)p(u|z)p(x)}{q(u)q(x)} \\&= \mathbb{E}_{p(f|u,x,z)}[p(y|f)] - \text{KL}(q(u) \parallel p(u|z)) - \text{KL}(q(x) \parallel p(x))\end{aligned}$$

$$\mathcal{L} = \mathbb{E}_{p(f|u,x,z)}[p(y | f)] - \text{KL}(q(u) \parallel p(u | z)) - \text{KL}(q(x) \parallel p(x))$$

- Expectation tractable (for some co-variances) Titsias et al., 2010
- Stochastic inference Hensman et al., 2013
- Importantly $p(x)$ only appears in $\text{KL}(\cdot \parallel \cdot)$ term!

Interesting Models



Summary

- Hopefully this gave you a flavour of the "practical" part of working with probabilistic models
- You are not expected to know this, but having it in the back of your mind
- Remember the no-free lunch, any result is relative to the assumptions that you put in
- Computations and implementations makes up a huge part of your assumptions

eof

References

References

-  Campbell, NDF and J Kautz (July 2014). “**Learning a manifold of fonts**”. In: *ACM Transactions on Graphics (TOG)* 33.4, p. 91.
-  Hensman, James, N Fusi, and Neil D Lawrence (2013). “**Gaussian Processes for Big Data**”. In: *Uncertainty in Artificial Intelligence*.
-  Lawrence, Neil D. (2004). “**Gaussian Process Models for Visualisation of High Dimensional Data**”. In: *Advances in Neural Information Processing Systems*. Ed. by Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf. Vol. 16. Cambridge, MA: MIT Press, pp. 329–336.

- 📄 MacKay, D. J. C. (1991). “**Bayesian Methods for Adaptive Models**”. PhD thesis. California Institute of Technology.
- 📄 Titsias, Michalis and Neil D Lawrence (2010). “**Bayesian Gaussian Process Latent Variable Model**”. In: *International Conference on Artificial Intelligence and Statistical Learning*, pp. 844–851.