Introduction to Probabilistic Machine Learning

for innumerate computer scientists

Jacob Moss

Last updated: November 30, 2020

Contents

Todo List:

1. Beta distribution derivation, arising from Bayesian inference;
2. Monte Carlo
3. Things they don't tell us...

1 Introduction

The purpose of this pamphlet is to give the foundations and intuitions for probablistic machine learning. The targeted audience are Computer Scientists who might have missed out on some critical components in their mathematical education. As I am a PhD student in the University of Cambridge, a notoriously Bayesian stronghold, there will be a significant Bayesian tinge to the topics discussed. I regularly update this document with new material, clarifications, and corrections. Please do contact me at jm2311@cam.ac.uk if you spot any mistakes or have any requests.

In machine learning, we typically try to fit a model to a dataset. This model may be parametererised by θ. In a Bayesian model, we assign some prior distribution over parameters. We also have a likelihood: the probability of the data given a particular parameter setting. For example, the likelihood of a coin toss is $p(x = 1|\theta) = \theta$ where $\theta = 0.5$ for a even-weighted coin. This is an example of a discrete distribution. In a Bayesian setting, however, this θ is not fixed–it is a distribution. In the next few sections, it will become clear how a Bayesian mindset enables us to apply our prior knowledge that θ is around 0.5!

The key thing to understand in Bayesian statistics is the *posterior update*; using the prior and likelihood, the posterior is updated according to the data. This is achieved using Bayes' theorem: the root of all Bayesian statistics,

$$p_\Theta(\theta|\mathbf{x}) = \frac{p_D(\mathbf{x}|\theta)p_\Theta(\theta)}{p_D(\mathbf{x})},$$

where $p_\Theta(\theta|\mathbf{x})$ is the posterior, $p_D(\mathbf{x}|\theta)$ is the likelihood, $p_\Theta(\theta)$ is the prior, and $p_D(\mathbf{x})$ is the evidence or the marginal likelihood. The marginal likelihood is very important, see Section 4.3.

Note that some literature may use $lik(\theta|\mathbf{x})$, which is the *likelihood function*: it is neither a conditional probability nor a probability density. Likelihood is the probability of observing the dataset given the parameters.

Some knowledge of expectation and variance is expected. Here is a derivation of a useful result that is assumed in later sections.

$$\begin{aligned}
Var(X) &= \mathbb{E}((X - \mathbb{E}(X))^2) = \mathbb{E}\left(X^2 + \mathbb{E}(X)^2 - 2X\mathbb{E}(X)\right) \\
&= \mathbb{E}(X^2) + \mathbb{E}(X)^2 - 2\mathbb{E}(X)^2 = \mathbb{E}(X^2) - \mathbb{E}(X)^2 \\
\implies Var(aX + b) &= \mathbb{E}\left((aX + b)^2\right) - \mathbb{E}\left(aX + b\right)^2 \\
&= \mathbb{E}\left(a^2X^2 + b^2 + 2abX\right) - (a\mathbb{E}(X) + b)^2 \\
&= a^2\mathbb{E}(X^2) + 2ab\mathbb{E}(X) - a^2\mathbb{E}(X)^2 - 2ab\mathbb{E}(X) \\
&= a^2(\mathbb{E}(X^2) - \mathbb{E}(X)^2) = a^2Var(X)
\end{aligned}$$

1.1 Types of Probability

- **Marginal:** Marginalisation is simply summing out/integrating over the probability of a R.V as follows:

$$p(x) = \sum_y p(x, y) = \int p(x, y)\,\mathrm{d}y$$

- **Joint:** $p_{\mathcal{X},\mathcal{Y}}(x, y) = p(\mathcal{X} = x \text{ and } \mathcal{Y} = y)$
- **Conditional:** $p(y|x)$

Another useful result is the law of total probability:

$$p(x) = \int p(x|y)p(y)\,\mathrm{d}y$$

1.2 Maximum Likelihood Estimation

MLE is simple, maximise the likelihood (or often the log-likelihood, since log is an increasing function and often makes the analytical solution easier to derive).

$$\theta* = \max_{\theta} \text{lik}(\theta|\mathbf{x}) = p_{\mathcal{X}}(\mathbf{x}|\theta)$$

where $\Pr_{\mathcal{X}}$ is either the probability mass for discrete variables or the density for continuous variables.

For example, with a binomial probability:

$$\theta* = \max_{\theta} \binom{n}{x} p^k (1-p)^{n-k}$$

todo, finish mle

1.3 Distributions

1.3.1 Gaussian

Very special distribution, used for measuring *magnitudes*, for example heights or temperatures. All roads lead to the Gaussian. The Central Limit Theorem states that the sum of independent random variables (distributed however) tend to a Gaussian.

A joint Gaussian distribution is the same as a multivariate Gaussian.

Counts

Suppose we have an experiment where we toss a coin n times, observing k heads. What is π, the probability of getting head? If we pursue a frequentist approach of maximum likelihood:

$$p(k|\pi, n) \propto \pi^k (1-\pi)^{n-k}$$

$$\arg\max_{\pi} p(k|\pi, n) = \arg\max_{\pi} \log p(k|\pi, n) = \arg\max_{\pi} k \log \pi + (n-k) \log(1-\pi)$$

$$\frac{\partial \log p(k|\pi, n)}{\partial \pi} = \frac{k}{\pi} - \frac{n-k}{1-\pi} = 0 \implies \pi = \frac{k}{n}$$

If we approach instead in a Bayesian setting, we apply a prior distribution on our probability π... a distribution over probabilities?

1.3.2 Beta Distribution

$$\text{Beta}(\pi|\alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \pi^{\alpha-1}(1-\pi)^{\beta-1}$$

α and β are shape parameters and relate to the pseudo-count +1. The Beta distribution is defined over $[0, 1]$ and is actually a special case of the Dirichlet distribution, which will be discussed in a later section. The Beta is a conjugate prior to the Binomial distribution (the posterior is Beta, likelihood is Binomial).

Continuing with the Bayesian approach; let our data consist of a single coin toss: $\mathcal{D} = \{k = 1\}$ with $n = 1$. Our likelihood:

$$p(\mathcal{D}|\pi) = \pi$$

Our posterior:

$$p(\pi|\mathcal{D}) = \frac{p(\pi|\alpha, \beta) p(\mathcal{D}|\pi)}{p\mathcal{D}} \propto \text{Beta}(\pi|\alpha, \beta)\pi$$

$$\propto \pi^{\alpha}(1-\pi)^{\beta-1} \propto \text{Beta}(\alpha+1, \beta)$$

Word Counts For example a word frequency bar chart. Zipf's law states that the frequency of any word is inversely proportional to its index in the ordered frequency table. Just counting words, not taking into account order, is very simple and not very useful. Words like 'the' are the most frequent but say the least.

1.3.3 Multinomial Distribution

1.3.4 Dirichlet Distribution

The Dirichlet distribution is, as mentioned, a generalisation of the Beta distribution, defined on the $m-1$ dimensional simplex. A simplex is a higher-dimensional triangle, defined by $m-1$ points in an m-dimensional space. The Dirichlet is the *conjugate prior* for the multinomial.

- The **binomial** to the **Bernoulli** is the **multinomial** to the **categorical**. (multiple trials)
- The **Dirichlet** to the **multinomial** is the **Beta** to the **binomial**.

$$\text{Dir}(\pi|\alpha_1,...,\alpha_m) = \frac{\Gamma(\sum_{i=1}^m \alpha_i)}{\prod_{i=1}^m \Gamma(\alpha_i)} \prod_{i=1}^m \pi^{\alpha_i - 1}$$

The α_is are the shape [illegible] t.

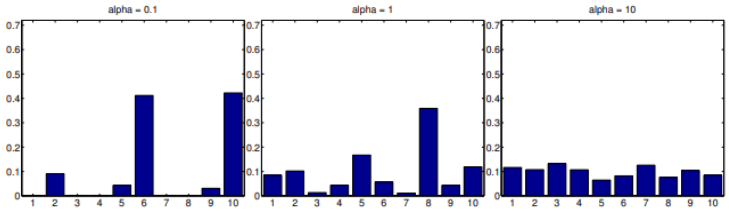


Figure 1: Symmetric Dirichlet plot with various settings of α

1.3.5 Mapping Between Random Variables

We will discuss a method called Inverse Transform Sampling here, which is used to sample from other distributions using the c.d.f. We discuss more complicated sampling methods in Section 7. Suppose we have a continuous r.v. U with p.d.f. $p_U(u)$ and c.d.f. $F_U(u)$, and we seek a function $f : U \to X$ where X is our desired distribution. For maintaining the order, f must be a monotonically increasing function.

The conservation of probability, $p_U(u)du = p_X(x)dx$, means:

$$p_U(f^{-1}(x))du = p_X(x)dx$$

$$\implies p_X(x) = p_U(f^{-1}(x))\frac{du}{dx} = p_U\left(f^{-1}(x)\right)\frac{df^{-1}(x)}{du}$$

The same applies to the c.d.f.:

$$F_X(x) = \int_{-\infty}^{x} p_X(t)dt = \int_{-\infty}^{x} p_U\left(f^{-1}(t)\right)\frac{df^{-1}(t)}{du}dt$$
$$\implies F_X(x) = F_U(f^{-1}(x))$$

Let U be a uniform distribution s.t. $F_U(u) = u$.

$$\implies F_X(x) = F_U(f^{-1}(x)) = f^{-1}(x) = u \implies x = F_X^{-1}(u)$$

Therefore, if we have the inverse c.d.f. then we can sample from the distribution. The same applies to discrete distributions.

1.4 Softmax Transformation

The softmax function is very useful in machine learning, it takes a length K vector and outputs a normalised probability distribution (adds to 1) consisting of K probabilities. It is used in multi-class classification, and can be used in multi-variate optimisation. For example, optimise

$$f(x_1, x_2, x_3) = 0.2x_1 + 0.3x_2 + 0.5x_3$$

where $x_1, x_2, x_3 \in [0, 1]$ and $x_1 + x_2 + x_3 = 1$

We can set $x_3 = 1 - x_1 - x_2$. The softmax is normalised, which will satisfy the constraint. Also due to this constraint, we have only two free variables.

$$x_1 = \frac{e^{\xi_1}}{e^{\xi_1} + e^{\xi_2} + 1},\ x_1 = \frac{e^{\xi_2}}{e^{\xi_1} + e^{\xi_2} + 1},\ x_3 = \frac{1}{e^{\xi_1} + e^{\xi_2} + 1}$$

todo finish

1.5 Types of Models

The **Generative** model estimates the joint distribution, and from that computes the posterior $p_{\mathcal{Y}}(y|x)$ to make predictions. Often this is done by inferring the likelihood $p_{\mathcal{X}}(x|y)$ and prior $p_{\mathcal{Y}}(y)$.

Learning the posterior directly is termed **Discriminative** classification.

1.6 Keep in mind

When marginalising out a variable, the limits are often not explicit.

$$\mathcal{L}_i(\Theta) = -\log\left(\prod_{j=1}^{K}\int p_{ij}^{y_{ij}+\alpha_{ij}-1} dp_{ij}\right)$$

$$\mathcal{L}_i(\Theta) = -\log\left(\prod_{j=1}^{K}\int_0^1 p_{ij}^{y_{ij}+\alpha_{ij}-1} dp_{ij}\right)$$

1.7 Disclaimer

Some images are taken from Carl Rasmussen's Probabilistic ML (4f13) course at Cambridge or Dariush Hosseini's data mining course at UCL.

2 Graphical Models

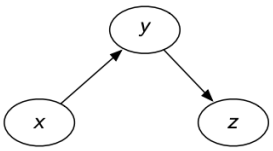
Graphical models are sometimes treated as a separate topic, but I prefer to view them as a tool for visualising and constructing probabilistic models. A graphical model is essentially a dependency graph, where there is an arrow from node A to node B, where A, B are r.v.s, if B is conditioned on A.

2.1 Directed Graphical Models

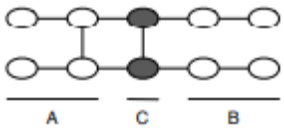
They are directed acyclic graphs, where for each node we assign a random variable X_i and a probability density function $f_i(x_i, x_{\pi_i})$, where π_i is the set of parents of X_i, which are those on which X_i conditions. The graphical model is possible using the chain rule of probability:

$$p_{X_1,...,X_n}(x_1, ..., x_n) = p_{X_1}(x_1)p_{X_2|X_1}(x_2|x_1) \cdots p_{X_1,...,X_{n-1}}(x_n|x_1, ..., x_{n-1}) \tag{1}$$

Equation 1 creates a fully-connected DAG, which can represent any probability distribution. The joint probability distribution for N variables where each variable can take on a value in $\mathcal{X}$, requires a table of size $|\mathcal{X}|^N$. When the conditional distributions involved do not depend on all conditioning variables some edges can be removed. Sparse graphs can lead to more efficient inference.



(a) $p_{X,Y,Z} = p_{Z|Y}p_{Y|X}p_X$



(b) $X_A \perp\!\!\!\perp X_B$

Figure 2: Directed and undirected graphical models

2.2 Markov Random Fields

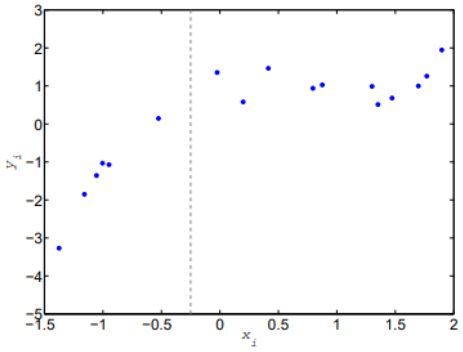
Markov Random Fields are **undirected graphical models** which satisfy the Markov property: the future and past are conditionally independent given the present, i.e., $X \perp\!\!\!\perp Y|Z$ (independent) iff all paths from a node in X to a node in Y pass through a node in Z. In (b), we see that nodes in A are indep. of nodes in B.

2.3 Factor Graphs

3 Regression

3.1 Least Squares Regression

How can we fit a line through these points? Perhaps a polynomial regression would be good (but which polynomial?). We use the $\phi_j(x)$ as the *basis function*. The model below is *linear in parameters* but not linear in variables.



$$
\begin{aligned}
y* &= f_w(x) = w_0 + w_1 x + w_2 x^2 + ... + w_M x^M \\
&= \sum_{j=0}^{M} w_j \phi_j(x) \quad \text{where } \phi_j(x) = x^j \\
\mathbf{y}* &= \mathbf{X}\mathbf{w}
\end{aligned}
$$

where $\mathbf{X}$ is the design matrix $[\phi_0(x), \phi_1(x), ...]^T$ and $\mathbf{w}$ is the row vector of weights.

We then attempt to minimise the error, which is the discrepancy between the actual points $\mathbf{y}$ and the model estimates $\mathbf{y}*$.

$$
\begin{aligned}
\mathbf{e} &= \mathbf{y} - \mathbf{y}* \\
E(\mathbf{w}) &= ||e||^2 = (\mathbf{y} - \mathbf{y}*)^\intercal (\mathbf{y} - \mathbf{y}*) \\
&= (\mathbf{y} - \mathbf{X}\mathbf{w})^\intercal (\mathbf{y} - \mathbf{X}\mathbf{w}) = y^\intercal y - w^\intercal X^\intercal y + (\mathbf{X}\mathbf{w})^\intercal (\mathbf{X}\mathbf{w}) \\
\frac{\partial E(\mathbf{w})}{\partial (w)} &= 2X^\intercal X w - 2y^\intercal X = 0 \\
\implies X^\intercal X w &= y^\intercal X \\
\mathbf{w} &= (X^\intercal X)^{-1} X^\intercal y \quad \text{which is the normal equation}
\end{aligned}
$$

3.2 Regression with Additive Independent Gaussian Noise

An interesting result is that the MLE and OLS regression as shown above yields the same result if we model the function as $y^{(i)} = f_w(x^{(i)}) + \varepsilon^{(i)}$ where $\varepsilon^{(i)} \sim \mathcal{N}(0, \sigma^2)$, σ^2 is the noise variance. Thus $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$, which is called an isotropic multivariate Gaussian. Isotropic because there is only variance along the diagonal, therefore it will be a hypersphere.

Since we are assuming noise is independent, and there are N terms ($len(\boldsymbol{\varepsilon}) = N$), the following result follows simply from the Gaussian PDF:

$$
p(\boldsymbol{\varepsilon}) = \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}) = \prod_{n=1}^{N} p(\varepsilon^{(n)}) = \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^N \exp\left(-\frac{\boldsymbol{\varepsilon}^\intercal \boldsymbol{\varepsilon}}{2\sigma^2}\right)
$$

Since $\mathbf{y} = \mathbf{y}* + \boldsymbol{\varepsilon}$, we can work out the probability of the data $\mathbf{y}$ given the model estimate $\mathbf{y}*$, by constructing the same normal distribution with the mean as the estimate and variance as the variance of the noise! In other words; centered around the point, with the spread at that point.

$$
\boldsymbol{\varepsilon}^\intercal \boldsymbol{\varepsilon} = ||\boldsymbol{\varepsilon}||^2 = ||\mathbf{y} - \mathbf{y}*||^2 = E(\mathbf{w}) \quad \text{from the sum of squared errors from before}
$$

$$
p(\mathbf{y}|\mathbf{y}*) = \mathcal{N}(\mathbf{y}^*, \sigma^2\mathbf{I}) = \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^N \exp\left(-\frac{||\mathbf{y} - \mathbf{y}*||^2}{2\sigma^2}\right)
$$

Since $\mathbf{y}* = \mathbf{X}\mathbf{w}$, for a given $\mathbf{X}$, $p(\mathbf{y}|\mathbf{y}*) = p(\mathbf{y}|\mathbf{w})$. Ahah! The likelihood function of $\mathbf{w}$ has fallen out. Thus we attempt to maximise the likelihood $\mathcal{L}(\mathbf{w}) \propto p(\mathbf{y}|\mathbf{w})$:

$$\mathbf{w}* = \arg\max_w \mathcal{L}(\mathbf{w}) = \arg\max_w \exp\left(-\frac{||\mathbf{y} - \mathbf{y}*||^2}{2\sigma^2}\right) = \arg\min_w E(\mathbf{w})$$

Above is possible since exp is an increasing function. Note how this is the same result at with the least squares method. Overfitting is still an issue, with MLE, more complex models overfit the data. Next we bring Bayes into it.

4 Bayesian Inference

4.1 Maximum a posteriori

The MAP estimate is the point estimate of a parameter θ, which is essentially the MLE but taking into account the *prior*. Suppose we have observations x from the distribution $f(x|\theta)$. Start with the MLE. We therefore need the likelihood: $\text{lik}(\theta|x) = f(x|\theta)$

We can calculate the posterior using Bayes' theorem. We have a prior belief in the form of a distribution over θ, $p_\Theta(\theta)$

$$f(\theta|x) = \frac{f(x|\theta)p_\Theta(\theta)}{f(x)}$$

We now maximise the posterior, noting that the marginal likelihood does not affect the maximisation of θ and is always positive so we can ignore it:

$$\theta* = \arg\max_\theta f(\theta|x) = \arg\max_\theta \frac{f(x|\theta)p_\Theta(\theta)}{f(x)} = \arg\max_\theta f(x|\theta)p_\Theta(\theta)$$

4.2 Fitting a Model

Take a model $\mathcal{M}$, representing a choice of model structure and parameter values. Let the structure be $y = f_w(x) + \varepsilon$, **the probability of the data is conditional:** $p(\mathbf{y}|\mathbf{x})$. The Gaussian likelihood is:

$$p(\mathbf{y}|\mathbf{x},\ \mathbf{w}, \mathcal{M}) \propto \prod_{i=1}^{N} \exp\left(-\frac{(y^{(i)} - f_w(x^{(i)})^2}{2\sigma^2}\right)$$

Fit the model by optimising for the weights (MLE):

$$\mathbf{w}* = \arg\max_w p(\mathbf{y}|\mathbf{x},\ \mathbf{w}, \mathcal{M})$$

After maximisation, make predictions from $p(y|x, \mathbf{w}*, \mathcal{M})$. Notice how it now uses the fitted weights.

With a certain likelihood distribution, the conjugate prior is the distribution such that the posterior will be tractable (there is a closed form). Here we will have a Gaussian likelihood $p(\mathbf{y}|\mathbf{x}, \mathbf{w}, \mathcal{M}) = \mathcal{N}(\mathbf{Xw}, \sigma^2\mathbf{I})$ and Gaussian prior $p(\mathbf{w}|\mathcal{M}) = \mathcal{N}(\mathbf{0}, \sigma^2_\mathbf{w})$.

Let's now apply Bayes' rule as before to include the prior. We first calculate the posterior:

$$\begin{aligned}
p(\mathbf{w}|\mathbf{x},\ \mathbf{y}, \mathcal{M}) &= \frac{p(\mathbf{y}|\mathbf{x}, \mathbf{w}, \mathcal{M})p(\mathbf{w}|\mathcal{M})}{p(\mathbf{y}|\mathbf{x}, \mathcal{M})} = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \\
&\propto \exp\left(-\frac{1}{2\sigma^2}|\mathbf{Y} - \mathbf{Xw}|^2\right) \exp\left(-\frac{1}{2}\mathbf{w}^\intercal\Sigma_w^{-1}\mathbf{w}\right) \\
&= \exp\left(-\frac{1}{2}((\mathbf{Xw})^\intercal\mathbf{Xw}\sigma^{-2} - 2\mathbf{y}^\intercal(\mathbf{Xw})\sigma^{-2} + \mathbf{w}^\intercal\Sigma_w^{-1}\mathbf{w})\right) \\
&= \exp\left(-\frac{1}{2}\left(\mathbf{w}^\intercal(\mathbf{X}^\intercal\mathbf{X}\sigma^{-2} + \Sigma_w^{-1})\mathbf{w} - 2\sigma^{-2}\mathbf{w}^\intercal\mathbf{X}^\intercal\mathbf{y}\right)\right) \qquad (2)
\end{aligned}$$

Notice equation (1), remember completing the square with a scalar equation e.g. x^2+xy+y^2. With matrices, the result is similar:

$$x^\intercal Ax + x^\intercal b + c = (x - \mu)^\intercal A(x - \mu) + k$$

where $\mu = -\frac{1}{2}A^{-1}b$, $k = c - \frac{1}{4}b^\intercal A^{-1}b$. Apply this result to the exponent as follows:

$$= \exp\left(-\frac{1}{2}(\mathbf{w} - \boldsymbol{\mu})^\intercal\boldsymbol{\Sigma}^{-1}(\mathbf{w} - \boldsymbol{\mu})\right)$$

$$\text{where } \boldsymbol{\Sigma} = (\sigma^{-2}\mathbf{X}^\intercal\mathbf{X} + \Sigma_w^{-1})^{-1},$$

$$\boldsymbol{\mu} = -\frac{1}{2} \times -2\sigma^{-2}\boldsymbol{\Sigma}^{-1^{-1}}\mathbf{X}^\intercal\mathbf{y} = \sigma^{-2}\boldsymbol{\Sigma}\mathbf{X}^\intercal\mathbf{y}$$

Reminder of the marginal: $p(x) = \int p(x,y)dy = \int p(x|y)p(y)dy$ we need to marginalise out the parameters in order to make predictions. The following is the **predictive distribution**:

$$p(y|x,\mathbf{x},\mathbf{y},\mathcal{M}) = \int p(y,\mathbf{w}|x,\mathbf{x},\mathbf{y},\mathcal{M})\,\mathrm{d}\mathbf{w} = \int p(y|x,\mathbf{w},\mathbf{x},\mathbf{y},\mathcal{M})p(\mathbf{w}|\mathbf{x},\mathbf{y},\mathcal{M})\,\mathrm{d}\mathbf{w}$$

4.3 Marginal Likelihood

We marginalise out $\mathbf{w}$ from the marginal likelihood $p(\mathbf{y}|\mathbf{x},\mathcal{M})$. If there is confusion regarding $\mathbf{w}$ not being conditioned on, I believe the intuition is that the distribution of $\mathbf{w}$ is in some way parameterised by $\mathbf{x}$. This is an application of the law of total probability.

$$p(\mathbf{y}|\mathbf{x},\mathcal{M}) = \int p(\mathbf{w}|\mathbf{x},\mathcal{M})p(\mathbf{y}|\mathbf{x},\mathbf{w},\mathcal{M})\,\mathrm{d}\mathbf{w}$$

One then optimises the marginal likelihood to tune the hyperparameters.

To demonstrate how the marginal likelihood assists in model selection, we apply Bayes' rule again:

$$p(\mathbf{y}|\mathbf{x},\mathcal{M}) = \frac{p(\mathcal{M}|\mathbf{y},\mathbf{x})p(\mathbf{y}|\mathbf{x})}{p(\mathcal{M})}$$

$$p(\mathcal{M}|\mathbf{y},\mathbf{x}) = \frac{p(\mathbf{y}|\mathbf{x},\mathcal{M})p(\mathcal{M})}{p(\mathbf{y}|\mathbf{x})} \propto p(\mathbf{y}|\mathbf{x},\mathcal{M})p(\mathcal{M})$$

Since the probability of a model given the data is proportional to the marginal likelihood, this provides some hand-wavey intuition as to why the marginal likelihood gives some kind of score to the model.

5 Gaussian Processes

5.1 Non-parametric Modelling

Previously the models discussed have been parametric. These parameters are marginalised over to yield a predictive distribution.

$$p(y^*|x^*, \mathbf{x}, \mathbf{y}) = \int p(y^*|x^*, \mathbf{w}, \mathbf{x}, \mathbf{y}) p(\mathbf{w}|\mathbf{x}, \mathbf{y}) \, \mathrm{d}\mathbf{w}$$

Observe the last term, the $p(\mathbf{w}|\mathbf{x}, \mathbf{y})$ is a bottleneck for the model; there are a fixed number of parameters for the model. In fact, the distribution over the parameters implies a distribution over functions. A non-parametric model would work directly with such a distribution. Moreover, the predictive distribution is usually an intractable integral. Which distribution is easily integrable?

Gaussians
Suppose we have the joint probability:

$$p(\mathbf{x_1}, \mathbf{x_2}) = \mathcal{N}\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \right)$$

- **What is the marginal distribution, $p(\mathbf{x}_1)$?**
  We integrate out $\mathbf{x}_2$:
  $$p(\mathbf{x}_1) = \int p(\mathbf{x_1}, \mathbf{x_2}) d\mathbf{x}_2 = \mathcal{N}(\mu_1, \Sigma_{11})$$
- **What is the conditional distribution, $p(\mathbf{x}_1|\mathbf{x}_2)$?**
  This has the solution $p(\mathbf{x}_1|\mathbf{x}_2) = \mathcal{N}(\mu_c, \Sigma_c)$ where:
  $$\mu_c = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(\mathbf{x}_2 - \mu_2))$$
  $$\Sigma_c = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$$

Recall a model using basis functions: $f_w(x) = \sum_{m=0}^{M} w_m \phi_m(x)$. A prior over the weights $p(\mathbf{w})$ induces a prior distribution over functions, $p(\mathbf{f})$.

How do we make predictions from such a distribution over functions? Given a parametric family of functions, $f(\mathbf{x}|\mathbf{f})$ and a prior over $\mathbf{f}$, Bayesian modelling helps us predict given the data, $p(\mathbf{f}|\mathbf{y}, \mathbf{x})$:

$$p(\mathbf{f}|\mathbf{y}, \mathbf{x}) = \frac{p(\mathbf{y}|\mathbf{x}, \mathbf{f})p(\mathbf{f})}{p(\mathbf{y}|\mathbf{x})}.$$

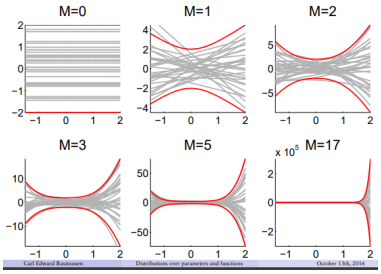


Figure 3: Polynomials are not a great prior over functions: after 1 they experience rapid growth.

5.2 Definition

Definition: a Gaussian process is a collection of random variables, every finite subset of which are jointly Gaussian.

where $m(x)$ is the mean function and $k(x, x')$ is the covariance function. A GP is a Gaussian distribution with an infinitely long mean vector and infinite covariance matrix. We can't really reason with an infinitely long mean vector and covariance matrix, so we restrict ourselves to a finite subset, and rely on the marginalisation property to marginalise out the infinite. So now we've learnt how to draw **random functions**, but this is not useful, we want to somehow model the data.

$$f \sim \mathcal{GP}(m, k)$$

Since we are in a non-parametric model, the parameters are the function itself. Let's go through the same Bayesian inference procedure as in Section 4.2 but plug in the function r.v. instead of the parameter r.v. We first find an expression for the likelihood:

$$p(\mathbf{y}|\mathbf{x}, \mathbf{f}) \sim \mathcal{N}(\mathbf{f}, \sigma^2\mathbf{I})$$

where $\mathbf{f}$ is the vector of function applied to the input data. We can do this instead of a Gaussian distribution over $\mathbf{y}|\mathbf{x}$ since we only know what the function is doing at the observations.
We have a Gaussian process prior for the functions, as we saw before: $p(\mathbf{f}) \sim \mathcal{GP}(m = 0, \mathbf{k})$
The posterior is calculated by multiplying the likelihood and the prior. Product of two Gaussians is a Gaussian. This leads to an infinite Gaussian, which is a Gaussian process.

$$p(\mathbf{f}|\mathbf{x},\ \mathbf{y}) \sim \mathcal{GP}(\mathbf{m}_{\text{post}}, \mathbf{k}_{\text{post}})$$

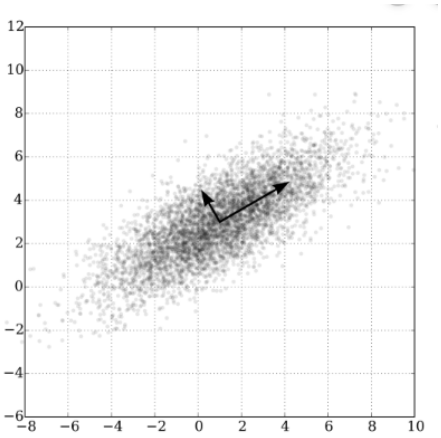


Figure 4: The arrows are eigenvectors of $\mathbf{\Sigma}$ scaled by the sqroot of their eigenvalues.

And the following predictive distribution:

$$p(y_*|x_*, \mathbf{x}, \mathbf{y}) = \int p(y_*, \mathbf{f}|\mathbf{x}, \mathbf{y}, x_*)\, \mathrm{d}\mathbf{f} = \int p(y_*|x_*, \mathbf{f}, \mathbf{x}, \mathbf{y}) p(\mathbf{f}|\mathbf{x},\ \mathbf{y})\, \mathrm{d}\mathbf{f}$$

What does the marginal likelihood look like? The evidence is just the probability of observations where the function has been marginalised out. This yields a *closed form* solution:

$$\log p(\mathbf{y}|\mathbf{x}) = -\frac{1}{2} y^\intercal [K + \sigma_n^2 I]^{-1} y - \frac{1}{2} \log[K + \sigma_n^2 I] - \frac{n}{2} \log(2\pi)$$

where the data fit measures how well the model fits the training data, and the complexity penalty penalises how big the model class that we're using. The nice thing is that there are no hyperparameters here like with regularisers; Occam's razor is automatic.

5.3 Hyperparameters & Covariance Functions

Kernel methods are useful since they can express an infinite amount of features in closed form using the dot product. This section will outline how we can obtain such a kernel. First, **what is positive definiteness?**

A symmetric matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is positive definite if $\mathbf{x}^\intercal \mathbf{A}\mathbf{x} > 0 \quad \forall x \neq 0$. Let's build an intuition of what this means.

$$f(x, y) = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 1 & 3 \\ 3 & 8 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = x^2 + 6xy + 8y^2$$

What are the roots of the equation $f(x, y) = 0$? To calculate them we can set $x = \alpha$ and $y = 1$.[1] Now our equation is $x^2 + 6x + 8$ with roots $x = -2$, $x = -4$. Plug in any value for x in between our two roots:

$$f(-3, 1) = 9 - 18 + 8 = -1$$

shows our matrix is **not** positive definite.

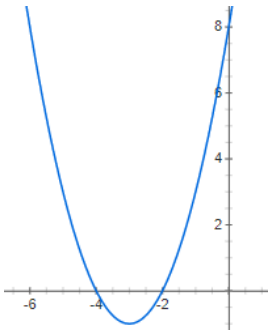


Figure 5: $x^2 + 6x + 8$

[1] We have essentially scaled y by α: let $x = \beta$ and $y = \gamma$. Now set $x = \beta/\gamma = \alpha$ and $y = 1$

Let us now apply this in the case of a covariance matrix. Suppose we have two variables jointly Gaussian:

$$(x, y) \sim \mathcal{N}\left(\begin{bmatrix}\mu & \nu\end{bmatrix}, \boldsymbol{\Sigma} = \begin{bmatrix}\sigma^2 & \alpha \\ \alpha & \rho^2\end{bmatrix}\right)$$

We will now see that for the covariance matrix to have any meaning, it must be positive semi-definite. Suppose $Z = aX + bY$

What is the distribution of Z? Gaussian. The mean is trivial: $\mathbb{E}(aX + bY) = a\mathbb{E}(X) + b\mathbb{E}(Y) = a\mu + b\nu$. The variance is also easy:

$$\begin{aligned}\text{Var}(aX + bY) &= \text{Var}(aX) + \text{Var}(bY) + 2\text{Cov}(aX, bY) = a^2\sigma^2 + b^2\rho^2 + 2\mathbb{E}((aX - a\mu)(bY - b\nu)) \\ &= a^2\sigma^2 + b^2\rho^2 + 2\left(ab\mathbb{E}(XY) - ab\mathbb{E}(X)\mathbb{E}(Y)\right) = a^2\sigma^2 + b^2\rho^2 + 2ab\text{Cov}(X, Y) \\ &= a^2\sigma^2 + b^2\rho^2 + 2ab\alpha\end{aligned}$$

Common sense tells us that $\text{Var}(Z) > 0$ (variances can't be negative). Thus the only covariance matrices that are permissable are those that are positive semi-definite.

$$\begin{aligned}Z &\sim \mathbb{N}\left(a\mu + b\nu,\ \begin{bmatrix}a & b\end{bmatrix}\begin{bmatrix}\sigma^2 & \alpha \\ \alpha & \rho^2\end{bmatrix}\begin{bmatrix}a \\ b\end{bmatrix}\right) \\ &\sim \mathbb{N}\left(a\mu + b\nu,\ \mathbf{w}^\intercal\boldsymbol{\Sigma}\mathbf{w}\right) \quad \text{where } \mathbf{w}^\intercal\boldsymbol{\Sigma}\mathbf{w} \geq 0\end{aligned}$$

Notice the similarly to the result $\text{Var}(aX) = a^2\text{Var}(X)$

5.3.1 Hilbert Spaces

Let $\mathcal{H}$ be a vector space over $\mathbb{R}$. The **inner product** $\langle\cdot,\cdot\rangle : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$ iff (i) $\langle\alpha f_1 + \beta f_2, g\rangle = \alpha\langle g, f_1\rangle + \beta\langle g, f_2\rangle$; (ii) $\langle f, g\rangle = \langle g, f\rangle$; (iii) $\langle f, f\rangle \geq 0$ and is 0 only if $f = 0$.

A **Hilbert space** is a space where the inner product is defined in addition to another technical condition relating to Cauchy sequences, which is not necessary to go into here. A **kernel** is defined as follows:

The function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel if there exists a Hilbert space and a map $\phi : \mathcal{X} \to \mathcal{H}$ s.t. $\forall x, x' \in \mathcal{X},\ k(x, x') = \langle\phi(x), \phi(x')\rangle$

5.3.2 Positive Definiteness of Inner Products in Hilbert Spaces

Let $\mathcal{H}$ be any Hilbert space, $\mathcal{X}$ a non-empty set and ϕ a feature mapping. This implies that $k(x, y) = \langle\phi(x), \phi(y)\rangle$ is a positive semidefinite function.

Proof. The norm is written as

$$||f|| = \sqrt{\langle f, f\rangle}$$

$$\begin{aligned}\sum_i\sum_j a_i a_j k(x_i, x_j) &= \sum_j \langle a_i\phi(x_i), a_j\phi(x_j)\rangle \\ &= \langle\sum_i a_i\phi(x_i), \sum_j a_j\phi(x_j)\rangle \\ &= \left\|\sum_i a_\phi(x_i)\right\|^2 \geq 0\end{aligned}$$

□

The reverse direction also holds. A positive semidefinite function is guaranteed to be the inner product in a Hilbert space. Thus positive semidefiniteness is a way of proving a function is a kernel.

5.3.3 Reproducing Kernel Hilbert Space

We now have kernels on feature spaces. We want to define what our functions on $\mathcal{X}$ look like. The space of these functions is known as a reproducing kernel Hilbert space.

Suppose our feature map $\phi : \mathbb{R}^2 \to \mathbb{R}^3$ is, for example, $\phi\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} x_1 \\ x_2 \\ x_1 x_2 \end{bmatrix}$ with a dot-product kernel $k(\mathbf{x}, \mathbf{y}) = x^\intercal y$. This feature space is denoted by $\mathcal{H}$

Let $f(\mathbf{x}) = ax_1 + bx_2 + cx_1x_2$ or $f(\cdot) = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$

We can also express f as $f(\mathbf{x}) = f(\cdot)^\intercal \phi(\mathbf{x}) = \langle f(\cdot), \phi(\mathbf{x}) \rangle$. $\phi(\mathbf{x})$ is a function mapping $\mathbb{R}^2 \to \mathbb{R}^3$ and defines the parameters of a function mapping $\mathbb{R}^2 \to \mathbb{R}$. To illustrate this further, take

$$k(\cdot, \mathbf{y}) = \begin{bmatrix} y_1 \\ y_2 \\ y_1 y_2 \end{bmatrix} = \phi(\mathbf{y})$$

For every $\mathbf{y}$, there is a vector $k(\cdot, \mathbf{y})$ s.t. $\langle k(\cdot, \mathbf{y}), \phi(\mathbf{x}) \rangle = ax_1 + bx_2 + cx_1x_2$ where $a = y_1, b = y_2, c = y_1y_2$

This is equivalent to

$$\langle k(\cdot, \mathbf{x}), \phi(\mathbf{y}) \rangle = uy_1 + vy_2 + wy_1y_2 = k(\mathbf{x},\mathbf{y})$$

So we can write $\phi(\mathbf{x}) = k(\cdot, \mathbf{x})$ and $\phi(\mathbf{y}) = k(\cdot, \mathbf{y})$ without ambiguity.

This shows that:

- every feature mapping is in the feature space: $\forall \mathbf{x} \in \mathcal{X},\ k(\cdot, \mathbf{x}) \in \mathcal{H}$;
- $\forall \mathbf{x} \in \mathcal{X}, \forall f \in \mathcal{H},\ \langle f, k(\cdot, \mathbf{x}) \rangle = f(\mathbf{x})$

This last property is the **reproducing property**. It yields another appealing property: the norm in an RKHS is a natural measure of how complex a function is.

5.3.4 Squared Exponential (RBF)

$$k(x, x') = v^2 \exp\left(-\frac{(x - x')^2}{2l^2}\right) + \sigma^2 \delta_{xx'}$$

Example: l is length scale for a squared exponential covariance function. It's roughly the length between inputs before the covariance is lower. You can have anisotropic RBF kernels where there are multiple length scales, one for each direction in the input space. That way you can accommodate input features that are on different scales.

5.3.5 Periodic

$$k(x, x') = \exp(-\frac{2\sin^2(\pi(x - x'))}{l^2})$$

The x's are first mapped to $u = [sin(x), cos(x)]^\intercal$ and then distances are measured in the u-space. If the length scale is larger than the period, then the covariance is high, whereas in the inverse case, there can be a lot of action within the period.

5.3.6 Reproducing Kernel for Vector-Valued Functions

Our reproducing kernel is now defined as a symmetric function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{D \times D}$ s.t. $k(x, x')$ yields a positive semidefinite **matrix**.

Let $\mathcal{H}$ be the vector-valued RKHS over functions $f : \mathcal{X} \to \mathbb{R}^D$. This means that $\forall \mathbf{x} \in \mathcal{X}, \forall f \in \mathcal{H}, \forall \mathbf{c} \in \mathbb{R}^D$, $f(\mathbf{x}') = k(\mathbf{x}, \mathbf{x}')\mathbf{c}$ belongs to $\mathcal{H}$

and the reproducing property is written now as: $\langle f, k(\cdot, \mathbf{x})\rangle = f(\mathbf{x})^\intercal \mathbf{c}$

5.4 Coregionalization

Coregionalization originated in geostatistics literature, where it is known as *cokriging*. In order for covariance functions to be valid kernels, as seen before, they must be positive semidefinite. Suppose we have a multi-output problem with D outputs. In the *linear model of coregionalization*, these outputs are a written as a linear combination of Q independent latent functions which have zero mean and a covariance function.

For each $d \in \{1, ..., D\}$, the output is determined by the function $f_d(\mathbf{x})$ with p-dimensional input vector $\mathbf{x}$.

$$f_d(\mathbf{x}) = \sum_{q=1}^{Q} a_{d,q} u_q(\mathbf{x})$$

where $u_q(\mathbf{x})$ are the latent functions with covariance $\text{Cov}(u_q(\mathbf{x}), u'_q(\mathbf{x}')) = \begin{cases} k_q(\mathbf{x}, \mathbf{x}') & q = q' \\ 0 & \text{otherwise} \end{cases}$, due to independence. Some of these latent functions can share the same covariance kernel and can be grouped:

$$f_d(\mathbf{x}) = \sum_{q=1}^{Q} \sum_{i=1}^{R_q} a^i_{d,q} u^i_q(\mathbf{x})$$

where $u^i_q(\mathbf{x})$ have covariance $\text{Cov}(u^i_q(\mathbf{x}), u^{i'}_{q'}(\mathbf{x}')) = k_q(\mathbf{x}, \mathbf{x}')$ for $i = i'$, $q = q'$. There are now Q groups of functions, within each one sharing a covariance function.

We can now write the cross-covariance between functions as:

$$\begin{aligned}
\text{Cov}(f_d(\mathbf{x}), f_{d'}(\mathbf{x}')) = (\mathbf{K}(\mathbf{x}, \mathbf{x}'))_{d,d'} &= \sum_{q=1}^{Q} \sum_{q'=1}^{Q} \sum_{i=1}^{R_q} \sum_{i'=1}^{R_q} a^i_{d,q} a^{i'}_{d',q'} \cdot \text{Cov}(u^i_q(\mathbf{x}), u^{i'}_{q'}(\mathbf{x}')) \\
&= \sum_{q=1}^{Q} \sum_{i=1}^{R_q} a^i_{d,q} a^i_{d',q'} \cdot k_q(\mathbf{x}, \mathbf{x}') \quad \text{by independence} \\
&= \sum_{q=1}^{Q} b^q_{d,d'} k_q(\mathbf{x}, \mathbf{x}')
\end{aligned}$$

where $b^q_{d,d'} = \sum_{i=1}^{R_q} a^i_{d,q} a^i_{d',q'}$ which forms a $D \times D$ matrix $\mathbf{B}_q$ called the **coregionalisation matrix**. The rank of $\mathbf{B}_q$, the number of linearly independent row or column vectors, is intuitively determined by R_q.

Writing our kernel $\mathbf{K}(\mathbf{x}, \mathbf{x}') = \sum_{q=1}^{Q} \mathbf{B}_q k_q(\mathbf{x}, \mathbf{x}')$, one can intuit that this is a sum of the products of two kernels (called *separable kernels*), one that models the output dependencies ($\mathbf{B}_q$) and one that models the input dependencies (k_q).

5.5 Relation with Linear in the Parameters Model

Consider $f(x) = ax + b$ where $a \sim \mathcal{N}(0, \alpha)$, $b \sim \mathcal{N}(0, \beta)$. We can work out the mean function (see A.1 for derivation):

$$\mu(x) = \mathbb{E}(f(x)] = \int\int (ax + b) p(a) p(b) \, \mathrm{d}a \, \mathrm{d}b = \int axp(a) \, \mathrm{d}a + \int bp(b) \, \mathrm{d}b = 0$$

And now the covariance function:

$$\begin{aligned} k(x,x') &= \mathbb{E}[(f(x)-0)(f(x')-0)] = \int\int (ax+b)(ax'+b)p(a)p(b)\,\mathrm{d}a\,\mathrm{d}b \\ &= \int\int (a^2xx'+b^2+ab(x+x'))p(a)p(b)\,\mathrm{d}a\,\mathrm{d}b \\ &= \int_b p(b)xx' \int_a a^2p(a) + (x'+x)\int_a\int_b ap(a)bp(b) + \int_a p(a)\int_b b^2p(b) \\ &= \alpha xx' + \beta \end{aligned}$$

So we have, in a very overly complicated way, constructed a linear model. We can now take this finite linear model and go to a Gaussian process (infinite):

$$f(x) = \sum_{m=1}^{M} w_m\phi_m(x) \quad p(\mathbf{w}) = \mathcal{N}(\mathbf{0},\ \mathbf{A})$$

The joint distribution of any vector $\mathbf{f} = [f(x_1), ..., f(x_N)]$ is a multivariate Gaussian and therefore a Gaussian process. The mean function is 0. The covariance:

$$\begin{aligned} k(x_i,x_j) &= \mathrm{Cov}_{\mathbf{w}}(f(x_i),f(x_j)) = \mathbb{E}(f(x_i)f(x_j)) - \mathbb{E}(f(x_i))\mathbb{E}(f(x_j)) = \mathbb{E}(f(x_i)f(x_j)) \\ &= \int ... \int \left(\sum_{k=1}^{M}\sum_{l=1}^{M} w_kw_l\phi_k(x_i)\phi_l(x_j)\right) p(\mathbf{w})\,\mathrm{d}\mathbf{w} = ... \\ &= \sum_{k=1}^{M}\sum_{l=1}^{M} \phi_k(x_i)\phi_l(x_j)\int\int w_kw_lp(w_k,w_l)\,\mathrm{d}w_k\,\mathrm{d}w_l \end{aligned}$$

This shows that any linear in the parameters model with Gaussian prior over weights, is also a Gaussian process. Mercer's theorem states that every GP also corresponds to a linear in the parameters model but not necessarily a finite one.

We will now show a very cool result that a squared exponential covariance function corresponds to a linear in parameters model with infinitely many Gaussian bumps.

Consider the following *Gaussian bump basis function*:

$$f(x) = \lim_{N\to\infty}\frac{1}{N}\sum_{-N/2}^{N/2}\gamma_n \exp\left(-\left(x-\frac{n}{\sqrt{N}}\right)^2\right) \quad \text{where } \gamma_n \sim \mathcal{N}(0,1)$$

We use the sum (limited to infinity) to place bumps everywhere along x.

But $\lim_{N\to\infty}\frac{1}{N}\sum_{-N/2}^{N/2} = \int_{-\infty}^{\infty}$, so

$$\int_{-\infty}^{\infty}\gamma(u)\exp(-(x-u)^2)\,\mathrm{d}u$$

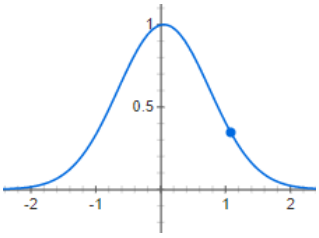


Figure 6: Example Gaussian bump

$$\begin{aligned} \mu(x) &= \mathbb{E}(f(x)) \\ &= \int_{-\infty}^{\infty}\exp(-(x-u)^2)\int_{-\infty}^{\infty}\gamma(u)p(\gamma(u))\,\mathrm{d}\gamma(u)\,\mathrm{d}u = 0 \end{aligned}$$

5.6 Eigenvectors and Relation to PCA

Let's visualise what a 2 variable joint Gaussian distribution (2-D multivariate Gaussian) would look like.

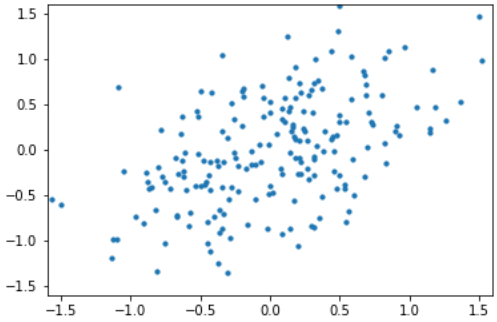


Figure 7: $(x, y) \sim \mathcal{N}\left(\mathbf{0},\ \mathbf{\Sigma} = \begin{bmatrix} 0.2 & 0.4 \\ 0.4 & 0.2 \end{bmatrix}\right)$

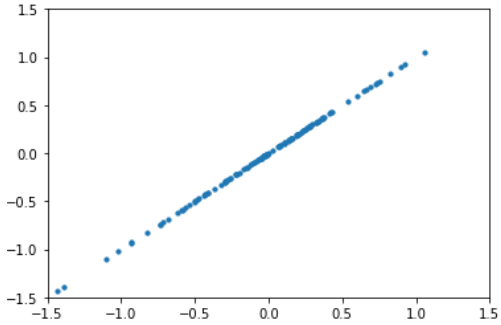


Figure 8: $(x, y) \sim \mathcal{N}\left(\mathbf{0},\ \mathbf{\Sigma} = \begin{bmatrix} 0.2 & 0.2 \\ 0.2 & 0.2 \end{bmatrix}\right)$

5.7 Cholesky Decomposition

Due to the marginalisation/consistency property (or even the definition of a GP), the marginal (and the conditional) distributions are also Gaussian.

Recall $p(\mathbf{f}) = \int p(\mathbf{f}, \mathbf{y})\, \mathrm{d}y$, but now suppose $\mathbf{y}$ is infinitely long.

$$p(\mathbf{x}, \mathbf{y}) = \mathcal{N}\left(\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}, \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^\intercal & \mathbf{C} \end{bmatrix}\right) \implies p(\mathbf{x}) = \mathcal{N}(\mathbf{a}, \mathbf{A})$$

Cholesky factorisation decomposes a positive-definite matrix $\mathbf{A} = \mathbf{R}^\intercal\mathbf{R}$ where $\mathbf{R}$ is upper triangular. Essentially the "square root".

We can sample now from a D-dimensional joint Gaussian with mean vector $\mathbf{m}$ and covariance matrix $\mathbf{K}$.

$$\mathbf{z} = randn(D, 1)$$
$$\mathbf{y} = chol(K)^\intercal\mathbf{z} + \mathbf{m}$$

$$\mathbb{E}\left((\mathbf{y} - \mathbf{m})(\mathbf{y} - \mathbf{m})^\intercal\right) = \mathbb{E}(\mathbf{R}^\intercal\mathbf{z}\mathbf{z}^\intercal\mathbf{R}) = \mathbf{R}^\intercal\mathbb{E}(\mathbf{z}\mathbf{z}^\intercal)\mathbf{R} = \mathbf{R}^\intercal\mathbf{I}\mathbf{R} = \mathbf{K}$$

5.8 Sequential Generation

Another method of generating from the distribution over functions is sequentially, by factorising the joint distribution using the chain rule of probability:

$$p(f_1, ..., f_N | x_1, ..., x_N) = \prod_{n=1}^{N} p(f_n | f_{n-1}, ..., f_1, x_n, ..., x_N)$$

I might not finish this part as it's not crucial and there's a lack of derivation in the slides.

5.9 Gaussian Process Example

$y = f + \epsilon$

$\mathbf{y}$ are our training outputs.

where the likelihood for our data $\mathbf{y}$ is $p(\mathbf{y}|\mathbf{f}) = \mathcal{N}(\mathbf{f}, \sigma^2\mathbf{I})$

Any set of function variables $\mathbf{f}$ has $p(\mathbf{f}|\mathbf{X}) = \mathcal{N}(\mathbf{0}, \mathbf{K})$

The marginal likelihood is $p(\mathbf{y}) = \int p(\mathbf{y}|\mathbf{f})p(\mathbf{f})d\mathbf{f} = \mathcal{N}(\mathbf{0}, \mathbf{K} + \sigma^2\mathbf{I})$

For prediction, consider joint training and test marginal likelihood:

$$p(\mathbf{y}, \mathbf{y}^*) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K_{xx} & K_{xt} \\ K_{tx} & K_{tt} \end{bmatrix}\right)$$

Conditioning on training outputs:

$p(\mathbf{y}^*|\mathbf{y}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

where

$\boldsymbol{\mu} = K_{tx}[K_{xx} + \sigma^2\mathbf{I}]^{-1}\mathbf{y}$

$\boldsymbol{\Sigma} = K_{tt} - K_{tx}K_{xx}^{-1}K_{xt}$

6 Gaussian Process Classification

Now, the likelihood is categorical and we have a new likelihood:

$$p(y|x) = \sigma(f(x))^y(1 - \sigma(f(x))^{1-y})$$

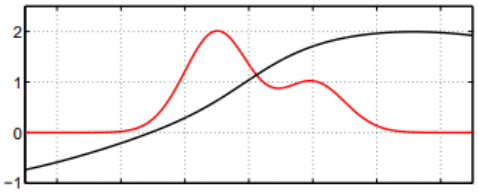
where σ is the sigmoid function.

The integral is no longer tractable.

7 Monte Carlo

How can we integrate an intractable function?

We want to find approximate expectations of a function $\phi(\mathbf{x})$ w.r.t. probability $p(\mathbf{x})$.



$$\mathbb{E}_{p(x)}[\phi(x)] = \bar{\phi} = \int \phi(\mathbf{x})p(\mathbf{x})\,\mathrm{d}\mathbf{x}$$

We could lay out a grid and compute $\int \phi(\mathbf{x})p(\mathbf{x})\,\mathrm{d}\mathbf{x} = \sum_{\tau=1}^{T} \phi(\mathbf{x}^{(\tau)})p(\mathbf{x}^{(\tau)})\Delta\mathbf{x}$. However, this requires too many points if the dimensionality increases (suppose we have a grid of 10 points along each dimension).

7.1 Monte Carlo

If we choose the points from the distribution $p(x)$ then

$$\mathbb{E}_{p(x)}[\phi(x)] \simeq \frac{1}{T}\sum_{\tau=1}^{T} \phi(\mathbf{x}^{(\tau)}) \quad \text{where } \mathbf{x}^{(\tau)} \sim p(x)$$

Furthermore, due to the central limit theorem, the sum of the independent samples yields an unbiased estimate: $\mathbb{V}[\hat{\phi}] = \frac{\mathbb{V}[\phi]}{T}$ so this variance is independent of the dimensionality of $\mathbf{x}$.

This leads to the question of how we sample from $p(\mathbf{x})$? What if it is intractable?

7.2 Markov Chains & Gibbs Sampling

What if we are trying to find the posterior $p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)}$. We don't need the denominator to get the shape of the posterior - we just need the relative value of the posterior at one point versus all others. This involves (for continuous variables) an infinite number of calculations. This is where dependent sampling comes in, and in particular Markov chains. A Markov chain is a sequence defined by a transition function $q(x'|x)$. Gibbs sampling states that if you generate a new x but keep all other components the same, by sampling it from this distribution:

$$x_i' \sim p(x_i|x_1, ..., x_{i-1}, x_{i+1}, ..., x_D)$$

If you iterate this over all the indices, and repeat this process many times then the sample will have the correct distribution. Of course this conditional distributions must be known.

https://www.youtube.com/watch?v=ER3DDBFzH2g

7.3 Example: Step Model

Suppose

$$x_{i\leq\theta} \sim \text{Poisson}(\lambda), \quad x_{i>\theta} \sim \text{Poisson}(\mu)$$

Such that

$$p(x_i) = \frac{e^{-\lambda}\lambda^x}{x!}$$
$$= \exp(x\log\lambda - \lambda - \log(x!)) \iff \log\frac{e^{-\lambda}\lambda^x}{x!} = -\lambda + x\log\lambda - \log(x!)$$

with priors:

$$\lambda \sim \text{Gamma}(a, b), \quad \mu \sim \text{Gamma}(a, b), \quad \theta \sim \mathcal{U}(1851, 1961)$$

$$p(\lambda) = \frac{1}{\Gamma(a)} b^a \lambda^{a-1} e^{-b\lambda}$$

But since $\log\left(\frac{1}{\Gamma(a)} b^a \lambda^{a-1} e^{-b\lambda}\right) = -\log\Gamma(a) + a\log b + (a-1)\log\lambda - b\lambda$:

$$\begin{aligned} p(\lambda) &= a\log b - \log\Gamma(a) + (a-1)\log\lambda - b\lambda \\ p(\mu) &= a\log b - \log\Gamma(a) + (a-1)\log\mu - b\mu \end{aligned}$$

We wish to find the posterior, so we can use Bayes' theorem:

$$\begin{aligned} p(\theta,\lambda,\mu|x) &\propto p(x|\theta,\lambda,\mu)p(\theta,\lambda,\mu) \\ &= p(x_{i\leq\theta}|\theta,\lambda)\cdot p(x_{i>\theta}|\theta,\mu)\cdot p(\theta,\lambda,\mu) \\ &= \prod_{i\leq\theta}\exp(x_i\log\lambda - \lambda - \log(x_i!))\prod_{i>\theta}\exp(x_i\log\mu - \mu - \log(x_i!))\cdot p(\theta,\lambda,\mu) \\ \log p(\theta,\lambda,\mu|x) &\propto \sum_{i\leq\theta}(x_i\log\lambda - \lambda - \log(x_i!)) + \sum_{i>\theta}(x_i\log\mu - \mu - \log(x_i!)) + \log p(\theta,\lambda,\mu) \end{aligned}$$

To find the conditional posteriors for the parameters, we ignore terms which do not include that parameter.

$$\begin{aligned} \log p(\lambda|x,\theta,\mu) &= \sum_{i\leq\theta}(x_i\log\lambda - \lambda - \log(x_i!)) + a\log b - \log\Gamma(a) + (a-1)\log\lambda - b\lambda \\ &\propto \sum_{i\leq\theta}(x_i\log\lambda - \lambda) + (a-1)\log\lambda - b\lambda \\ &\propto (a-1+\sum_{i\leq\theta}(x_i))\log\lambda - \sum_{i\leq\theta}\lambda - b\lambda \\ &\propto (a-1+\sum_{i\leq\theta}(x_i))\log\lambda - (\theta+b)\lambda \\ &\propto \log\text{Gamma}(a+\sum_{i\leq\theta}(x_i), \theta+b) \end{aligned}$$

Similarly;

$$\begin{aligned} \log p(\mu|x,\lambda,\theta) &\propto (a-1+\sum_{i>\theta}(x_i))\log\mu - (N-\theta+b)\mu \\ &\propto \log\text{Gamma}(a+\sum_{i\leq\theta}(x_i), N-\theta+b) \end{aligned}$$

It is worth noting at this point that the Gamma and Poisson are conjugates.

$$\begin{aligned} \log p(\theta|x,\lambda,\mu) &\propto \sum_{i\leq\theta}(x_i\log\lambda - \lambda - \log(x_i!)) + \sum_{i>\theta}(x_i\log\mu - \mu - \log(x_i!)) \\ &\propto \sum_{i\leq\theta}(x_i\log\lambda) - \theta\lambda + \sum_{i>\theta}(x_i\log\mu) - (N-\theta)\mu \end{aligned}$$

This is not a standard distribution, but simple enough to sample. Sample it from $i = 0 : N$ and use it to construct a multinomial distribution

8 Ranking

8.1 Towards Probabilistic Ranking

Suppose we have player 1 and player 2, with skills w_1, w_2 respectively. The skill difference is therefore $s = w_1 - w_2$. The performance may not be perfectly consistent though, so we can add noise:

$$t = s + n \quad \text{where } n \sim \mathcal{N}(0,1)$$

The game outcome is given as $y = sign(t) = \begin{cases} +1 & \text{player 1 wins} \\ -1 & \text{player 2 wins} \end{cases}$

$$p(t|w_1, w_2) = \mathcal{N}(w_1 - w_2, 1)$$
$$p(y = 1|w_1, w_2) = p(t > 0|w_1, w_2) = \Phi(w_1 - w_2) \quad \text{where } \Phi \text{ is the cum. dist.}$$

We can now construct the likelihood:

$$p(\mathbf{y}|w_1, w_2) = \Phi(y(w_1 - w_2))$$

We can also write the likelihood as this kind of chain:

$$p(\mathbf{y}|w_1, w_2) = \iint p(y|t)p(t|s)p(s|w_1, w_2)\,\mathrm{d}t\,\mathrm{d}s$$

Recall Bayes' rule:

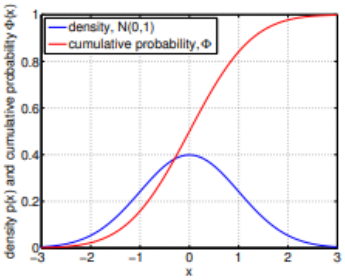
$$p(\mathbf{w}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{w})p(\mathbf{w})}{p(\mathbf{y})}$$
$$p(w_1, w_2|y) = \frac{p(y|w_1, w_2)p(w_1)p(w_2)}{p(y)} \qquad \text{our data is } y$$
$$= \frac{p(y|w_1, w_2)p(w_1)p(w_2)}{\iint p(w_1)p(w_2)p(y|w_1, w_2)\,\mathrm{d}w_1\,\mathrm{d}w_2}$$
$$= \frac{\Phi(y(w_1 - w_2))\mathcal{N}(w_1|\mu_1, \sigma_1^2)\mathcal{N}(w_2|\mu_2, \sigma_2^2)}{\iint \Phi(y(w_1 - w_2))\mathcal{N}(w_1|\mu_1, \sigma_1^2)\mathcal{N}(w_2|\mu_2, \sigma_2^2)\,\mathrm{d}w_1\,\mathrm{d}w_2}$$

where we set the prior to be $p(w_i) = \mathcal{N}(w_i|\mu_i, \sigma^2)$

Everytime one has a game and an outcome, then skills are correlated, since one player will win, so their skill will be higher. This posterior does not have a closed form, since it is not Gaussian.

The prior over y, the normalising constant, the model evidence, the marginal likelihood, does have a closed form:

$$p(y) = \Phi\left(\frac{y(\mu_1 - \mu_2)}{\sqrt{1 + \sigma_1^2 + \sigma_2^2}}\right)$$



Consider if we are very uncertain about the skills, then the argument to the cumulative distribution gets closer to zero, so the probability of the outcome gets closer to 50%.

8.2 Gibbs Sampling in TrueSkill

Suppose for games $g = 1, ..., G$ we have a the variable I_g and J_g for the id of the first and second player. The outcome is $y = \begin{cases} +1 & I_g \text{ wins,} \\ -1 & \text{otherwise} \end{cases}$.

We first initialise $\mathbf{w}$ from a prior $p(\mathbf{w})$. We then need the performance differences:

$$p(t_g|w_{I_g}, y_g) \propto \delta(y_g - sign(t_g))\mathcal{N}(w_{I_g} - w_{J_g}, 1)$$

The conditional distribution of the performance differences as above is a univariate truncated Gaussian, sampled either using rejection sampling or "inverse transformation" method.

Jointly sample the skills:

$$p(\mathbf{w}|\mathbf{t}, \mathbf{y}) = p(\mathbf{w}|\mathbf{t}) \propto p(\mathbf{w}) \prod_{g=1}^{G} p(t_g|w_{I_g}, w_{J_g})$$

But $t = s + \mathcal{N}(0, 1) \implies p(t_g|w_{I_g}, w_{J_g}) \propto \mathcal{N}(\mathbf{w}; w_{I_g} - w_{J_g}, 1)$

Let $\mu_g = w_{I_g} - w_{J_g}$ and $t_g = \mu_1 - \mu_2$. Refer to appendix for an expansion which doesn't show much.

For Gibbs sampling, we then iterate back to calculating the performance differences.

When one conditions on a random variable, the value of the random variable is fixed, which simplifies things.

The product of two Gaussians yields an unnormalised Gaussian:

$$\mathcal{N}(\mu_a, \Sigma_a)\mathcal{N}(\mu_b, \Sigma_b) = z_c\mathcal{N}(\mu_c, \Sigma_c)$$

$$\Sigma_c^{-1} = \Sigma_a^{-1} + \Sigma_b^{-1} \qquad \mu_c = \Sigma_c(\Sigma_a^{-1}\mu_a + \Sigma_b^{-1}\mu_b)$$

Suppose $p(\mathbf{w}) \sim \mathcal{N}(\mu_0, \Sigma_0)$. Using the above results of multiplying Gaussians:

$$\implies \Sigma^{-1} = \Sigma_0^{-1} + \sum_{g=1}^{G} \Sigma_g^{-1}$$

$$\mu =$$

$$\begin{aligned} p(t_g|w_{I_g}, w_{J_g}) &\propto \exp(-\frac{1}{2}(w_{I_g} - w_{J_g} - t_g)^2) \\ &\propto \mathcal{N}(-\frac{1}{2}\begin{bmatrix} w_{I_g} - \mu_1 & w_{J_g} - \mu_2 \end{bmatrix}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\begin{bmatrix} w_{I_g} - \mu_1 \\ w_{J_g} - \mu_2 \end{bmatrix}, 1) \end{aligned}$$

An alternative to Gibbs sampling for TrueSkill is message passing on graphs.

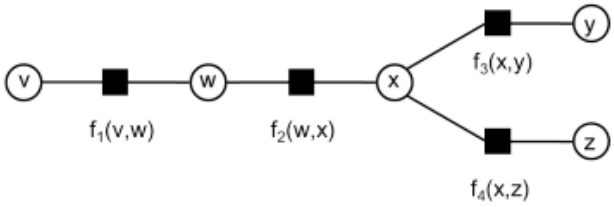
8.3 Message Passing on Factor Graphs

Factor graphs are a type of *probabilistic graphical model*.

We can lay out the probabilities in a factor graph. Suppose:

$$p(v, w, x, y, z) = f_1(v, w)f_2(w, x)f_3(x, y)f_4(x, z)$$

We can now ask questions like what are the marginal distributions, conditional distributions, $p(w)$?



$$p(w) = \sum_v \sum_x \sum_y \sum_z f_1(v, w)f_2(w, x)f_3(x, y)f_4(x, z)$$

Computing this is K^4 sums (where K is the number of values the variables can take) and K possible values of w therefore $\mathcal{O}(K^5)$. We can break up the factor graph above into two subgraphs split at w.

$$\begin{aligned} p(w) &= \sum_v f_1(v,w) \sum_x \sum_y \sum_z f_2(w,x) f_3(x,y) f_4(x,z) \quad \text{and while we're here...} \\ &= \sum_v f_1(v,w) \sum_x f_2(w,x) \sum_y f_3(x,y) \sum_z f_4(x,z) \end{aligned}$$

We call these components **messages**

$$\begin{aligned} \mathrm{m}_{f_1 \to w}(w) &= \sum_v f_1(v,w) \\ \mathrm{m}_{f_2 \to w}(w) &= \sum_x \sum_y \sum_z f_2(w,x) f_3(x,y) f_4(x,z \\ &= \sum_x f_2(w,x) \sum_y \sum_z f_3(x,y) f_4(x,z) \\ &= \sum_x f_2(w,x) \mathrm{m}_{x \to f_2}(x) \end{aligned}$$

$$p(w) = \mathrm{m}_{f_1 \to w}(w) \cdot \sum_x \mathrm{m}_{f_3 \to x}(x) \cdot \mathrm{m}_{f_4 \to x}(x)$$

So nodes take incoming messages and passes them on.

In summary message passing involves three update equations:

- Marginals are the product of all incoming messages from neighbour factors

$$p(t) = \prod_{f \in F_t} \mathrm{m}_{f \to t}(t)$$

- Messages from factors sum out all variables except the receiving one
- Messages from variables are the product of all incoming messages except the message from the receiving factor

$$\mathrm{m}_{t \to f}(t) = \frac{p(t)}{\mathrm{m}_{f \to t}(t)}$$

The benefits of this are partial and localised computations.

9 Expectation-Maximisation

9.1 Gaussian Mixture Models

In a GMM, the parameters are $\theta = \{\mu_j, \sigma_j^2, \pi_j\}_{j=1...k}$. We have one latent variable for each datapoint z_i, an assignment to a class. GMMs can represent any distribution. We wish to find:

$$\arg\max_\theta p(X|\theta) = \prod_i^N p(x_i|\theta) = \prod_i^N \pi_1 \mathcal{N}(x_i|\mu_1, \Sigma_1) + ...$$

subject to $\sum_c^K \pi_c = 1$ and $\Sigma_i \succ 0$. This psd constraint is quite tough. So we introduce the EM algorithm for efficiently training these models.

We represent the GMM as a latent variable problem, where we introduce a latent variable z for each datapoint such that $p(z = c|\theta) = \pi_c$. We assign a Gaussian prior on data. We now marginalise away the latent variable:

$$p(x|z = c, \theta) = \mathcal{N}(x|\mu_c, \Sigma_c)$$
$$p(x|\theta) = \sum_{c=1}^K p(x|z = c, \theta)p(z = c|\theta)$$

which gives us the same likelihood result as without the latent variable, z, which we refer to as the source.

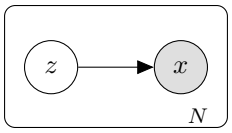


Figure 9: Graphical model for the GMM.

The idea of the EM algorithm is that we keep iterating between keeping the parameters and the sources fixed.

Keeping the sources fixed, we can calculate the parameters:

$$p(x|z = 1, \theta) = \mathcal{N}(x|\mu_1, \sigma_1^2)$$

where $\mu_1 = \frac{\sum_{i \in \mathcal{I}_1} x_i}{|\mathcal{I}_1|}$, $\sigma_1^2 = \frac{\sum_{i \in \mathcal{I}_1} (x_i - \mu_1)^2}{|\mathcal{I}_1|}$, and $p(z_i = j|\theta) = \pi_j$

However, we can use soft assignments too, where $\mu_1 = \frac{\sum_i p(z_i=1|x_i,\theta)x_i}{p(z_i=1|x_i,\theta)}$

Keeping the parameters fixed, we can work out the sources. Suppose our parmeters are $p(x|z = 1, \theta) = \mathcal{N}(-2, 1)$, and we want $p(z = 1|x, \theta)$,

$$p(z = 1|x, \theta) = \frac{p(z = 1|\theta)p(x|z = 1, \theta)}{p(x|\theta)}$$

The normalising constant can be determined explicitly: it is marginalising over just K terms.

9.2 Mathematical Machinery

9.2.1 Jensen's Inequality

A function f is concave is such that all function values between any two points are greater or equal to the value of the line at that point which joints the two points:

$$\forall a, b, \alpha : f(\alpha a + (1 - \alpha)b) \geq \alpha f(a) + (1 - \alpha)f(b),$$

where $\alpha \in [0,1]$ represents the distance along the line between a and b. This generalises to multiple αs:

$$f\left(\mathbb{E}_{p(z)} z\right) \geq \mathbb{E}_{p(z)} f(z) \quad \text{(Jensen's inequality)}$$

9.2.2 Kullback-Leibler Divergence

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

This is an asymmetric and non-negative metric. Proof:

$$-\mathcal{KL}(p||q) = \mathbb{E}_p\left(-\log \frac{p}{q}\right) = \mathbb{E}_p\left(\log \frac{q}{p}\right)$$

$$\mathbb{E}_p\left(\log \frac{p}{q}\right) \leq \log\left(\mathbb{E}\frac{p}{q}\right) = \log \int p(x)\frac{p(x)}{q(x)} dx = 0 \quad \text{using Jensen's inequality}$$

9.2.3 Mutual Information

$$\mathcal{MI}(X,Y) = \mathcal{KL}(p_{X,Y}, p_x p_y)$$

Todo look at HSIC

9.3 Expectation Maximisation Algorithm

This algorithm is for probabilistic models with latent variables, observed variables, and parameters.

We can write out the likelihood from the graphical model in Figure 10, by marginalising the joint:

$$p(x_i|\theta) = \frac{p(x_i|z_i = c, \theta)p(z_i = c|\theta)}{p(z_i = c|x_i, \theta)} = \sum_{c=1}^{K} p(x_i|z_i = c, \theta)p(z_i = c|\theta)$$

and now we wish to compute the maximum marginal likelihood:

$$\sum_{i=1}^{N} \log \sum_{c=1}^{K} p(x_i|z_i = c, \theta)p(z_i = c|\theta)$$

we could maximise with a stochastic optimizer, but it's not very efficient.

$$\sum_{i=1}^{N} \log \sum_{c=1}^{K} \frac{q(z_i = c)}{q(z_i = c)} p(x_i, z_i = c|\theta) \;\geq\; \sum_{i=1}^{N} \sum_{c=1}^{K} q(z_i = c) \log \frac{p(x_i, z_i = c|\theta)}{q(z_i = c)}$$

This q is known as the variational lower bound. In order to illustrate what maximising this lower bound does, we start with Bayes' rule:

$$p(x|\theta) = \frac{p(x|z,\theta)p(z|\theta)}{p(z|x,\theta)} = \frac{p(x|z,\theta)p(z|\theta)}{q(z)} \frac{q(z)}{p(z|x,\theta)}$$
$$\log p(x|\theta) = \log \frac{p(x|z,\theta)p(z|\theta)}{q(z)} + \log \frac{q(z)}{p(z|x,\theta)}$$

Finally we average both sides w.r.t. $q(z)$:

$$\log p(x|\theta) = \int q(z) \log \frac{p(x|z,\theta)p(z|\theta)}{q(z)} \,\mathrm{d}z + \int q(z) \log \frac{q(z)}{p(z|x,\theta)} \,\mathrm{d}z$$

The first term is the **lower-bound functional** and the second term is the $\mathcal{KL}$**-divergence**. The lower-bound functional is a lower bound since the second term is always non-negative.

We now have the log-marginal-likelihood as a sum of two terms. But how do we select q? We do this using the EM algorithm. The EM algorithm is as follows:

- Initialise randomly $\theta^{t=0}$, then for $t = 1...T$:
- **E-step**: for fixed θ^{t-1} maximise the lower-bound wrt $q(z)$

  Since the marginal likelihood term $\log p(x|\theta)$ does not depend on $q(z)$, maximising the lower-bound consists of minimising the $\mathcal{KL}$-divergence, by setting $q^t(t) = p(z|x, \theta^{t-1})$
- **M-step**: for fixed $q^t(z)$, maximise the lower-bound wrt θ

$$\arg\max_\theta \int q(z) \log \frac{p(x|z,\theta)p(z|\theta)}{q(z)} \,\mathrm{d}z = \arg\max_\theta \int q(z) \log p(y|z,\theta)p(z|\theta) \,\mathrm{d}z - \int q(z) \log q(z) \,\mathrm{d}z$$

The second term is the entropy of $q(z)$ but is not dependent on θ so throw it away.

$$\theta^t = \arg\max_\theta \int q^t(z) \log p(y|z,\theta)p(z|\theta) \,\mathrm{d}z$$

9.4 Example: Text Modelling

This section covers three models for modelling text documents, with increasing complexity. Starting with simple bag-of-words models, moving to the EM algorithm and finally Latent Dirichlet Allocation.

Symmetric Dirichlet

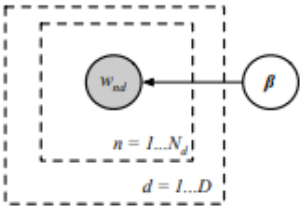
A symmetric Dirichlet simply has all $\alpha_i = \alpha \quad \forall i$ identical.

9.4.1 Document Models

The general layout of the problem is as follows.

- There are D documents,
- with a vocab size of M words,
- N_d is the word count in doc d,
- w_{nd} the n-th word in doc d. $\sim \text{Cat}(\boldsymbol{\beta})$ where $\boldsymbol{\beta}$ are the parameters of the multinomial distribution (probabilities of each word).

The simplest model is shown on the right. Estimate $\hat{\boldsymbol{\beta}}$ using MLE:



$$\hat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta}} \prod_{d<D} \prod_{n<N_d} \text{Cat}(w_{nd}|\boldsymbol{\beta}) = \arg\max_{\boldsymbol{\beta}} \text{Mult}(c_1, ..., c_M|\boldsymbol{\beta}, N)$$

$$\implies \hat{\beta}_m = \frac{c_m}{N}$$

where N is total number of words in corpus, c_m is total count of word m.

$$p(\mathbf{w}|\boldsymbol{\beta}) = \prod_{d<D} \prod_{n<N_d} \beta_{w_{nd}} \implies \log p(\mathbf{w}|\boldsymbol{\beta}) = \sum_{m=1}^{M} c_m \log \beta_m$$

$\beta_m^{c_m}$ is the β for $m = nd$, n-th word in the d-th document.

We need to ensure the β's normalises to 1, so we invoke Lagrange:

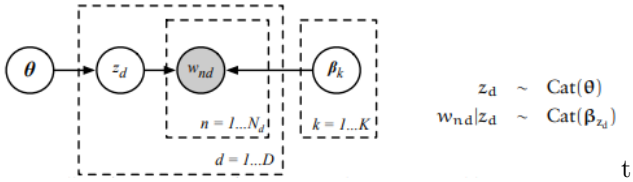
$$F = \sum_{m=1}^{M} c_m \log \beta_m + \lambda(1 - \sum_{m=1}^{M} \beta_m)$$

Taking derivatives and set to zero:

$$\frac{\partial F}{\partial \beta_m} = \frac{c_m}{\beta_m} - \lambda \implies \beta_m = \frac{c_m}{\lambda} \quad \text{and} \quad \frac{\partial F}{\partial \lambda} = 0 \implies \sum_{m=1}^{M} \beta_m = 1$$

This model predicts $\beta_m = c_m/n$ where n is total number of words. This is intuitive but limited since all documents are modelled by the same global word frequency distribution, but we want a model which accommodates different topics of documents.

where $z_d \in \{1, ..., K\}$ are the latent variables which assign document d to one of K topics, $\theta_k = p(z_d = k)$ and $\boldsymbol{\theta}$ is the parameter to the categorical distribution.



We will try to find the likelihood:

$$\begin{aligned}
p(\mathbf{w}|\boldsymbol{\theta}, \boldsymbol{\beta}) &= \prod_{d<D} p(\mathbf{w_d}|\boldsymbol{\theta}, \boldsymbol{\beta}) \\
&= \prod_{d<D} \sum_{k=1}^{K} p(\mathbf{w_d}, z_d = k|\boldsymbol{\theta}, \boldsymbol{\beta}) \quad \text{marginalising the } z_k \\
&= \prod_{d<D} \sum_{k=1}^{K} p(z_d = k|\boldsymbol{\theta}) p(\mathbf{w_d}|z_d = k, \boldsymbol{\beta}) \\
&= \prod_{d<D} \sum_{k=1}^{K} p(z_d = k|\boldsymbol{\theta}) \prod_{n_{N_d}} p(\mathbf{w_{nd}}|z_d = k, \boldsymbol{\beta})
\end{aligned}$$

How can we work with probabilistic models with latent variables and parameters?

10 Variational Inference

Recall our GMM model:

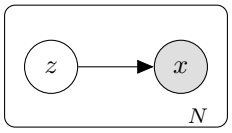


Figure 10: Graphical model for the GMM.

We are interested in the **posterior distribution** of the latent variable given the observations: $p(z|x)$

This is often **intractable**, why???

VI approximates the intractable distribution with a simpler one. The parameters of which are called **variational parameters** and are optimised. One such objective function is the ELBO which we came across in Section 9. Supposing we use a Gaussian posterior approximation, we would optimise the mean and deviation parameters **for each observation**. Variational parameters, therefore, scale with the dataset size.

10.1 Amortizing Variational Inference

Instead of optimising the free variational parameters directly, we can create a parameterised function that maps from observation space to variational parameter space. We end up with a constant number of variational parameters. The downside is that this results in less expressivity than free optimisation.

10.2 General Form

To give a more general comment on variational inference, we take:

$$\begin{aligned}\log p(x) &= \log \int p(x|z)p(z)dz = \log \int p(x,z)\frac{q(z)}{q(z)}dz \\ &= \log \mathbb{E}\left(\frac{p(x,z)}{q(z)}\right) \\ &\geq \mathbb{E}\left(\log \frac{p(x,z)}{q(z)}\right) = \int q(z) \log \frac{p(x|z)p(z)}{q(z)}dz\end{aligned}$$

which can be further decomposed into the lower bound and the $\mathcal{KL}$-divergence:

$$\begin{aligned}\int q(z) \log \frac{p(x|z)p(z)}{q(z)}dz &= \int q(z) \log p(x|z)dz - \int q(z) \log \frac{q(z)}{p(z)} \\ &= F(q) - \mathcal{KL}(q||p)\end{aligned}$$

10.2.1 Mean Field Approximation

This is a method of finding this $\mathcal{KL}$ term. We select a family of distributions $Q = \{q|q(z) = \prod_{i=1}^{d} q_i(z_i)\}$

$$\min_{q_k} \mathcal{KL}(\prod_{i=1}^{d} q_i||p) = \int \prod_{i=1}^{d} q_i \log \frac{\prod_{i=1}^{d} q_i}{p} dz = \int \prod_{j=1}^{d} q_j \log \prod_{i=1}^{d} q_i dz - \int \prod_{i=1}^{d} q_i \log p \, dz$$
$$= \sum_{i=1}^{d} \int \prod_{j=1}^{d} q_j \log q_i - \int \prod_{i=1}^{d} q_i \log p \, dz$$
$$= \int \prod_{j=1}^{d} q_j \log q_k dz + \sum_{i \neq k}^{d} \int \prod_{j=1}^{d} q_j \log q_j - \int \prod_{i=1}^{d} q_i \log p \, dz$$

but $\int \prod_{j=1}^{d} q_j \log q_k dz = \int q_k \log q_k \underbrace{\int \prod_{j \neq k}^{d} q_j dz_{\neq k} \, dz_k}_{1 \times \ldots \times 1 = 1}$ and since we are optimising w.r.t q_k:

$$\implies \min_{q_k} \mathcal{KL}(\prod_{i=1}^{d} q_i||p) = \int q_k \log q_k dz_k - \int \prod_{i=1}^{d} q_i \log p \, dz$$

following a similar procedure for the second term

$$= t.b.c$$

10.3 Examples

10.3.1 Inducing Point Approximation for Gaussian Processes

We will now go through the inducing point approximation which is commonly used with multi-output GPs. With D outputs, let y_d be the dth Gaussian process, such that $y_d = \mathbf{f}_d + \epsilon$ and $\epsilon \sim \mathcal{N}(0, \beta^2)$. The inducing points are $\mathbf{u}_d$. First, we apply the general form of VI from Section 10.2:

$$\log p(Y) = \log \int p(Y|X)p(X)dX \geq \int q(X) \log p(Y|X)dX - \int q(X) \log \frac{q(X)}{p(X)} dX \tag{3}$$
$$= F(q) - \mathcal{KL}(q||p) \tag{4}$$

We use the variational distribution $q(\mathbf{f}_d, \mathbf{u}_d) = p(\mathbf{f}_d|\mathbf{u}_d)\phi(\mathbf{u}_d)$. The marginal likelihood is now:

$$\log p(y_d|X) = \log \iint p(y_d|\mathbf{f}_d)p(\mathbf{f}_d|\mathbf{u}_d)p(\mathbf{u}_d)d\mathbf{f}_d d\mathbf{u}_d$$
$$= \log \iint \frac{p(y_d|\mathbf{f}_d)p(\mathbf{f}_d|\mathbf{u}_d)p(\mathbf{u}_d)p(\mathbf{f}_d|\mathbf{u}_d)\phi(\mathbf{u}_d)}{p(\mathbf{f}_d|\mathbf{u}_d)\phi(\mathbf{u}_d)} d\mathbf{f}_d d\mathbf{u}_d$$

Now apply Jensen's inequality where $\mathbb{E}$ is over variational distribution

$$\log p(y_d|X) \geq \iint p(\mathbf{f}_d|\mathbf{u}_d)\phi(\mathbf{u}_d) \log \frac{p(y_d|\mathbf{f}_d)p(\mathbf{f}_d|\mathbf{u}_d)p(\mathbf{u}_d)}{p(\mathbf{f}_d|\mathbf{u}_d)\phi(\mathbf{u}_d)} d\mathbf{f}_d d\mathbf{u}_d \tag{5}$$
$$= \int \phi(\mathbf{u}_d) \left[\int p(\mathbf{f}_d|\mathbf{u}_d) \log p(y_d|\mathbf{f}_d) d\mathbf{f}_d + \log \frac{p(\mathbf{u}_d)}{\phi(\mathbf{u}_d)} \right] d\mathbf{u}_d \tag{6}$$
$$= \int \phi(\mathbf{u}_d) \underbrace{\mathbb{E}_{p(\mathbf{f}_d|\mathbf{u}_d)}[\log p(y_d|\mathbf{f}_d)]}_{\leq \log \mathbb{E}_{p(\mathbf{f}_d|\mathbf{u}_d)}[p(y|\mathbf{f}_d)] = \log p(y|\mathbf{u}_d)} d\mathbf{u}_d + \underbrace{\int \phi(\mathbf{u}_d) \log \frac{p(\mathbf{u}_d)}{\phi(\mathbf{u}_d)} d\mathbf{u}_d}_{\mathcal{KL}(\phi||p)} \tag{7}$$
$$= \mathcal{L}_1 \tag{8}$$

- Note that $p(\mathbf{f}_d|\mathbf{u}_d) = \mathcal{N}(\mathbf{f}_d|\boldsymbol{\alpha}_d, K_{NN} - K_{NM}K_{MM}^{-1}K_{MN})$ and $p(y_d|\mathbf{f}_d) = \mathcal{N}(0, \sigma^2)$.

- Note that the brown expectation is a lower bound on the conditional $p(y|\mathbf{u}_d)$. With a Gaussian likelihood, $p(y|\mathbf{u}_d)$ is tractable, but in $O(N^3)$ compared with $O(M^3)$ for the lower bound.
- $\mathcal{L}_1$ is the lower bound from Titsias and Lawrence [2010], Hensman et al. [2013].

We take the trace of the log Gaussian since the trace of a scalar is the scalar.

$$\begin{aligned}
\mathbb{E}_{p(\mathbf{f}_d|\mathbf{u}_d)}[\log p(y_d|\mathbf{f}_d)] &= \mathbb{E}_{p(\mathbf{f}_d|\mathbf{u}_d)}\left[-\frac{N}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\left(y_d^\intercal y_d - 2\mathbf{f}_d^\intercal y + \mathbf{f}_d^\intercal \mathbf{f}_d\right)\right] \\
&= -\frac{N}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2} tr\left(y_d^\intercal y_d - 2\mathbb{E}_{p(\mathbf{f}_d|\mathbf{u}_d)}\left[\mathbf{f}_d^\intercal y\right] + \underbrace{\mathbb{E}_{p(\mathbf{f}_d|\mathbf{u}_d)}\left[\mathbf{f}_d^\intercal \mathbf{f}_d\right]}_{\text{Cov}[f]+\alpha\alpha^\intercal}\right) \\
&= \underbrace{-\frac{N}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2} tr\left(y_d^\intercal y_d - 2\boldsymbol{\alpha}_d^\intercal y + \boldsymbol{\alpha}_d\boldsymbol{\alpha}_d^\intercal\right)}_{\log \mathcal{N}(y_d|\boldsymbol{\alpha}_d,\sigma^2)} + K_{NN} - K_{NM}K_{MM}^{-1}K_{MN}\Big) \\
&= \log \mathcal{N}(y_d|\boldsymbol{\alpha}_d, \sigma^2) - \frac{1}{2\sigma^2} tr\left(K_{NN} - \underbrace{K_{NM}K_{MM}^{-1}K_{MN}}_{Q_{nn}}\right) \qquad (9)
\end{aligned}$$

where $\boldsymbol{\alpha}_d = K_{NM}K_{MM}^{-1}\mathbf{u}_d$. We now plug Eq. 9 into Eq. 7, giving us Eq. 13 from Titsias and Lawrence [2010]:

$$\mathcal{L}_1 = \int \phi(\mathbf{u}_d) \log \frac{\mathcal{N}(y_d|\boldsymbol{\alpha}_d, \sigma^2)p(\mathbf{u}_d)}{\phi(\mathbf{u}_d)} d\mathbf{u}_d - \frac{1}{2\sigma^2} tr\left(K_{NN} - Q_{nn}\right) \qquad (10)$$

How do we acquire the optimal variational distribution?

1. We could take the (functional) derivative of $\mathcal{L}_1$ w.r.t. $\phi(\mathbf{u}_d)$, set it to zero, and plug that back into $\mathcal{L}_1$.

$$\begin{aligned}
0 = \frac{d\mathcal{L}_1}{d\phi} &= \frac{d}{d\phi}\left[\int \phi(\mathbf{u}_d) \log \frac{\mathcal{N}(y_d|\boldsymbol{\alpha}_d, \sigma^2)p(\mathbf{u}_d)}{\phi(\mathbf{u}_d)} d\mathbf{u}_d - \frac{1}{2\sigma^2} tr\left(K_{NN} - Q_{nn}\right)\right] && (11) \\
&= \frac{d}{d\phi}\int \phi \log \frac{A}{\phi} d\mathbf{u}_d && (12) \\
&= \int \frac{d}{d\phi}\phi \log \frac{A}{\phi} d\mathbf{u}_d = \int \frac{d}{d\phi}\left[\phi \log A - \phi \log \phi\right] d\mathbf{u}_d && (13) \\
&= \int \log A - \left[\log \phi + \phi \frac{1}{\phi}\right] d\mathbf{u}_d = \int \log \frac{A}{\phi} - 1 \, d\mathbf{u}_d && (14) \\
&= \int \log \frac{A}{\phi} - \log e \, d\mathbf{u}_d && (15) \\
\int \log \frac{A}{\phi} d\mathbf{u}_d &= \int \log e \, d\mathbf{u}_d && (16) \\
\frac{A}{\phi} = e \implies \phi &\propto A = \mathcal{N}(y_d|\boldsymbol{\alpha}_d, \sigma^2)p(\mathbf{u}_d) && (17)
\end{aligned}$$

2. Another way is to reverse the Jensen's inequality, which we shall see now.

$$\mathcal{L}_1 = \underbrace{\int \phi(\mathbf{u}_d) \log \frac{\mathcal{N}(y_d|\boldsymbol{\alpha}_d, \sigma^2)p(\mathbf{u}_d)}{\phi(\mathbf{u}_d)} d\mathbf{u}_d}_{} - \frac{1}{2\sigma^2} tr\left(K_{NN} - Q_{nn}\right) \tag{18}$$

$$\leq \log \int \phi(\mathbf{u}_d) \frac{\mathcal{N}(y_d|\boldsymbol{\alpha}_d, \sigma^2)p(\mathbf{u}_d)}{\phi(\mathbf{u}_d)} d\mathbf{u}_d \quad \text{by reverse Jensen's inequality}$$

$$\implies \mathcal{L}_1 \leq \log \int \mathcal{N}(y_d|\boldsymbol{\alpha}_d, \sigma^2)p(\mathbf{u}_d)d\mathbf{u}_d - \frac{1}{2\sigma^2} tr\left(K_{NN} - Q_{nn}\right) \tag{19}$$

$$= \log \int \mathcal{N}(y_d|\boldsymbol{\alpha}_d, \sigma^2)\mathcal{N}(u|0, K_{MM})d\mathbf{u}_d - \frac{1}{2\sigma^2} tr\left(K_{NN} - Q_{nn}\right) \tag{20}$$

$$= \log \int \mathcal{N}(y_d|\boldsymbol{\alpha}_d, \sigma^2) \underbrace{\mathcal{N}(K_{NM}K_{MM}^{-1}u|0, K_{NM}K_{MM}^{-1}K_{MN})}_{\text{scaled Gaussian: } \mathrm{Cov}(BX)=B\mathrm{Cov}(X)B^\top} d\mathbf{u}_d - \frac{1}{2\sigma^2} tr\left(K_{NN} - Q_{nn}\right) \tag{21}$$

$$= \log \int \mathcal{N}(y_d|0, Q_{NN} + \sigma^2 I)d\mathbf{u}_d - \frac{1}{2\sigma^2} tr\left(K_{NN} - Q_{NN}\right) \tag{22}$$

$$= \log \mathcal{N}(y|0, Q_{NN} + \sigma^2 I+) - \frac{1}{2\sigma^2} tr\left(K_{NN} - Q_{nn}\right) \tag{23}$$

where Eq. 22 comes from: $\mathcal{N}(a|\mu_1, \Sigma_1)\mathcal{N}(\mu_1|\mu_2, \Sigma_2) = \mathcal{N}(a|\mu_2, \Sigma_1 + \Sigma_2)$. This gives us $\mathcal{L}_2$ from Hensman et al. [2013].

This is from Titsias, it maby not be necessary. Plugging into the lower bound from Eq. 4:

$$F_d(q) \geq \int q(X) \left[\int \phi(\mathbf{u}_d)[\log \frac{\mathcal{N}(y_d|\boldsymbol{\alpha}_d, \sigma^2)p(\mathbf{u}_d)}{\phi(\mathbf{u}_d)} d\mathbf{u}_d - \frac{1}{2\sigma^2} tr\left(K_{NN} - Q_{nn}\right)\right] dX$$

$$= \underbrace{\int \phi(\mathbf{u}_d) \left[\langle \log \mathcal{N}(y_d|\boldsymbol{\alpha}_d, \sigma^2)\rangle_{q(X)} + \log \frac{p(\mathbf{u}_d)}{\phi(\mathbf{u}_d)}\right] d\mathbf{u}_d}_{\text{now reverse Jensen's inequality: } \mathbb{E}(\cdot)\to\log\mathbb{E}\exp(\cdot)} - \frac{1}{2\sigma^2}\langle tr\left(K_{NN} - Q_{nn}\right)\rangle_{q(X)}$$

$$= \log \int \phi(\mathbf{u}_d) \exp\left(\langle \log \mathcal{N}(y_d|\boldsymbol{\alpha}_d, \sigma^2)\rangle_{q(X)}\right) \frac{p(\mathbf{u}_d)}{\phi(\mathbf{u}_d)} d\mathbf{u}_d - \frac{1}{2\sigma^2}\langle tr\left(K_{NN} - Q_{nn}\right)\rangle_{q(X)}$$

$$= \log \int \exp\left(\langle \log \mathcal{N}(y_d|\boldsymbol{\alpha}_d, \sigma^2)\rangle_{q(X)}\right) p(\mathbf{u}_d)d\mathbf{u}_d - \frac{1}{2\sigma^2}\langle tr\left(K_{NN} - Q_{nn}\right)\rangle_{q(X)}$$

We can now work out the bound from Hensman et al. [2013]. Let $\mathcal{L}_1 = \mathbb{E}_{p(\mathbf{f}_d|\mathbf{u}_d)}[\log p(\mathbf{y}_d|\mathbf{f}_d)]$:

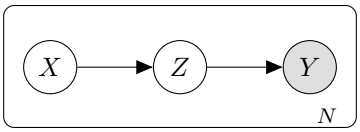
$$\exp(\mathcal{L}_1) = \prod_{i=1}^{N} \mathcal{N}(y_d^{(i)}|\boldsymbol{\alpha}_d^{(i)}, \sigma^2) \exp\left(\frac{1}{2\sigma^2}\tilde{k}_{ii}\right)$$

where $\tilde{k}_{ii}$ is the ith element of $tr\left(K_{NN} - Q_{nn}\right)$. Note that this assumes that $p(\mathbf{y}_d|\mathbf{f}_d)$ factorises over the data. Next, by plugging in the exponentiated $\mathcal{L}_1$ into the likelihood, we get:

$$\log p(y_d|X) \geq \log \int \exp(\mathcal{L}_1)p(\mathbf{u}_d)d\mathbf{u}_d = \mathcal{L}_2$$

10.3.2 Deep Gaussian Processes

Imagine each layer is a multi-output GP, for example with Q inputs and D output units, $\mathbf{f} : \mathbb{R}^Q \to \mathbb{R}^D$. Each layer, therefore, would add many model parameters. Moreover, how do we pick the size of layers and number of layers? Deep GPs as defined in Damianou and Lawrence [2013], therefore, marginalise out the entire latent space. We start with a two-layer model:



First, the marginal likelihood:

$$\log p(\mathbf{Y}) = \log \iint p(\mathbf{Y}|\mathbf{X}p(\mathbf{X}|\mathbf{Z})p(\mathbf{Z})d\mathbf{X}d\mathbf{Y}$$

where $Z \sim \mathcal{N}(0, I)$

$$\log p(\mathbf{Y}) \geq \int q \log \frac{p(\mathbf{Y}, \mathbf{F}^{\mathbf{Y}}, \mathbf{F}^{\mathbf{X}}, \mathbf{X}, \mathbf{Z})}{q} d\mathbf{X}d\mathbf{Y} \tag{24}$$

$$p(\mathbf{Y}, \mathbf{Y}^{\mathbf{Y}}, \mathbf{F}^{\mathbf{X}}, \mathbf{XZ}) = p(\mathbf{Y}|\mathbf{Y}^{\mathbf{Y}})p(\mathbf{Y}^{\mathbf{Y}}|\mathbf{X})p(\mathbf{X}|\mathbf{F}^{\mathbf{X}})$$

11 Stochastic Calculus

11.1 Brownian Motion

Brownian motion, B_t, is a stochastic process, in particular the continuous random walk that particles exhibit, for example in a gas. It satisfies the following properties:

- independent increments
- $\forall s < t : B_t - B_s \sim \mathcal{N}(0, t-s)$
- paths are continuous
- $B_0 = 0$

There are some interesting properties we can show. First, it is a discrete random walk taken to an infinite limit. Let X_n be zero-mean i.i.d. random variables with variance 1. The random walk starts at $S_0 = 0$ and proceeds with $S_n = S_0 + \sum_{i=1}^{n} X_i \quad \forall n \geq 1$. Suppose a continuous-time process:

$$B_t^n = \frac{S_{[nt]}}{\sqrt{n}} \quad \text{where } [nt] \text{ integer part of } nt$$

The Central Limit Theorem states that $Z = \frac{\hat{X}_n - \mu}{\sqrt{\sigma \cdot n}} = \mathcal{N}(0,1)$ as $n \to \infty$.

$$\implies B_t^n = \frac{S_{[nt]}}{\sqrt{nt}} \frac{\sqrt{nt}}{\sqrt{n}} = \frac{S_{[nt]}}{\sqrt{nt}} \sqrt{t} = \sqrt{t} \cdot \mathcal{N}(0,1) \quad \text{as } n \to \infty$$

Therefore, B_n^t converges to a scaled normal r.v., the variance of which will be:

$$\text{Var}\big[\sqrt{t}\mathcal{N}(0,1)\big] = t \implies B_t^n \sim \mathcal{N}(0,t)$$

Furthermore,

$$B_t^n - B_s^n = \frac{S_{[nt]} - S_{[ns]}}{\sqrt{n}} = \frac{1}{\sqrt{n}} \sum_{i=[ns]+1}^{[nt]} X_i \overset{d}{=} \frac{S_{[nt]-[ns]}}{\sqrt{n(t-s)}} \sqrt{t-s} \sim \mathcal{N}(0, t-s)$$

11.1.1 Brownian Motion as a GP

Brownian motion can be interpreted as a zero-mean GP with covariance

$$\begin{aligned} \text{Cov}(B_s, B_t) &= \text{Cov}(B_s, B_s + B_t = B_s) = \underbrace{\text{Cov}(B_s, B_s)}_{\text{Var}(B_s)} + \underbrace{\text{Cov}(B_s, B_t - B_s)}_{0 \text{ (indep. increments)}} \\ &= s \end{aligned}$$

if $s < t$. Thus, in general, the kernel is $\kappa(t,s) = \min(s,t)$.

References

Andreas Damianou and Neil Lawrence. Deep gaussian processes. In *Artificial Intelligence and Statistics*, pages 207–215, 2013.

James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*, 2013.

Michalis Titsias and Neil D Lawrence. Bayesian gaussian process latent variable model. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 844–851, 2010.

A Derivations

A.1 Double Integral of a Mean Function

$$\begin{aligned}
\int_b \int_a (ax+b)p(a)p(b) &= \int_b \int_a axp(a)p(b) + bp(a)p(b) \\
&= \int_b \int_a axp(a)p(b) + \int_b \int_a bp(a)p(b) \\
&= \int_a axp(a) \int_b p(b) + \int_a p(a) \int_b bp(b) \quad \text{but } \int_a p(a) = 1 \\
&= \int_a axp(a) + \int_b bp(b)
\end{aligned}$$

A.2 Expansion of Strange Gaussian Identity

$$\begin{aligned}
p(t_g|w_{I_g}, w_{J_g}) &\propto \exp(-\frac{1}{2}(w_{I_g} - w_{J_g} - t_g)^2) \\
&\propto \mathcal{N}(-\frac{1}{2}\begin{bmatrix} w_{I_g} - \mu_1 & w_{J_g} - \mu_2 \end{bmatrix}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\begin{bmatrix} w_{I_g} - \mu_1 \\ w_{J_g} - \mu_2 \end{bmatrix}, 1) \\
&\propto \mathcal{N}\left(-\frac{1}{2}\begin{bmatrix} (w_{I_g} - \mu_1) - (w_{J_g} - \mu_2) & -(w_{I_g} - \mu_1) + w_{J_g} - \mu_2 \end{bmatrix}\begin{bmatrix} w_{I_g} - \mu_1 \\ w_{J_g} - \mu_2 \end{bmatrix}\right) \\
&\propto \mathcal{N}\left(-\frac{1}{2}\begin{bmatrix} w_{I_g} - w_{J_g} - \mu_1 + \mu_2 & -w_{I_g} + w_{J_g} + \mu_1 - \mu_2 \end{bmatrix}\begin{bmatrix} w_{I_g} - \mu_1 \\ w_{J_g} - \mu_2 \end{bmatrix}\right) \\
&\propto \mathcal{N}\left(-\frac{1}{2}\begin{bmatrix} w_{I_g} - w_{J_g} - t_g & -w_{I_g} + w_{J_g} + t_g \end{bmatrix}\begin{bmatrix} w_{I_g} - \mu_1 \\ w_{J_g} - \mu_2 \end{bmatrix}\right) \\
&\propto \mathcal{N}\left(-\frac{1}{2}(w_{I_g} - w_{J_g} - t_g)(w_{I_g} - \mu_1) + (-w_{I_g} + w_{J_g} + t_g)(w_{J_g} - \mu_2)\right) \\
&\propto \mathcal{N}\left(-\frac{1}{2}(w_{I_g}^2 - w_{J_g}w_{I_g} - t_g w_{I_g} - \mu_1(w_{I_g} - w_{J_g} - t_g) - w_{I_g}w_{J_g} + w_{J_g}^2 + t_g w_{J_g} + \mu_2(w_{I_g} - w_{J_g} - t_g))\right) \\
&\propto \mathcal{N}\left(-\frac{1}{2}(w_{I_g}^2 - 2w_{J_g}w_{I_g} - t_g w_{I_g} + (\mu_2 - \mu_1)(w_{I_g} - w_{J_g} - t_g) + w_{J_g}^2 + t_g w_{J_g})\right) \\
&\propto \mathcal{N}\left(-\frac{1}{2}(w_{I_g}^2 - 2w_{J_g}w_{I_g} - t_g w_{I_g} + t_g(w_{J_g} - w_{I_g} + t_g) + w_{J_g}^2 + t_g w_{J_g})\right) \\
&\propto \mathcal{N}\left(-\frac{1}{2}(w_{I_g}^2 + w_{J_g}^2 + t_g^2 - 2w_{J_g}w_{I_g} - 2t_g w_{I_g} + 2t_g w_{J_g})\right)
\end{aligned}$$

B Kullback Leibler divergence

To minimise the KL divergence $\int q(x) \log \frac{q(x)}{p(x)} \, \mathrm{d}x$, we add a Lagrange multiplier to normalise the $q(z)$ to 1. We will work with $\mathcal{KL}(q(x)||p(x))$

$$\frac{\delta}{\delta q(x)}\left[\int q(x) \log \frac{q(x)}{p(x)} \, \mathrm{d}x + \lambda(1 - \int q(x) \, \mathrm{d}x)\right] = \log \frac{q(x)}{p(x)} + 1 - \lambda$$

Since $q(x) \log \frac{q(x)}{p(x)} = q(x) \log q(x) - q(x) \log p(x)$

$\implies \frac{\delta}{\delta q(x)} = q(x)/q(x) + \log q(x) - \log p(x) = 1 + \log \frac{q(x)}{p(x)}$

$q(x) = \exp(\lambda - 1)p(x)$ and we set $\lambda = 1$ for normalisation, so $q(x) = p(x)$ at the minimum.