1 Regression ([1] Chapter 2)

1.1 Weight-space view

1.1.1 Standard linear model

1.1.1.1 Standard and Bayesian model definitions

- We're trying to learn the distribution $p(y|X, W)$
  - X is the input data, W is the model parameters, y is the output
  - $p(y|X, W)$ is the conditional distribution of y after everything we know about X and W, distribution of errors
- Standard linear model: $f(X) = X^T W, y = f(X) + \epsilon$ with our Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma_n^2 I)$, which produces $p(y|X, W) = \mathcal{N}(y|f(X), \sigma_n^2)$
- Bayesian linear model: firstly, specify the linear prior distribution $p(W)$ over the weights $p(W) \sim \mathcal{N}(0, \Sigma_p)$
  - A prior $p(W)$ expresses our beliefs about the parameters before we see the data
  - Linear model specifies that our weights follow a zero mean Gaussian prior with a covariance matric Σ_p
- Then, we update our beliefs about the weights after seeing the data, using Bayes' theorem

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}} \tag{1}$$

$$p(W|X, y) = \frac{p(y|X, W)p(W)}{p(y|X)} \tag{2}$$

  - $p(y|X, W)$ is the density of the residuals after applying our priors $p(W)$ to the data X, W under our assumed noise model ϵ
  - $p(W)$ is the prior distribution of the weights
  - $p(y|X)$ is the marginal likelihood, which is the probability of the data given the model

$$p(y|X) = \int p(y|X, W)p(W)dW \tag{3}$$

  - $p(y|X)$ is the normalising constant, ensures the posterior distribution integrates to 1
  - $p(W|x, y)$ is the distribution of the weights given the data - combines the likelihood and the prior, representing everything we know about the parameters
- To understand how our posterior varies with our weights, we can write terms that only depend on weights (i.e. likelihood and prior, not marginal likelihood)

$$p(W|X, y) \propto p(y|X, W)p(W) \tag{4}$$

  - We will adopt the same idea throughout: if a term doesn't depend on weights, we simply remove it

1.1.1.2 Deriving our posterior

- Given our linear model $f(X) = X^T W$ and our Gaussian noise ϵ, we can write $p(Y|X, W)$ as the distribution of errors for each data point i

$$p(y|X, W) = \prod_{i=1}^{N} \mathcal{N}(y_i|X^T W, \sigma_n^2) \tag{5}$$

- We can find $p(y|X, W)$ by multiplying the Gaussian density $-\frac{1}{2\sigma_n^2}$ and the squared errors from our model $||y - X^T W||^2$, so our final likelihood becomes

$$p(y|X, W) = exp\left(-\frac{1}{2\sigma_n^2}||y - X^T W||^2\right) \tag{6}$$

- Given that our $p(W)$ is a zero mean Gaussian prior with covariance Σ_p, we can substitute this into the Gaussian density function

$$p(W) = \frac{1}{[\sqrt{\sigma_p}]\sqrt{2\pi}} exp\left(-\frac{1}{2}\frac{([W] - [0])}{[\Sigma_p]}\right) \tag{7}$$

- The first term of this $p(W)$ is another normalising constant, so rewriting the fraction in the exponent as a negative exponential gives us

$$p(W) \propto exp\left(-\frac{1}{2}W^T \Sigma_p^{-1} W\right) \tag{8}$$

- Putting both expressions for $p(y|X, W)$ and $p(W)$ together, we can write the posterior as

$$p(W|X, y) \propto exp\left(-\frac{1}{2\sigma_n^2}||y - X^T W||^2\right) exp\left(-\frac{1}{2}W^T \Sigma_p^{-1} W\right) \tag{9}$$

- To simplify, first we can expand $||y - X^T W||^2$ to $y^T y - 2y^T XW + W^T X^T XW$, and substitute this expanded expression to get

$$p(W|X, y) \propto exp\left(-\frac{1}{2\sigma_n^2}(y^T y - 2y^T XW + W^T X^T XW)\right) exp\left(-\frac{1}{2}W^T \Sigma_p^{-1} W\right) \tag{10}$$

- Then, we put both exponentials together (by adding their powers)

$$p(W|X, y) \propto exp\left(\frac{1}{\sigma_n^2}(y^T y - 2y^T XW + W^T X^T XW) + \left(-\frac{1}{2}W^T \Sigma_p^{-1} W\right)\right) \tag{11}$$

- We can rearrange the inside term to be a quadratic, linear and constant term in W:

$$p(W|X, y) \propto exp\left(\frac{1}{2}W^T \left(\frac{1}{\sigma_n^2}X^T X + \Sigma_p^{-1}\right) W - \left(\frac{1}{\sigma_n^2}y^T X\right) W + \frac{1}{2}y^T y\right) \tag{12}$$

- We can ignore the constant final term, and introduce $A = \Sigma_p^{-1} + \frac{1}{\sigma_n^2}X^T X$ and $b = \frac{1}{\sigma_n^2}y^T X$ to get

$$p(W|X, y) \propto exp\left(-\frac{1}{2}W^T AW + b^T W\right) \tag{13}$$

1.1.1.3 Deriving the properties of the posterior by completing the square

- Now we have a simplified form of the posterior density, we need to get it into a Gaussian form to recover the properties of the posterior distribution
- Firstly, we can bring all terms inside the exponential to a single term

$$-\frac{1}{2}W^TAW + b^TW = \frac{1}{2}\left(-W^TAW + 2b^TW\right) \tag{14}$$

- We can "complete the square" on this term $W^TAW - 2b^TW$ to rewrite it in a form that is easier to interpret

$$W^TAW - 2b^TW = (W - A^{-1}b)^TA(W - A^{-1}b) - b^TA^{-1}b \tag{15}$$

- Substituting this back into our posterior density gives us

$$p(W|X, y) \propto exp\left(-\frac{1}{2}\left((W - A^{-1}b)^TA(W - A^{-1}b) - b^TA^{-1}b\right)\right) \tag{16}$$

- If we look at our Gaussian density:

$$N(W|\mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}}exp\left(-\frac{1}{2}(W - \mu)^T\Sigma^{-1}(W - \mu)\right) \tag{17}$$

- We can see that our expression lines up with the RHS Gaussian "kernel" term $exp\left(-\frac{1}{2}(W - \mu)^T\Sigma^{-1}(W - \mu)\right)$, where $\mu = A^{-1}b$ and $\Sigma^{-1} = A$ thus $\Sigma = A^{-1}$
- So we can write our posterior density in Gaussian form

$$p(W|X, y) \sim N(A^{-1}b, A^{-1}) \tag{18}$$

1.1.1.4 Gaussian posteriors and ridge regression

- For Gaussian posteriors, our mean $A^{-1}b$ is also its mode, called the maximum a posteriori (MAP) estimate of W
  - Due to symmetries in linear model and posterior, not the case in general
- In non-Bayesian settings the MAP point is the MLE estimation
- B is dependent on σ_n^2, y and X - all known
- A is dependent on σ_n^2, X and Σ_p - all known except Σ_p
- Our weight variance Σ_p under the Bayesian linear model is "isotropic", meaning it is the same in all directions

$$\Sigma_p = \tau^2 I \tag{19}$$

  - I is our $D \times D$ correlation matrix, here we assume independence so our correlation matrix is an "identity matrix" (each diagonal element is 1 and all off-diagonal elements are 0)

  - τ^2 is a scalar variance term, chosen as a prior

- We can substitute our new isotropic prior Σ_p into A to get

$$A = \Sigma_p^{-1} + \frac{1}{\sigma_n^2} X^T X = \left[\tau^2 I\right]^{-1} + \frac{1}{\sigma_n^2} X^T X = \frac{1}{\tau^2} I + \frac{1}{\sigma_n^2} X^T X = \frac{1}{\sigma_n^2} \left(X^T X + \frac{\sigma_n^2}{\tau^2} I \right) \tag{20}$$

- Now we have full expressions for A and B, we can substitute them into our MAP estimation for W to get

$$W_{\text{MAP}} = A^{-1} b = \left[\frac{1}{\sigma_n^2} (X^T X + \frac{\sigma_n^2}{\tau^2} I \right]^{-1} \cdot \left[\frac{1}{\sigma_n^2} y^T X \right] \tag{21}$$

- We can compute the LHS inversion of A

$$A^{-1} = \frac{\sigma_n^2}{X^T X + \frac{\sigma_n^2}{\tau^2} I} = \sigma_n^2 \left(X^T X + \frac{\sigma_n^2}{\tau^2} I \right)^{-1} \tag{22}$$

- Substituting this back into W_{MAP} cancels out the σ_n^2 term in A with the $\frac{1}{\sigma_n^2}$ term in B, giving us

$$W_{\text{MAP}} = \sigma_n^2 \left(X^T X + \frac{\sigma_n^2}{\tau^2} I \right)^{-1} \cdot \frac{1}{\sigma_n^2} y^T X = \left(X^T X + \frac{\sigma_n^2}{\tau^2} I \right)^{-1} \cdot y^T X \tag{23}$$

- The solution to ridge regression is very similar

$$W_{\text{ridge}} = \left(X^T X + \lambda I \right)^{-1} X^T y \tag{24}$$

- In ridge regression, λ is a regularisation parameter that controls the amount of shrinkage which is usually selected to maximise likelihood/minimise error

- MAP estimation in Bayesian linear regression with isotropic priors is equivalent to ridge regression with a regularisation parameter $\lambda = \frac{\sigma_n^2}{\tau^2}$

  - The higher our λ, the more biased our model is towards the prior, and the more we shrink our weights towards zero and the prior has more influence
  - A lower τ causes a higher λ - smaller weight variances around zero means lower weights because of higher confidence in priors
  - A higher σ_n also causes a higher λ - larger noise variances means lower weights because of lower confidence in weights forces deference to the prior

1.1.1.5 Deriving the predictive distribution

- Ultimately, our goal is to approximate a data-generating function f_* (or a new observation y_*) that produced a new X_* given training data X and y and weights W from the same f_*

- In non-Bayesian frameworks, we make predictions by choosing a single parameter value W to maximise the likelihood of the data, which is our MLE estimate

- In a Bayesian framework, we average over all possible parameter values weighted by their posterior probability $p(W|X,y)$, e.g. for a linear model $\hat{W} = \mathbb{E}_{p(W|X,y)}[W] = W_{\text{MAP}} = A^{-1}b$
- In this framework, we can make comments about our uncertainty of W by forming a "predictive distribution" $p(f_*|X_*, X, y)$

$$p(f_*|X_*, X, y) = \int p(f_*|X_*, W) \cdot p(W|X, y)dW \tag{25}$$

  - $p(f_*|X_*, W)$ is what we think the function looks like after producing a prediction using X_* and perfect knowledge of W
  - $p(W|X, y)$ is the posterior distribution of the weights given the training data, e.g. minimised for W_{MAP}
  - $p(f_*|X_*, W) \cdot p(W|X, y)$ is the joint distribution of our predictions and our posterior weights, which gets us the conditional distribution $p(f_*, W|X_*, X, y)$ by definition of conditional probability
  - $p(f_*, W|X_*, X, y)$ relies on our perfect knowledge of W, which we don't have, so we integrate over all possible W to get the predictive distribution $p(f_*|X_*, X, y)$
- We already know $p(W|X, y)$

$$p(W|X, y) \propto exp\left(\frac{1}{2}(-W^T AW + 2b^T W)\right) \tag{26}$$

- $p(f_*|X_*, W)$ is our errors, which we assume to be distributed normally and independently with our I identity matrix:

$$p(f_*|X_*, W) = \mathcal{N}(f_*|W^T X_*, \sigma_n^2 I) \tag{27}$$

- Plugging these into our Gaussian density and ignoring the LHS normalisation term yields

$$p(f_*|X_*, w) \propto exp\left(-\frac{1}{2}\frac{1}{\sigma_n^2}(f_* - W^T X_*)^2\right) \tag{28}$$

- We can multiply $P(f_*|X_*, W)$ and $p(W|X, y)$ to get our conditional $p(f_*, W|X_*, X, y)$, and add the exponents to simplify

$$p(f_*, W|X_*, X, y) \propto exp\left(\frac{1}{2}(-W^T AW + 2b^T W) + \left(-\frac{1}{2}\frac{1}{\sigma_n^2}(f_* - W^T X_*)^2\right)\right) \tag{29}$$

- We can further combine these with a single factor of $\frac{1}{2}$ to get

$$p(f_*, W|X_*, X, y) \propto exp\left(-\frac{1}{2}\left(W^T AW - 2b^T W + \frac{1}{\sigma_n^2}(f_* - W^T X_*)^2\right)\right) \tag{30}$$

- Expanding the squared term gives us

$$p(f_*, W|X_*, X, y) \propto exp\left(-\frac{1}{2}\left(W^T AW - 2b^T W + \frac{1}{\sigma_n^2}(f_*^2 - 2f_* W^T X_* + W^T X_* X_*^T X_*)\right)\right) \tag{31}$$

- Similar to our posterior, we can rearrange this to be a quadratic, linear and constant term in W

$$p(f_*, W|X_*, X, y) \propto exp\left(-\frac{1}{2}\left(W^T\left(A + \frac{1}{\sigma_n^2}X_*X_*^T\right)W - 2\left(b + \frac{1}{\sigma_n^2}f_*X_*\right)^T W + \frac{1}{\sigma_n^2}f_*^2\right)\right) \tag{32}$$

- By defining $A_* = A + \frac{1}{\sigma_n^2}X_*X_*^T$ and $b_* = b + \frac{1}{\sigma_n^2}f_*X_*$, we can rewrite this as

$$p(f_*, W|X_*, X, y) \propto exp\left(-\frac{1}{2}\left(W^T A_* W - 2b_*^T W + \frac{1}{\sigma_n^2}f_*^2\right)\right) \tag{33}$$

- We have to integrate this wrt W to get our predictive distribution $p(f_*|X_*, X, y)$

$$p(f_*|X_*, X, y) = \int p(f_*, W|X_*, X, y)dW \propto \int exp\left(-\frac{1}{2}\left(W^T A_* W - 2b_*^T W + \frac{1}{\sigma_n^2}f_*^2\right)\right)dW \tag{34}$$

- We can factor out the $\frac{1}{\sigma_n^2}f_*^2$ term from the integral, as it does not depend on W so remains the same since $\int exp(X)dX = exp(X)$

$$= exp\left(-\frac{1}{2}\frac{1}{\sigma_n^2}f_*^2\right) \times \int exp\left(-\frac{1}{2}\left(W^T A_* W - 2b_*^T W\right)\right)dW \tag{35}$$

- The RHS term is a multivariate Gaussian integral (beyond your paygrade) which evaluates to:

$$\int exp\left(-\frac{1}{2}\left(W^T A_* W - 2b_*^T W\right)\right)dW = \frac{(2\pi)^{D/2}}{\sqrt{|A_*|}}exp\left(\frac{1}{2}b_*^T A_*^{-1} b_*\right) \tag{36}$$

- Substituting this back into our predictive distribution gets us

$$p(f_*|X_*, X, y) \propto exp\left(-\frac{1}{2}\frac{1}{\sigma_n^2}f_*^2\right) + \frac{(2\pi)^{D/2}}{\sqrt{|A_*|}} \cdot exp\left(\frac{1}{2}b_*^T A_*^{-1} b_*\right) \tag{37}$$

  - Note that no part of our expression is now dependent on W

- Now we need an expression of everything that changes f_*

- Absorb the second term, since it does not depend on f_* into the proportionality constant, and combining the remaining exponential terms by adding their powers gives us

$$p(f_*|X_*, X, y) \propto \exp\left(-\frac{1}{2}\frac{1}{\sigma_n^2}f_*^2 + \frac{1}{2}b_*^T A_*^{-1} b_*\right) \tag{38}$$

- Similar to deriving properties from our posterior, we can rearrange this expression, complete the square and derive the properties of our predictive distribution

$$p(f_*|X_*, W) \sim N(X_*^T A^{-1} b, X_*^T A^{-1} X_*) \tag{39}$$

- Note that the variance should have $+\sigma_n^2$ if dealing with y_*

- Predictive variance is quadratic form of test input with A^{-1}, showing that predictive uncertainties grow with size of X_*

1.1.2 Projections of inputs into feature space

- Bayesian linear models suffer from limited expressiveness due to the linearity of the model
- To address this, we can project our inputs into a higher dimensional feature space and apply linear model in this space
- e.g. a scalar x could be projected into the space of powers of x: $\phi(x) = [1, x, x^2, \ldots, x^d]^T$ for a polynomial basis expansion of degree d
- How to choose $\phi(x)$? Gaussian process formalism allows us to answer this question, but for now assume $\phi(x)$ is a given
- $\phi(X)$ maps a D-dimensional input vector X into an N dimensional feature space
- So our full model looks like:

$$f(X) = \phi(X)^T W \tag{40}$$

- And our predictive distribution becomes

$$p(f_*|X_*, X, y) = N(\phi(X_*)^T A_\phi^{-1} b_\phi, \phi(X_*)^T A_\phi^{-1} \phi(X_*)) \tag{41}$$

  - $A_\phi = \Sigma_p^{-1} + \frac{1}{\sigma_n^2}\phi(X)^T\phi(X)$
  - $b_\phi = \frac{1}{\sigma_n^2}\phi(X)^T y$

1.1.3 Computational issues

1.1.3.1 Avoiding inversion of A_ϕ

- This formulation of our predictive distribution inverts the $N \times N$ matrix A_ϕ, where N is dimension of feature space, to get the expected value and variance
- Inverting matrices is $O(N^3)$ - not feasible for large N - so we need to restate our predictive distribution in a form that avoids this inversion
- Substitute b_ϕ into our predictive distribution mean

$$\mathbb{E}_{p(f_*|X_*,X,y)}[f_*] = \phi(X_*)^T \cdot A_\phi^{-1} \cdot \left[\frac{1}{\sigma_n^2}\phi(X)^T y\right] \tag{42}$$

- Rearranging to isolate $A_\phi^{-1}\phi(X)$

$$= \frac{1}{\sigma_n^2}\left[A_\phi^{-1}\phi(X)\right]^T y \tag{43}$$

- We can use the Sherman-Morrison identity (beyond your paygrade) to get an expression for A_ϕ^{-1} directly, where $K = \phi(X)^T\Sigma_p\phi(X)$

$$A_\phi^{-1} = \Sigma_p - \Sigma_p\phi(X)(K + \sigma_n^2 I)^{-1}\phi(X)^T\Sigma_p \tag{44}$$

- For the mean, we can use the Sherman-Morrison identity again to get an expression for $A_\phi^{-1}\phi(X)$

$$A_\phi^{-1}\phi(X) = \sigma_n^2 \Sigma_p \phi(X)(K + \sigma_n^2 I)^{-1} \tag{45}$$

- Substitute in this expression for $A_\phi^{-1}\phi(X)$ into our predictive distribution mean

$$\mathbb{E}_{p(f_*|X_*,X,y)}[f_*] = \phi(X_*)\frac{1}{\sigma_n^2}\left[\sigma_n^2 \Sigma_p \phi(X)(K + \sigma_n^2 I)^{-1}\right]^T y \tag{46}$$

- $\frac{1}{\sigma_n^2}$ and σ_n^2 cancel out, leaving us with this final expression for the mean

$$\mathbb{E}_{p(f_*|X_*,X,y)}[f_*] = \phi(X_*)^T \cdot \Sigma_p \phi(X)(K + \sigma_n^2 I)^{-1} y \tag{47}$$

- For the variance, we can't use the Sherman-Morrison identity to arrive at an expression for $A_\phi^{-1}\phi(X_*)$ because $\phi(X_*)$ is an arbitrary N-vector, not one of the columns of $\phi(X)$

- Instead, we use the A_ϕ^{-1} expression we derived earlier to get an expression for $A_\phi^{-1}\phi(X_*)$

$$A_\phi^{-1}\phi(X_*) = \Sigma_p \cdot \phi(X_*) - \Sigma_p \phi(X)(K + \sigma_n^2 I)^{-1}\phi(X)^T \Sigma_p \cdot \phi(X_*) \tag{48}$$

- Substituting this into our predictive distribution variance gives us this final expression

$$\mathrm{Var}_{p(f_*|X_*,X,y)}[f_*] = \phi(X_*)^T \Sigma_p \phi(X_*) - \phi(X_*)^T \Sigma_p \phi(X)(K + \sigma_n^2 I)^{-1}\phi(X)^T \Sigma_p \phi(X_*) \tag{49}$$

- So our final predictive distribution is

$$\begin{aligned} p(f_*|X_*, X, y) = \mathcal{N}(\\ \phi(X_*)^T \Sigma_p \phi(X)(K + \sigma_n^2 I)^{-1} y, \\ \phi(X_*)^T \Sigma_p \phi(X_*) - \phi(X_*)^T \Sigma_p \phi(X)(K + \sigma_n^2 I)^{-1}\phi(X)^T \Sigma_p \phi(X_*) \\ ) \end{aligned}$$

- With this mean and variance, we need to invert $n \times n$ matrix $K + \sigma_n^2 I$, where n is the number of training data points, instead of $N \times N$ matrix A_ϕ, where N is the dimension of the feature space

- This formulation is faster if $n < N$
  - For polynomial basis expansions, N is degree D multiplied by number of features, so N can be very large
  - Some kernels (e.g. RBF) have infinite dimensional feature spaces, so N is infinite
  - Some data domains (e.g. text classification, genomic data) have very high dimensional feature spaces

- Geometrically, note that n datapoints can span at most n dimensions in the feature space - if $N > n$, the data forms a subspace of the feature space

1.1.3.2 Kernels and the kernel trick

- In the alternative formulation, note that ϕ is always in the same general form but different combinations of $\phi(X)$ and $\phi(X_*)$, which generalises to $\phi(X)^T\Sigma_p\phi(X')$ where X and X' are either in training $\phi(X)$ or test $\phi(X_*)$ data
  - In the first term of the mean and second term of the variance, $\phi(X_*)^T\Sigma_p\phi(X)$
  - In the first term of the variance, $\phi(X_*)^T\Sigma_p\phi(X_*)$
  - In the last term of the variance, $\phi(X)^T\Sigma_p\phi(X_*)$
  - In the definition of $K = \phi(X)^T\Sigma_p\phi(X)$
- We can define $k(X, X') = \phi(X)^T\Sigma_p\phi(X')$ as a covariance function or kernel
- Note that this kernel is an inner product with a positive definite correlation matrix Σ_p
- This lets us define $\psi(X) = \Sigma_p^{1/2}\phi(X)$
- Substituting $\psi(X)$ back into our kernel allows the $\Sigma_p^{1/2}$ to cancel out, giving us a simple dot product representation

$$k(X, X') = \psi(X) \cdot \psi(X') \tag{50}$$

- Now we have defined a kernel solely in terms of inner products in the input space, we know that there must be another equivelant $k(X, X'$ that does not require us to explicitly compute $\phi(X)$ or $\phi(X')$ in the feature space
  - e.g. if we had $\phi(X) = [1, x^1, ..., x^D]^T$ and Σ_p as an identity matrix
  - We could define $k(X, X')$ using ψ

$$k(X, X') = \psi(X)\cdot\psi(X') = \phi(X)\cdot\phi(X') = [1, x^1, ..., x^D]^T \cdot [1, x'^1, ..., x'^D]^T \tag{51}$$

  - This approach requires arranging ϕ and $\phi(X')$ into a D sized vector, then taking the dot product
  - This is trivial for small D, but as D becomes infinite (e.g. RBF kernel), arranging a D sized vector requires too much memory
  - Instead, we can define $k(X, X')$ as an equivelant function of X and X' directly

$$k(X, X') = (1 + XX')^D \tag{52}$$

  - This is the polynomial kernel, which is equivalent to the polynomial basis expansion $\phi(X)$
  - We still need to perform the same calculations but we avoid the memory cost of explicitly computing $\phi(X)$ and $\phi(X')$ in the feature space
- It is mostly more convenient to compute the kernel directly rather than the feature vectors themselves, leading to the kernel being the object of primary interest

1.2 Function-space view

1.2.1 Gaussian processes (GP)

1.2.1.1 Definition

- A GP is a collection of random variables, any finite number of which have a joint Gaussian distribution
- GPs describe a distribution over functions, where each function is a sample from the GP
- Defined completely by its mean function $m(X)$ and covariance function $k(X, X')$ of a real process $f(X)$

$$\begin{aligned} f(X) &\sim \mathcal{GP}(m(X), k(X, X')), \\ m(X) &= \mathbb{E}[f(X)], \\ k(X, X') = \text{Cov}(f(X), f(X')) &= \mathbb{E}[(f(X) - m(X))(f(X') - m(X'))] \end{aligned} \tag{53}$$

- These random variables represent the value of $f(X)$ at location X, e.g. often Gaussian processes are defined over time so X can be a time point
- The covariance function specifies the covariance between pairs of random variables

1.2.1.2 Consistency requirement

- This definition implies a consistency requirement that means that examining a larger set does not change the distribution of a smaller set
  - e.g. if GP implies that $(f(X_1), f(X_2)) \sim \mathcal{N}(\mu, \Sigma)$, then it must specify $(f(X_1) \sim \mathcal{N}(\mu_1, \Sigma), f(X_2) \sim \mathcal{N}(\mu_2, \Sigma))$ where $\mu_\theta = m(X_\theta)$ and $\Sigma_{\theta\theta} = k(X_\theta, X_\theta)$
- This requirement is also called the marginalisation property because to get the smaller distribution of $f(X_1)$, we marginalise out the larger distribution of $f(X_1), f(X_2)$ by integrating the larger distribution wrt $f(X_2)$
  - Similar to how we integrated over W to get the predictive distribution in Bayesian linear regression
- Consistency is automatically gained if our covariance function specifies entries in the covariance matrix
  - Note that we wouldn't have consistency if we specified the entries of the "precision matrix" (inverse covariance matrix), as we would need to use all entries of the covariance matrix instead of just the θ we are looking for

1.2.1.3 Bayesian linear regression as a GP

- We can view Bayesian linear regression model $f(X) = \phi(X)^W$ with prior $W \sim N(0, \Sigma_P)$ as a GP

$$\begin{aligned} m(X) &= \phi(X)^T \mathbb{E}[W] = \phi(X)^T[0] = 0 \\ k(X, X') &= \phi(X)^T \mathbb{E}[WW^T]\phi(X') = \phi(X)^T \Sigma_P \phi(X') \end{aligned} \tag{54}$$

- We will use a squared exponential (SE) covariance function, also known as the radial basis function (RBF) or Gaussian kernel

$$k(f(X), f(X')) = \exp\left(-\frac{1}{2}\frac{|X - X'|^2}{\mathcal{L}^2}\right) \tag{55}$$

  - Covariance between outputs $f(X)$ and $f(X')$ is written as a function of inputs X and X' only (kernel trick)
  - For SE, covariance is almost unity between outputs whose inputs are close together, and decays exponentially as inputs get further apart
  - $\mathcal{L}$ is a length scale parameter that controls the rate of decay of the covariance function

- It can be shown that SE corresponds to a Bayesian linear regression model with infinite basis functions
  - For every positive definite covariance function $k(X, X')$, there exists a possibly infinite set of basis functions - Mercer theorem
  - SE can also be obtained from the linear combination of infinite Gaussian-shaped basis functions

- Because SE is infinitely differentiable, it produces smooth functions

1.2.1.4 Function evaluations to a random function

- We can choose a subset of five inputs X_{*1} from our test data X_* and apply a GP to get five outputs $f(X_{*1})$

- $f(X_{*1})$ can be described as a multivariate Gaussian distribution,e.g. in the Bayesian linear model $f(X_{*1}) \sim N(0, k(X_{*1}, X_{*1}))$
  - Each output $f(X_{\theta*1})$ in our $f(X_{*1})$ vector is a random variable with mean 0 and covariance with each other $K_{\theta\theta'} = k(X_{\theta*}, X_{\theta'*})$

- There exists some random function $g(X_{*1})$ for our subsets such that $f(X_{*1}) = g(X_{*1})$
  - We only know the value of $g(X_{*1})$ at the points X_{*1}, so $g(X_{*1}) = X_{*1} : f(X_{*1})$
  - Because $g(X)$ entirely consists of random points, we can think of $g(X_{*1})$ as a random function
  - $g(X)$ is continuous because SE guarantees consistency

- Thanks to consistency, if we marginalised out our subset from the entire distribution $f(X_*)$, we would recover the subset distribution $N(0, K_*(X_{*1}, X_{*1}))$ that describes our random function $g(X_{*1})$

- Therefore, the specification of the covariance function imples that our GP can also be seen as a distribution of o, where each sample produces a random function $g(X_*)$ that passes through the points $f(X_{*1})$

1.2.2 Predictive distributions with noise-free observations

1.2.2.1 Prior distribution over functions

- We are trying to get the best function f that describes our training data $f(X)$ and test data $f(X_*)$
- Initially we consider the simple special case where the observations are noise free, i.e. we know $f(X)$ is the y we observe
- Therefore, our training data $f(X)$ and test data $f(X_*)$ are jointly distributed according to the prior

$$\begin{pmatrix} f(X) \\ f(X_*) \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} K(X,X) & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{pmatrix}\right) \tag{56}$$

1.2.2.2 Posterior distribution over functions

- We are not interested in drawing random functions from the prior, but incorporating the knowledge that the training data provides about the function
- To get the posterior distribution over functions given the training data, we can condition the joint prior distribution on the training data
- Intuitively, like generating random functions g and rejecting those that don't pass through the training data (but computationally impossible like this)
- Probabilistically, we need to condition our joint Gaussian prior distribution on the observations, or produce an expression for $p(f(X_*)|X_*, X, f(X))$
- We can substitute our prior and our data we're conditioning it on into the Gaussian multivariate conditioning identity:

$$\begin{aligned} p(f(X_*)|X_*, X, f(X)) \sim N(\\ [0] + [K(X_*,X)][K(X,X)]^{-1}([f(X)] - [0]), \\ [K(X*,X*)] - [K(X_*,X)][K(X,X)]^{-1}[K(X,X_*)] \\ ) \end{aligned} \tag{57}$$

  - Note that although we condition on X_*, X, and $f(X)$, we only substitute $f(X)$
  - X_* and X are known constants, but $f(X)$ is random because it is a sample from the prior
  - We also swap $f(X_*)$ and $f(X)$ in our prior to match the conditioning identity, such that our input vector into the conditioning identity is $(f(X_*), f(X))^T$

- Simplifying the last term in the mean we get our final expression for the posterior distribution

$$\begin{aligned} p(f(X_*)|X_*, X, f(X)) \sim N(\\ K(X,X_*)K(X,X)^{-1}f(X), \\ K(X_*,X_*) - K(X_*,X)K(X,X)^{-1}K(X,X_*) \\ ) \end{aligned} \tag{58}$$

1.2.3 Predictive distributions with noisy observations

1.2.3.1 Noisy observations prior

- It is typical for more realistic situations to not have the observation $f(X)$ directly as our training data, but have $y = f(X) + \epsilon$, where ϵ is a noise term
- Linear model assumes ϵ is additive, independent and identically distributed

$$\text{Cov}(y_p, y_q) = K(X_p, X_q) + \sigma_n^2 \delta_{pq} \tag{59}$$

  - δ_{pq} is the Kronecker delta, which returns 1 if indices (p, q) are equal and 0 otherwise
  - σ_n^2 is the noise variance, which is a constant for all observations
  - Note that the subscript n in σ_n^2 is not the number of observations, but a reminder it's noise and not part of the prior

- More generally,

$$\text{Cov}(Y) = k(X, X) + \sigma_n^2 I \tag{60}$$

  - $\sigma_n^2 I$ is the noise multiplied by the correlation matrix I, which under independence is our identity matrix

- This gives us this prior

$$\begin{pmatrix} Y \\ f(X_*) \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} K(X, X) + \sigma_n^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{pmatrix} \right) \tag{61}$$

1.2.3.2 Noisy observations posterior

- As before, we can form a predictive distribution using the Gaussian multivariate conditioning identity

$$\begin{aligned} p(f(X_*)|X_*, X, Y) \sim N(\\ K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1} Y, \\ K(X_*, X_*) - K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1} K(X, X_*) \\ ) \end{aligned} \tag{62}$$

- Note that substituting $K(X, X') = \phi(X)^T \Sigma_p \phi(X')$ in gives us the exact same result in the weight-space view's final predictive distribution in the alternative formulation using the Sherman-Morrison identity
- Note that our mean is solely in terms of Y and our covariance function, so it can be seen as as a linear predictor
- Alternatively, we can write it as a linear combination of n kernel function evaluations at the training data points X_i by defining $\alpha_i = K(X_i, X_i) + \sigma_n^2 I)^{-1} Y_i$

$$\mathbb{E}_{p(f(X_*)|X_*, X, Y)}[f(X_*)] = \sum_{i=1}^{n} \alpha_i k(X_*, X_i) \tag{63}$$

- Representer theorem - we can express the posterior mean as these linear combinations despite our GP being represented as a (possibly infinite) number of basis functions
  - Our prior defines a joint Gaussian distribution over all y variables, one for each point in the index set $\mathcal{X}$, which could be infinite
  - Since we are conditioning our prior on a finite number of X_i, we only care about the distribution of the n training points and the 1 test point we are predicting for
  - This is the block of the joint covariance matrix we extract by marginalising
- Note that our variance is independent of the targets y, only our inputs X and X_*
- Our variance is two terms: our prior covariance $K(X_*, X_*)$, a term representing the information the observations give us about the function
- We can compute the predictive distribution of Y_* by adding the noise term $\sigma_n^2 I$ to the variance

1.2.3.3 Marginal likelihood

- We need some measure of how well our GP fits the data, which we can get by computing the marginal likelihood $p(Y|X)$

$$p(Y|X) = \int p(Y|f, X)p(f|X)df \tag{64}$$

- $p(f|X)$ is our prior distribution $\sim N(0, K)$
  - Our mean will always be the same under our prior, but our $K(X, X')$ tells us how "wiggly" our function is
  - The closer our $p(f|X)$ distribution is to the true complexity of the function, the higher our marginal likelihood
  - e.g. for SE, if our data is close together $|X - X'|$ is small then our covariance $k(X, X')$ on our function distribution prior is large, so we get a high variety of functions and thus a higher probability of sampling a more complex function
  - If we specify a lower $\mathcal{L}$, we can "artificially" get a high $k(X, X')$
  - So if we supply some new data $k(X_*, X'_*)$ that is far away from each other and would require a more complex function, its density in this prior smooth distribution will be low, so we get a low $p(f|X_*)$
- $p(y|f, X)$ is our familar predictive distribution $p(y|f, X) \sim N(f, \sigma_n^2 I)$, and represents how good a fit our function f is to the training data labels y and inputs X
- We can express $p(Y|X)$ as a Gaussian integral over the joint distribution of f and Y, which yields this PDF of the marginal likelihood

$$logp(Y|X) = -\frac{1}{2}Y^T(K + \sigma_n^2 I)^{-1}Y - \frac{1}{2}log|K + \sigma_n^2 I| - \frac{n}{2}log(2\pi) \tag{65}$$

- Alternatively, $y = f(X) + \epsilon$, where we assume $\epsilon \sim N(0, \sigma_n^2 I)$
- Since these are both Gaussian, we can simply add their means and variances to get $p(Y|X) = N(0, K + \sigma_n^2 I)$
- This itself is a Gaussian that we can plug into our Gaussian PDF to get the same result

1.2.3.4 Practical algorithm

1. Take in inputs X, outputs y, covariance function k, noise level σ_n^2, and test input X_*
2. Invert our $[K(X, X) + \sigma_n^2 I]$ matrix needed for mean variance using Cholesky decomposition
   (a) $L = \text{cholesky}(K(X, X) + \sigma_n^2 I)$
3. Prepare the mean of our predictive distribution in linear combination form by computing the α vector
   (a) $alpha = L^T \backslash (L \backslash y)$
4. Actually compute the mean
   (a) $mu = K(X_*, X)^T \cdot alpha$
5. Prepare to compute variance by computing v, the form in which L is used in the variance
   (a) $v = L \backslash K(X_*, X)^T$
6. Compute the variance
   (a) $var = K(X_*, X_*) - v^T v$
7. Compute the log marginal likelihood
   (a) $\log p(Y|X) = -\frac{1}{2} y^T \cdot alpha - \frac{1}{2} log|K(X, X) + \sigma_n^2 I| - \frac{n}{2} log(2\pi)$
8. Return the mean mu, variance var, and log marginal likelihood

References

[1] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, Nov. 2005. ISBN: 9780262256834. DOI: 10.7551/mitpress/3206.001.0001. eprint: https://direct.mit.edu/book-pdf/2514321/book_9780262256834.pdf. URL: https://doi.org/10.7551/mitpress/3206.001.0001.