1 Linear Methods for Regression

1.1 Multivariate Normal Distributions

Multivariate Normal distributions (MVNs) are generalizations of the usual univariate Normal.

$$N_{\mathbf{x}}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) := \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right]$$

with covariance matrix $\boldsymbol{\Sigma}$ and mean $\boldsymbol{\mu}$.

The covariance matrix, being by construction real symmetric can be diagonalized. This is a very useful property that means we can rewrite any $MVN(x)$ in terms of a the product of independent Normal distributions in the elements of some new variable $\mathbf{y}$ that relates to $\mathbf{x}$ through a coordinate transformation $\mathbf{y} = \boldsymbol{U}^T(\mathbf{x}-\boldsymbol{\mu})$ where $\boldsymbol{U}$ is constructed through eigendecomposition of $\boldsymbol{\Sigma}$ s.t. $\boldsymbol{\Sigma}^{-1} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T$.

Note that by doing this we can sample from a MVN by actually sampling from a series of univariate Normals and running the inverse coordinate transform.

Mahalanobis distance Mahalanobis distance is defined as

$$d(\mathbf{x})^2 := (\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}),$$

i.e. the distance from a point $\mathbf{x}$ to the centre of $N_{\mathbf{x}}$. If we consider the diagonalized coordinates $\mathbf{y}$ as well, then we can see that it is equivalently the distance between $\mathbf{x}$ and $\boldsymbol{\mu}$ rotated by $\boldsymbol{U}$. It has applications in outlier identification as well as clustering and classification tasks, by identifying the probability of a test point belonging to a set defined by $N_{\mathbf{x}}$.

Moments

$$E[\mathbf{x}] = \boldsymbol{\mu}, \quad E[\mathbf{x}\mathbf{x}^T] = \boldsymbol{\mu}\boldsymbol{\mu}^T + \boldsymbol{\Sigma}$$

See Appendix A.1.1 for proof.

1.1.1 Conditional MVN

An important property of MVNs is that if the joint distribution of a pair of variables is Gaussian, then the conditional distribution of one on the other is also Gaussian, in addition to both marginals. This property can be demonstrated by partitioning a MVN, i.e. defining the following quantities:

$$\mathbf{x} = \begin{pmatrix}\mathbf{x}_a \\ \mathbf{x}_b\end{pmatrix}, \quad \boldsymbol{\mu} = \begin{pmatrix}\boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b\end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix}\boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb}\end{pmatrix}, \text{and} \quad \boldsymbol{\Sigma}^{-1} = \boldsymbol{\Theta} = \begin{pmatrix}\boldsymbol{\Theta}_{aa} & \boldsymbol{\Theta}_{ab} \\ \boldsymbol{\Theta}_{ba} & \boldsymbol{\Theta}_{bb}\end{pmatrix}$$

writing out the full joint distribution and looking for terms quadratic and linear in each variable. What we find is that

$$\begin{aligned} \mathrm{Cov}_{\mathbf{x}_a|\mathbf{x}_b}[\mathbf{x}_a] &= \boldsymbol{\Theta}_{aa}^{-1} \\ E_{\mathbf{x}_a|\mathbf{x}_b}[\mathbf{x}_a] &= \boldsymbol{\mu}_a - \boldsymbol{\Theta}_{aa}^{-1}\boldsymbol{\Theta}_{ab}(\mathbf{x}_b - \boldsymbol{\mu}_b) \\ \implies p(\mathbf{x}_a \mid \mathbf{x}_b) &= N_{\mathbf{x}_a}(\boldsymbol{\mu}_a - \boldsymbol{\Theta}_{aa}^{-1}\boldsymbol{\Theta}_{ab}(\mathbf{x}_b - \boldsymbol{\mu}_b), \boldsymbol{\Theta}_{aa}^{-1}) \end{aligned}$$

If we calculate the marginal distributions as well, by integrating out the other variable from the joint distribution, we find that $p(\mathbf{x}_a) = N_{\mathbf{x}_a}(\boldsymbol{\mu}_a, \boldsymbol{\Sigma}_{aa})$.

See [1] section 2.3 for more details of these derivations.

1.1.2 Gaussian Linear Model

In a Gaussian Linear Model (i.e. a linear model with Gaussian prior and likelihood) we can derive Gaussian marginals and posterior, as given below.

Prior $p(\mathbf{x}) = N_{\mathbf{x}}(\boldsymbol{\mu}, \boldsymbol{\Theta}^{-1})$

Likelihood $p(\mathbf{y} \mid \mathbf{x}) = N_{\mathbf{y}}(\boldsymbol{A}\mathbf{x} + \boldsymbol{b}, \boldsymbol{L}^{-1})$

Marginal $p(\mathbf{y}) = N_{\mathbf{y}}(\boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{b}, \boldsymbol{L}^{-1} + \boldsymbol{A}\boldsymbol{\Theta}^{-1}\boldsymbol{A}^T)$

Posterior $p(\mathbf{x} \mid \mathbf{y}) = N_{\mathbf{x}}(\boldsymbol{\Sigma}[\boldsymbol{A}^T\boldsymbol{L}(\mathbf{y} - \boldsymbol{b}) + \boldsymbol{\Theta}\boldsymbol{\mu}], \boldsymbol{\Sigma})$

where $\boldsymbol{\Sigma} = (\boldsymbol{\Theta} + \boldsymbol{A^T L A})^{-1}$). Details are again given in [1] section 2.3.

Maximum Likelihood Estimator We can calculate the Maximum Likelihood Estimates of the parameters of our MVN in much the same way as in the univariate case, i.e. by maximising the log-likelihood w.r.t the parameters of interest, i.e. $\text{argmax}_{\boldsymbol{\mu},\boldsymbol{\Sigma}} L(\boldsymbol{\mu}, \boldsymbol{\Sigma}, D)$ for a given dataset D with $L = \log Z - \frac{n}{2}\log\det\boldsymbol{\Sigma} - \frac{1}{2}\text{Tr}(\overline{\mathbf{X}}^T\overline{\mathbf{X}}\boldsymbol{\Sigma}^{-1})$ and where $\overline{\mathbf{X}}$ is the centred dataset $\overline{\mathbf{X}}^T = [(\mathbf{x}_1 - \boldsymbol{\mu}) \dots (\mathbf{x}_n - \boldsymbol{\mu})] \in R^{d\times n}$. The solutions we obtain via the maximization (Appendix A.1.2) are that

$$\boldsymbol{\mu}_{MLE} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$$

$$\boldsymbol{\Sigma}_{MLE} = \frac{1}{n}\overline{\mathbf{X}}_{MLE}^T\overline{\mathbf{X}}_{MLE} \quad \text{with} \quad \overline{\mathbf{X}}_{MLE}^T = [(\mathbf{x}_1 - \boldsymbol{\mu}_{MLE}) \dots (\mathbf{x}_n - \boldsymbol{\mu}_{MLE})]$$

1.2 Bias-Variance Decomposition

When we build a model we need a fair way to assess it, given what we know about the tradeoffs between model flexibility and overfitting. *Bias-Variance Decomposition* is a method for analysing the training error that quantifies this tradeoff. Consider a model $f(\mathbf{x})$ and loss function $L(f, D)$. To generalize our assessment of the model beyond the specific dataset D, we want to measure the *expected loss* given by $\text{E}_D[L(f, D)] = \text{E}_D\left[\sum_{i\in D}[y_i - f(\mathbf{x}_i; \mathbf{w}_{LS})]^2\right] = \sum_{i=1\dots n}\text{E}_D[[y_i - f(\mathbf{x}_i; \mathbf{w}_{LS})]^2 \mid \mathbf{x}_i]$. The first step in our analysis is to assume a generating process with an additive noise assumption; i.e. of the form $y_i = g(\mathbf{x}_i) + \epsilon_i$ where $g(\mathbf{x})$ is a deterministic function, $\forall i$ ϵ_i is independent of $\mathbf{x}_i$ and $\text{E}_\epsilon[\epsilon_i] = 0$. From this, we can derive the following decomposition of the expected loss (see A.2):

$$\text{E}_D[(y - f_{LS}(\mathbf{x}_i))^2 \mid \mathbf{x}_i] = \underbrace{\text{Var}_\epsilon[\epsilon_i]}_{\text{irreducible error}} + \underbrace{(g(\mathbf{x}_i) - \text{E}_\epsilon[f_{LS}(\mathbf{x}_i) \mid \mathbf{x}_i])^2}_{\text{bias}} + \underbrace{\text{Var}_\epsilon[f_{LS}(\mathbf{x}_i) \mid \mathbf{x}_i]}_{\text{variance}}$$

This breakdown enables us to interpret the expected loss in terms of the randomness in the data generating process, the accuracy of our model, and the sensitivity of our prediction function to the randomness in the data (as described by the corresponding three terms in the equations above). We can now also see the *bias-variance tradeoff* in action. As the complexity of f increases, it is more able to represent the data and decrease the bias, but simultaneously, it is more sensitive to noise and hence the variance term increases; thus, we see a balance between the two terms.

In practice, the expected loss shown above is not very useful for determining model performance since we know neither the underlying generative process $g(\mathbf{x})$ nor the distribution of ϵ. As a result, we choose to calculate an approximation of the *out-of-sample* performance (as opposed to the *in-sample*) where we calculate the error we measure when making predictions on an unseen dataset D_1 using the prediction function f_0 fit on the training set D_0. This approximated error is given by: $\frac{1}{n'}\sum_{(y',\mathbf{x}')\in D_1}(y' - f_0(\mathbf{x}'))^2$. This choice is justified in Appendix A.2.3.

1.3 Feature Transforms and Kernel Methods

Feature Transforms are rooted in the idea that we can approximate a function by a *linear basis expansion* in some basis $\boldsymbol{\phi}(\mathbf{x})$: $g(\mathbf{x}) \simeq f(\mathbf{x};\mathbf{w}) = \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x})\rangle = \sum_i w_i \phi_i(\mathbf{x})$. Previously, we used a *polynomial basis* ($\phi(x) = [x, x^2, \ldots, x^b]$), upto order b, which from a Taylor expansion, we can see would be a reasonable representation for a smooth function. For a periodic function, a Fourier series representation might be more appropriate, in which case we would use a *trigonometric basis* ($\phi(x) = [\sin(x), \cos(x), \sin(2x), \cos(2x), \ldots, \sin(bx), \cos(bx)]$). There are many other potential choices of basis, but another common choice is the Radial Basis Function.

Radial Basis Function (RBF) ($\phi_i(\mathbf{x}) := \exp(-\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{2\sigma^2})$). Note that for the RBF, we must specify the bandwidth σ (usually by the median of the pairwise distances of $\mathbf{x}$) and instead of 'orders' of terms, we apply the RBF to b *RBF centroids*, which are just randomly chosen points from D. When picking b, we need to make sure that we cover the space effectively such that $f(\mathbf{x};\mathbf{w})$ is supported [1] whereever $g(\mathbf{x})$ is supported. In high dimensional spaces, the RBF looks like a ball, and so the required number b is the *packing number* and grows exponentially in d, the number of dimensions; so we find that the *Curse of Dimensionality* is once again a problem.

1.3.1 Kernel trick

So far we have considered b dimensional transforms that map to a b-dim. feature space R^b. Suppose we want to use a feature transform that maps to an infinite dimensional space instead. Using our usual least-squares solution we would need to calculate an infinite number of parameters, $\mathbf{w}$. As it turns out though, we can rewrite the solution as $\mathbf{w}_{LS-R} = \mathbf{\Phi}^T(\mathbf{\Phi}\mathbf{\Phi}^T + \lambda\mathbf{I})^{-1}\mathbf{y}$ (Appendix A.3.1). We have replaced our infinite dimensional square matrix $\mathbf{\Phi}^T\mathbf{\Phi}$ with an $n \times n$ matrix instead. Now we define an object $\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}^T$ with $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = \langle\boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j)\rangle$ and $k_i = k(\mathbf{x}, \mathbf{x}_i) = \langle\boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{x}_i)\rangle$. Using this new $\mathbf{K}$ we can rewrite our prediction function as $f(\mathbf{x};\mathbf{w}_{LS-R}) = \mathbf{k}^T(\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{y}$. The value in this process is that now the feature transform function has disappeared from our prediction formula, meaning we don't need to explicitly calculate it (useful when we are using infinite dimensional maps) and now we need only specify an appropriate *kernel function* $k(\mathbf{x}, \mathbf{x}')$ that satisfies the relevant properties (i.e. can be represented by an inner-product on some vector space - see A.3.2 for an example using the polynomial

[1] $\text{supp}(f) := \{\mathbf{x} \mid f(\mathbf{x}'\mathbf{w}) \neq 0\}$

kernel). There are many possible choices of k which are largely task-dependent. Some common choices are the following:

Linear $k(\mathbf{x},\mathbf{x}') = \langle \mathbf{x},\mathbf{x}' \rangle$

Polynomial $k(\mathbf{x},\mathbf{x}') = (\langle \mathbf{x},\mathbf{x}' \rangle + 1)^b$

RBF $k(\mathbf{x},\mathbf{x}') = \exp(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2})$

Often for $\mathbf{x} \in R^d$ the RBF is considered to be a good choice.

1.4 Model Selection

1.4.1 Frequentist Model Selection

Under the frequentist framework, we minimise some expected loss function and assess model performance with testing error (as we saw in 1.2). This approach tends to work reasonably well, but suffers from a few weaknesses. First, we need to retain some data as a hold-out for testing, which means we will lose some information. Secondly, we tend to need to run multiple train-test splits (under CV) to obtain robust performance estimates which is computationally expensive and relies on the assumption of IID data.

1.4.2 Probabilistic Model Selection

Under a probabilistic framework we instead introduce priors over models, i.e. $p(m)$ for $m \in \mathcal{M} = \{m_1, \ldots, m_k\}$ and as usual we can write down our model posterior $p(m \mid D) \propto p(D \mid m)p(m)$. This gives us a weighting for the relative importance of our model m from the set $\mathcal{M}$ but rather than pick the largest value (as we would following frequentist logic), we instead *marginalize* over m to get a *model average*: $p(\hat{y} \mid D) = \sum_{m \in \mathcal{M}} p(\hat{y} \mid D, m)p(m \mid D)$.

Model Evidence To compute our model average, we need to calculate the *model evidence* or *marginal likelihood* for each m i.e. $p(D \mid m) = \int p(D \mid \mathbf{w}, m)p(\mathbf{w} \mid m)d\mathbf{w}$. [2] We can approximate the (log) evidence with the following (A.4.1) $\log p(D \mid m) \simeq \log p(D \mid \mathbf{w}_{MAP}, m) + \log \frac{\Delta posterior}{\Delta prior}$. This shows us that there is a trade-off between the two terms, the latter of which is always negative and *more* negative the sharper the posterior distribution is (which corresponds to having a more flexible model). Ultimately we find that models with the highest evidence have intermediate complexity.

Hyper-parameters Usually we parameterise our models m by some set of *hyper-parameters* α and marginalize over them to get our predictive distribution. This procedure is often intractable, so we hope that we can approximate $p(\alpha \mid D)$ by $\delta(\alpha - \hat{\alpha})$ and avoid doing a complex integral over α. To do this, we need to compute $\hat{\alpha}$ first, which we do by *Marginal Likelihood Maximization*: $\hat{\alpha} := \text{argmax}_\alpha \int p(D|\mathbf{w}, \alpha)p(\mathbf{w}|\alpha)d\mathbf{w}$. A.4.2.

[2] Note that the model evidence is the normalizing factor for the parameter posterior $p(\mathbf{w}|D, m) = \frac{p(D \mid \mathbf{w}, m)p(\mathbf{w} \mid m)}{p(D \mid m)}$.

Appendices

A Appendix

A.1 MVN

A.1.1 Moments

$$\begin{aligned}
E[\mathbf{x}] &= \frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}|^{1/2}} \int \mathbf{x} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right] d\mathbf{x} \\
&= \frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}|^{1/2}} \int (\mathbf{z}+\boldsymbol{\mu}) \exp\left[-\frac{1}{2}\mathbf{z}^T\mathbf{\Sigma}^{-1}\mathbf{z}\right] d\mathbf{z} \\
&= \boldsymbol{\mu}
\end{aligned}$$

where we have used the substitution $\mathbf{z} = \mathbf{x} - \boldsymbol{\mu}$ and that the integral of an odd function over symmetric limits $(-\infty, \infty)$ vanishes ($\mathbf{z}$ is odd, MVN is even) and that the second term of the integral is just the product of $\boldsymbol{\mu}$ and the usual definition of the (normalised) MVN in $\mathbf{z}$ and hence a constant ($\boldsymbol{\mu}$) multiplied by 1.

$$\begin{aligned}
E[\mathbf{x}\mathbf{x}^T] &= \frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}|^{1/2}} \int \mathbf{x}\mathbf{x}^T \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right] d\mathbf{x} \\
&= \frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}|^{1/2}} \int (\mathbf{z}+\boldsymbol{\mu})(\mathbf{z}+\boldsymbol{\mu})^T \exp\left[-\frac{1}{2}\mathbf{z}^T\mathbf{\Sigma}^{-1}\mathbf{z}\right] d\mathbf{z} \\
&= \frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}|^{1/2}} \int (\mathbf{z}\mathbf{z}^T + \boldsymbol{\mu}\boldsymbol{\mu}^T + \mathbf{z}\boldsymbol{\mu}^T + \boldsymbol{\mu}\mathbf{z}^T) \exp\left[-\frac{1}{2}\mathbf{z}^T\mathbf{\Sigma}^{-1}\mathbf{z}\right] d\mathbf{z}
\end{aligned}$$

As before, we have that the terms linear in $\mathbf{z}$ vanish by symmetry and that the term quadratic in $\boldsymbol{\mu}$ integrates to $1 \cdot \boldsymbol{\mu}\boldsymbol{\mu}^T$ leaving us with just the quadratic $\mathbf{z}$ term to do.

Now substitute the change of variables that diagonalizes $\mathbf{\Sigma}^{-1}$:

$$\begin{aligned}
\mathbf{\Sigma}^{-1} &= \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T \\
\mathbf{y} = \boldsymbol{U}^T\mathbf{z} &\implies \mathbf{z} = \boldsymbol{U}\boldsymbol{y} \\
d\mathbf{z} &= \det \boldsymbol{U} d\mathbf{y} = d\mathbf{y} \\
\implies &\frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}|^{1/2}} \int \boldsymbol{U}\boldsymbol{y}\boldsymbol{y}^T\boldsymbol{U}^T \exp\left[-\frac{1}{2}\mathbf{y}^T\boldsymbol{U}^T\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T\boldsymbol{U}\boldsymbol{y}\right] d\mathbf{y} \\
&= \frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}|^{1/2}} \prod_m \sum_{jk} \int U_{ij} y_j y_k U^T_{kl} \mathrm{e}^{-\frac{1}{2}\sum_k \frac{y_k^2}{d_k}} dy_m \\
&= \frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}|^{1/2}} \sum_j \int U_{ij} y_j^2 U^T_{jl} \mathrm{e}^{-\frac{1}{2}\frac{y_j^2}{d_j}} dy_j \\
&= \frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}|^{1/2}} \sum_j U_{ij} U^T_{jl} d_j^{3/2} \cdot (2\pi)^{d/2} \\
&= \sum_j U_{ij} d_j U^T_{jl} \\
\mathbf{\Sigma} &= \boldsymbol{U}\boldsymbol{D}^{-1}\boldsymbol{U}^T
\end{aligned}$$

where we have used the same symmetry arguments as before to set $j = k$ in the integral above, the fact that $\det \mathbf{\Sigma} = \prod_k d_k$ and the univariate result (proved below).

Putting all of this together we get the final result that

$$E[\mathbf{x}\mathbf{x}^T] = \boldsymbol{\mu}\boldsymbol{\mu}^T + \mathbf{\Sigma}$$

Aside: prove that in the univariate case with $\mu = 0$, $E[x^2] = \sigma^2$:

$$\begin{aligned}
\frac{1}{(2\pi\sigma^2)^{1/2}} \int x^2 \mathrm{e}^{\frac{-x^2}{2\sigma^2}} dx &= \frac{1}{(2\pi\sigma^2)^{1/2}} \left(-\partial_{\frac{1}{2\sigma^2}} \int \mathrm{e}^{\frac{-x^2}{2\sigma^2}} dx\right) \\
&= \frac{1}{(2\pi\sigma^2)^{1/2}} \left(-\partial_{\frac{1}{2\sigma^2}} (2\pi\sigma^2)^{1/2}\right) \\
-\partial_{\frac{1}{2\sigma^2}} &= 2\sigma^4 \partial_{\sigma^2} \\
\implies &= \frac{1}{(2\pi\sigma^2)^{1/2}} 2\sigma^4 \partial_{\sigma^2} (2\pi\sigma^2)^{1/2} \\
&= \sigma^2
\end{aligned}$$

A.1.2 MLE

Prove

$$\begin{aligned}
\boldsymbol{\mu}_{MLE} &= \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i \\
\mathbf{\Sigma}_{MLE} &= \frac{1}{n}\overline{\mathbf{X}}^T_{MLE}\overline{\mathbf{X}}_{MLE} \quad \text{with} \quad \overline{\mathbf{X}}^T_{MLE} = [(\mathbf{x}_1 - \boldsymbol{\mu}_{MLE}) \ldots (\mathbf{x}_n - \boldsymbol{\mu}_{MLE})]
\end{aligned}$$

given that $L = \log Z - \frac{n}{2}\log\det\mathbf{\Sigma} - \frac{1}{2}\operatorname{Tr}(\overline{\mathbf{X}}^T\overline{\mathbf{X}}\mathbf{\Sigma}^{-1})$.

To prove the first result, we want to differentiate L w.r.t $\boldsymbol{\mu}$, which will require us to use two results from [2] (equation 111 and 137 [the chain rule]):

$$\partial_{\mathbf{X}} g(\boldsymbol{U}) = \mathrm{Tr}\left[(\partial_{\boldsymbol{U}} g(\boldsymbol{U}))^T \partial_{\mathbf{X}} \boldsymbol{U}\right] \tag{1}$$

$$\partial_{\mathbf{X}} \mathrm{Tr}(\mathbf{X}\boldsymbol{B}\mathbf{X}^T) = \mathbf{X}\boldsymbol{B}^T + \mathbf{X}\boldsymbol{B} \tag{2}$$

Hence we get

$$\begin{aligned}
\partial_{\boldsymbol{\mu}} L &= -\frac{1}{2}\partial_{\boldsymbol{\mu}} \mathrm{Tr}(\overline{\mathbf{X}}^T \overline{\mathbf{X}} \boldsymbol{\Sigma}^{-1}) = -\frac{1}{2}\partial_{\boldsymbol{\mu}} \mathrm{Tr}(\overline{\mathbf{X}} \boldsymbol{\Sigma}^{-1} \overline{\mathbf{X}}^T) \\
0 &= -\mathrm{Tr}[\partial_{\overline{\mathbf{X}}} \mathrm{Tr}(\overline{\mathbf{X}} \boldsymbol{\Sigma}^{-1} \overline{\mathbf{X}}^T)^T \partial_{\boldsymbol{\mu}} \overline{\mathbf{X}}] \\
&= \partial_{\overline{\mathbf{X}}} \mathrm{Tr}(\overline{\mathbf{X}} \boldsymbol{\Sigma}^{-1} \overline{\mathbf{X}}^T) \cdot \mathrm{Tr}[\partial_{\boldsymbol{\mu}} \overline{\mathbf{X}}] \\
&= (\overline{\mathbf{X}} \boldsymbol{\Sigma}^{-T} + \overline{\mathbf{X}} \boldsymbol{\Sigma}^{-1}) \cdot (-n) \\
&= \mathrm{Tr}(\overline{X}_{ij} \Sigma^{-1}_{jk}) = \sum_l \overline{X}_{lj} \Sigma^{-1}_{jl} \\
&= \sum_l (X_{lj} - \mu_j) \\
\implies \sum_l X_{lj} &= n\mu_j \\
\implies \boldsymbol{\mu}_{MLE} &= \frac{1}{n} \sum_i \mathbf{x}_i
\end{aligned}$$

Now to prove the second result, we want to differentiate L w.r.t $\boldsymbol{\Sigma}$, which will require the following additional results from [2] (equation 57 and 124):

$$\partial_{\mathbf{X}} \log\det \mathbf{X} = \mathbf{X}^{-T} \tag{3}$$

$$\partial_{\mathbf{X}} \mathrm{Tr}(\boldsymbol{A}\mathbf{X}^{-1}\boldsymbol{B}) = -\mathbf{X}^{-T}\boldsymbol{A}^T\boldsymbol{B}^T\mathbf{X}^{-T} \tag{4}$$

And so now we get

$$\begin{aligned}
\partial_{\boldsymbol{\Sigma}} L &= -\frac{n}{2}\partial_{\boldsymbol{\Sigma}} \log|\boldsymbol{\Sigma}| - \frac{1}{2}\partial_{\boldsymbol{\Sigma}} \mathrm{Tr}(\overline{\mathbf{X}} \boldsymbol{\Sigma}^{-1} \overline{\mathbf{X}}^T) \\
0 &= n\boldsymbol{\Sigma}^{-T} - \boldsymbol{\Sigma}^{-T}\overline{\mathbf{X}}^T\overline{\mathbf{X}}\boldsymbol{\Sigma}^{-T} \\
\implies \boldsymbol{\Sigma}_{MLE} &= \frac{1}{n}\overline{\mathbf{X}}^T\overline{\mathbf{X}}
\end{aligned}$$

A.2 Bias-Variance Decomposition

Prove that

$$\begin{aligned}
\mathrm{E}_D[(y - f_{LS}(\mathbf{x}_i))^2 \mid \mathbf{x}_i] &= \mathrm{Var}_\epsilon[\epsilon_i] + (g(\mathbf{x}_i) - \mathrm{E}_\epsilon[f_{LS}(\mathbf{x}_i) \mid \mathbf{x}_i])^2 + \mathrm{Var}_\epsilon[f_{LS}(\mathbf{x}_i) \mid \mathbf{x}_i] \\
&= \mathrm{E}_\epsilon[(y - f_{LS}(\mathbf{x}_i))^2 \mid \mathbf{x}_i] = \mathrm{E}_\epsilon[(g(\mathbf{x}_i) + \epsilon_i - f(\mathbf{x}_i))^2 \mid \mathbf{x}_i] \\
&= \mathrm{E}_\epsilon[\epsilon_i^2] + \mathrm{E}_\epsilon[g(\mathbf{x}_i)^2 + f(\mathbf{x}_i)^2 - 2g(\mathbf{x}_i)f(\mathbf{x}_i) + 2\epsilon_i g(\mathbf{x}_i) - 2\epsilon_i f(\mathbf{x}_i) \mid \mathbf{x}_i] \\
&= \mathrm{Var}_\epsilon[\epsilon] + \mathrm{Var}_\epsilon[f(\mathbf{x}_i) \mid \mathbf{x}_i] + \mathrm{E}_\epsilon[f(\mathbf{x}_i) \mid \mathbf{x}_i]^2 + g(\mathbf{x}_i)^2 - 2\,\mathrm{E}_\epsilon[f(\mathbf{x}_i)g(\mathbf{x}_i) \mid \mathbf{x}_i] \\
&= \mathrm{Var}_\epsilon[\epsilon_i] + (g(\mathbf{x}_i) - \mathrm{E}_\epsilon[f_{LS}(\mathbf{x}_i) \mid \mathbf{x}_i])^2 + \mathrm{Var}_\epsilon[f_{LS}(\mathbf{x}_i) \mid \mathbf{x}_i]
\end{aligned}$$

where we have first used the Law of the Unconcious Statistician to change the expectation over D to one over ϵ and then used that $\mathrm{E}_\epsilon[\epsilon] = 0$ and $\mathrm{Var}_\epsilon[f] = \mathrm{E}_\epsilon[f^2] - \mathrm{E}_\epsilon[f]^2$ and hence that $\mathrm{Var}_\epsilon[\epsilon] = \mathrm{E}_\epsilon[\epsilon^2]$.

A.2.1 Unbiasedness of least squares

Prove that for $g(\mathbf{x}) = f(\mathbf{x}; \mathbf{w}^*) = \langle \mathbf{w}^*, \mathbf{x} \rangle$ the bias term vanishes.

$$\begin{aligned} \mathrm{E}[f_{LS} \mid \mathbf{x}_i] &= \mathrm{E}[\mathbf{w}^T \mathbf{x}_i] \\ &= \mathrm{E}[g(\mathbf{X})^T \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_i] \\ &= \mathbf{w}^{*T}\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_i \\ &= \mathbf{w}^{*T}\mathbf{x}_i \\ &= g(\mathbf{x}_i) \end{aligned}$$

Where we have used that $y_i = g(\mathbf{x}_i) + \epsilon_i$ to get $\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(g(\mathbf{X}) + \boldsymbol{\epsilon})$ and that $\mathrm{E}[\epsilon] = 0$.

A.2.2 Explicit model variance

Additionally, prove that $\mathrm{Var}[f_{LS} \mid \mathbf{x}_0] = \sigma^2 \langle \boldsymbol{h}, \boldsymbol{h} \rangle$ where $\boldsymbol{h}^T = \mathbf{x}_0^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T, \boldsymbol{h} \in R^n$.

$$\begin{aligned} \mathrm{Var}[f_{LS} \mid \mathbf{x}_0] &= \mathrm{E}[f_{LS}^2 \mid \mathbf{x}_0] - \mathrm{E}[f_{LS} \mid \mathbf{x}_0]^2 \\ &= \mathrm{E}[(\mathbf{w}^T\mathbf{x}_0)^2] - \mathrm{E}[\mathbf{w}^T\mathbf{x}_0]^2 \end{aligned}$$

Where we again use the substitution for y and the properties of our additive noise to get (and that the transpose of a scalar is the original scalar):

$$\begin{aligned} \mathrm{E}[\mathbf{w}^T\mathbf{x}_0]^2 &= (g(\mathbf{X})^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_0)^2 \\ \mathrm{E}[(\mathbf{w}^T\mathbf{x}_0)^2] &= \mathrm{E}[\mathbf{w}^T\mathbf{x}_0]^2 + \mathrm{E}[(\boldsymbol{\epsilon}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_0)^2] \\ \implies \mathrm{Var}[f_{LS} \mid \mathbf{x}_0] &= \mathbf{x}_0^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\,\mathrm{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T]\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_0 \\ &= \sigma^2\langle \boldsymbol{h}, \boldsymbol{h} \rangle \end{aligned}$$

A.2.3 Out-of-sample error

Calculation of the error over the entire distribution of $\mathbf{x}$:

$$\begin{aligned} \mathrm{E}_{\mathbf{x}}\,\mathrm{E}_{\epsilon}[(y - f_{LS}(\mathbf{x}))^2 \mid \mathbf{x}] &= \mathrm{E}_{\mathbf{x}}\,\mathrm{E}_y[(y - f_{LS}(\mathbf{x}))^2 \mid \mathbf{x}] \\ &= \mathrm{E}_{p(y,\mathbf{x})}[(y - f_{LS}(\mathbf{x}))^2] \\ &\simeq \mathrm{E}_{p(y,\mathbf{x})}[y - f_0(\mathbf{x})^2] \\ &\simeq \frac{1}{n'} \sum_{(y',\mathbf{x}')} (y' - f_0(\mathbf{x}'))^2 \end{aligned}$$

where we have assumed that $f_{LS}(\mathbf{x})$ is well approximated by $f_0(\mathbf{x})$ and that, by the law of large numbers, the average error over the (sufficiently large) dataset D_1 should be close to the desired expected value.

A.3 Kernel Methods

A.3.1 Least squares solution

Rewrite the least squares solution ($\mathbf{w}_{LS-R} = (\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \lambda\mathbf{I})^{-1}\boldsymbol{\Phi}^T\mathbf{y}$) using the Woodbury Identity: $(\boldsymbol{P}^{-1} + \boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T = \boldsymbol{P}\boldsymbol{B}^T(\boldsymbol{B}\boldsymbol{P}\boldsymbol{B}^T + \mathbf{I})^{-1}$. First identify that

$\boldsymbol{P}^{-1} = \lambda\mathbf{I}$, $\boldsymbol{B} = \boldsymbol{\Phi}$ and hence $\boldsymbol{P} = \frac{1}{\lambda}\mathbf{I}$.

$$\begin{aligned}
\mathbf{w}_{LS-R} &= (\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \lambda\mathbf{I})^{-1}\boldsymbol{\Phi}^T\mathbf{y} \\
&= \frac{1}{\lambda}\mathbf{I}\boldsymbol{\Phi}^T(\frac{1}{\lambda}\boldsymbol{\Phi}\mathbf{I}\boldsymbol{\Phi}^T + \mathbf{I})^{-1}\mathbf{y} \\
&= \frac{1}{\lambda}\boldsymbol{\Phi}^T\lambda(\boldsymbol{\Phi}\boldsymbol{\Phi}^T + \lambda\mathbf{I})^{-1}\mathbf{y} \\
&= \boldsymbol{\Phi}^T(\boldsymbol{\Phi}\boldsymbol{\Phi}^T + \lambda\mathbf{I})^{-1}\mathbf{y}
\end{aligned}$$

A.3.2 Induced feature transform

In general, any given k will *induce* a related feature transform $\phi(x)$. For linear, it's clearly the linear transform $\phi(x) = x$. For the polynomial kernel we can use a binomial expansion to derive the induced feature transform:

$$\begin{aligned}
k(\mathbf{x}, \mathbf{x}') &= ((\langle\mathbf{x}, \mathbf{x}'\rangle + 1)^b) \\
&= \sum_{k=0}^{b} \binom{b}{k} \langle\mathbf{x}, \mathbf{x}'\rangle^k
\end{aligned}$$

Now if we initially consider just the case when $b = 2$, we can expand the brackets by hand and see a concrete example:

$$\begin{aligned}
(\langle\mathbf{x}, \mathbf{x}'\rangle + 1)^2 &= 1 + \langle\mathbf{x}, \mathbf{x}'\rangle^2 + 2\langle\mathbf{x}, \mathbf{x}'\rangle \\
&= 1 + \sum_{ij}^{d} x_i x_j x'_i x'_j + 2\sum_i x_i x'_i
\end{aligned}$$

From this, we can define the $b = 2$ feature transform as

$$\boldsymbol{\phi}^{(2)}(\mathbf{x}) = [1, \sqrt{2}\mathbf{x}^T, (\mathbf{x} \otimes \mathbf{x})^T]^T$$

with $\otimes$ defined to be the Kronecker product where the product of two column vectors $\boldsymbol{u} \in R^p$ and $\boldsymbol{v} \in R^q$ is a third column vector $\boldsymbol{w} \in R^{pq}$ defined by the vectorization of the outer product of $\boldsymbol{u}$ and $\boldsymbol{v}$.

Hence we can check that the inner product of the feature transform correctly reproduces the $b = 2$ kernel:

$$\begin{aligned}
\langle\boldsymbol{\phi}^{(2)}(\mathbf{x}), \boldsymbol{\phi}^{(2)}(\mathbf{x}')\rangle &= [1, \sqrt{2}\mathbf{x}^T, (\mathbf{x} \otimes \mathbf{x})^T][1, \sqrt{2}\mathbf{x}^T, (\mathbf{x} \otimes \mathbf{x})^T]^T \\
&= 1 + 2\mathbf{x}^T\mathbf{x}' + \langle\mathbf{x} \otimes \mathbf{x}, \mathbf{x}' \otimes \mathbf{x}'\rangle
\end{aligned}$$

where we can see that the last term is correct if we consider the vector $\mathbf{x}$ in terms of the basis $\{\boldsymbol{e}_i\}$, i.e. $\mathbf{x} = \sum_i x_i \boldsymbol{e}_i$. With this, we get the following:

$$\begin{aligned}
\langle \mathbf{x} \otimes \mathbf{x}, \mathbf{x}' \otimes \mathbf{x}' \rangle &= \sum_{ijkl} x_i x_j x'_k x'_l \langle \boldsymbol{e}_i \otimes \boldsymbol{e}_j, \boldsymbol{e}_k \otimes \boldsymbol{e}_l \rangle \\
&= \sum_{ijkl} x_i x_j x'_k x'_l \langle \boldsymbol{e}_i, \boldsymbol{e}_k \rangle \otimes \langle \boldsymbol{e}_j, \boldsymbol{e}_l \rangle \\
&= \sum_{ijkl} x_i x_j x'_k x'_l \delta_{ik} \delta_{jl} \\
&= \sum_{ij} x_i x_j x'_i x'_j
\end{aligned}$$

Where we have taken the inner product to be equivalent to summing over all indices the Hadamard (elementwise) product, i.e., for the usual vector inner product we have that $\langle \boldsymbol{a}, \boldsymbol{b} \rangle = \sum_i (\boldsymbol{a} \odot \boldsymbol{b})_i = \sum_i a_i b_i$. For a rank-2 tensor we have

$$\begin{aligned}
\langle \boldsymbol{a} \otimes \boldsymbol{b}, \boldsymbol{c} \otimes \boldsymbol{d} \rangle &= \sum_{ij} ((\boldsymbol{a} \otimes \boldsymbol{b}) \odot (\boldsymbol{c} \otimes \boldsymbol{d})) \\
&= \sum_{ij} (a_i b_j c_i d_j) \\
&= \sum_{ij} ((\boldsymbol{a} \odot \boldsymbol{c}) \otimes (\boldsymbol{b} \odot \boldsymbol{d}) \\
&= \langle \boldsymbol{a}, \boldsymbol{c} \rangle \otimes \langle \boldsymbol{b}, \boldsymbol{d} \rangle
\end{aligned}$$

which is a scalar, as intended (since $s \otimes t$ is a scalar).

Now, generalizing to arbitrary order tensors created with the tensor product by induction assuming that the above relationship holds for a rank k tensor:

$$\begin{aligned}
(\boldsymbol{a}_1 \otimes \cdots \otimes \boldsymbol{a}_{k+1}) \odot (\boldsymbol{b}_1 \otimes \cdots \otimes \boldsymbol{b}_{k+1}) &= \left[\underbrace{(\boldsymbol{a}_1 \otimes \cdots \otimes \boldsymbol{a}_k)}_{\boldsymbol{A}} \otimes \boldsymbol{a}_{k+1} \right] \odot \left[\underbrace{(\boldsymbol{b}_1 \otimes \cdots \otimes \boldsymbol{b}_k)}_{\boldsymbol{B}} \otimes \boldsymbol{b}_{k+1} \right] \\
&= \boldsymbol{A} \odot \boldsymbol{B} \otimes \boldsymbol{a}_{k+1} \odot \boldsymbol{b}_{k+1} \\
&= (\boldsymbol{a}_1 \odot \boldsymbol{b}_1 \otimes \cdots \otimes \boldsymbol{a}_k \odot \boldsymbol{b}_k) \otimes (\boldsymbol{a}_{k+1} \odot \boldsymbol{b}_{k+1}) \\
&= (\boldsymbol{a}_1 \odot \boldsymbol{b}_1 \otimes \cdots \otimes \boldsymbol{a}_{k+1} \odot \boldsymbol{b}_{k+1})
\end{aligned}$$

This gives us the generic order inner product that we need:

$$\langle \mathbf{x}^{\otimes k}, \mathbf{x}'^{\otimes k} \rangle = \sum_{i_1, \ldots, i_k}^{d} x_{i_1} \cdots x_{i_k} x'_{i_1} \cdots x'_{i_k}$$

Note that here the above proofs do not rely on the vectorization of the outer product.

Finally putting this together with the b order kernel, we have the general feature transform:

$$\begin{aligned}\boldsymbol{\phi}^{(b)}(\mathbf{x})^T &= \left[1, \sqrt{\binom{b}{1}}\mathbf{x}, \sqrt{\binom{b}{2}}\mathbf{x}\otimes\mathbf{x}, \ldots, \sqrt{\binom{b}{b}}\mathbf{x}\otimes\cdots\otimes\mathbf{x}\right] \\ &= \bigoplus_{k=0}^{b}\sqrt{\binom{b}{k}}\mathbf{x}^{\otimes k}\end{aligned}$$

where $\mathbf{x}^{\otimes k}$ is understood to mean $\mathbf{x}\underbrace{\otimes\cdots\otimes}_{k}\mathbf{x}$.

So we find that the feature transform is the polynomial feature we defined before, but with a binomial coefficient prefactor at each term.

A.4 Model Selection

A.4.1 Model Evidence

Given the model evidence, $p(D|m) = \int p(D|\mathbf{w}, m)p(\mathbf{w}|m)d\mathbf{w}$ we approximate it by assuming that we can represent the posterior as a step function centred on $\mathbf{w}_{MAP}$ s.t. $\int p(D \mid \mathbf{w}, m)p(\mathbf{w} \mid m)d\mathbf{w} \simeq p(D \mid \mathbf{w}MAP, m)p(\mathbf{w}_{MAP} \mid m)\Delta\text{posterior}$. If we also assume that the prior can be similarly represented as a uniform distribution, s.t. it's density can be represented by $p(\mathbf{w} \mid m) \simeq \frac{1}{\Delta\text{prior}}$ then we get $p(D \mid m) \simeq p(D \mid \mathbf{w}_{MAP}, m)\frac{\Delta\text{posterior}}{\Delta\text{prior}}$. If we now consider the log evidence and a set of prior and posterior distributions (which we denote $q(\mathbf{w} \mid \boldsymbol{\theta})$ and $q(\mathbf{w} \mid D, \boldsymbol{\theta})$ respectively) over $\mathbf{w}$ that can be factorized into identical independent distributions for each w_i and s.t. their ratio is constant for all w_i then we get that

$$\begin{aligned}\text{prior} = q(\mathbf{w} \mid \boldsymbol{\theta}) &= \prod_i^b q(w_i \mid \theta_i) = \prod_i^b q(w_i \mid \theta) \\ \text{posterior} = q(\mathbf{w} \mid D, \boldsymbol{\theta}) &= \prod_i^b q(w_i \mid D, \theta_i) = \prod_i^b q(w_i \mid D, \theta) \\ \implies \frac{\Delta\text{posterior}}{\Delta\text{prior}} &= \left(\prod_i^b \frac{q(w_i \mid D, \theta)}{q(w_i \mid \theta)}\right) \\ &= \left(\frac{\delta\text{posterior}}{\delta\text{prior}}\right)^b \\ \implies \log p(D \mid m) &= \log p(D \mid \mathbf{w}_{MAP}, m) + b\log\frac{\Delta\text{posterior}}{\Delta\text{prior}}\end{aligned}$$

where δposterior represents the posterior for a single index of $\mathbf{w}$ and then in the last line we have relabelled $\delta\text{posterior} \to \Delta\text{posterior}$.

A.4.2 Marginal Likelihood

If we take the predictive distribution

$$\begin{aligned} p(\hat{y} \mid D) &= \int p(\hat{y} \mid D, \alpha) p(\alpha \mid D) d\alpha \\ &= \int \int p(\hat{y} \mid \mathbf{w}, \alpha) p(\mathbf{w} \mid D\alpha) p(\alpha \mid D) d\mathbf{w} d\alpha \end{aligned}$$

and take that $p(\alpha \mid D) \simeq \delta(\alpha - \hat{\alpha})$ then we immediately get

$$p(\hat{y} \mid D) = \int p(\hat{y} \mid \mathbf{w}, \hat{\alpha}) p(\mathbf{w} \mid D, \hat{\alpha}) d\mathbf{w}.$$

To find α we want to maximize $p(\alpha \mid D)$ which we do by using Bayes rule again:

$$p(\alpha \mid D) \propto p(D \mid \alpha) p(\alpha) = p(\alpha) \int p(D \mid \mathbf{w}, \alpha) p(\mathbf{w} \mid \alpha) d\mathbf{w}$$

and finally, with the assumption that $p(\alpha)$ is approximately constant, we are left with the *marginal likelihood maximization*

$$\hat{\alpha} := \underset{\alpha}{\operatorname{argmax}} \int p(D \mid \mathbf{w}, \alpha) p(\mathbf{w} \mid \alpha) d\mathbf{w}$$

Using the example of a linear regression model, with the following setup

$$\begin{aligned} p(\{y_i\} \mid \{\mathbf{x}_i\}; \mathbf{w}, b) &:= \prod_{i \in D} N_{y_i}(\langle \mathbf{w}, \boldsymbol{\phi}_b(\mathbf{x}_i) \rangle, \sigma^2) \\ p(\mathbf{w}) &:= N_{\mathbf{w}}(\mathbf{0}, \sigma_{\mathbf{w}}^2 \mathbf{I}) \end{aligned}$$

calculate the marginal likelihood

$$p(\{y_i\} \mid \{\mathbf{x}_i\}; b, \sigma, \sigma_{\mathbf{w}}) = \int p(\{y_i\} \mid \{\mathbf{x}_i\}; \mathbf{w}, b, \sigma, \sigma_{\mathbf{w}}) p(\mathbf{w}) d\mathbf{w}.$$

To compute this, we do the usual procedure, and collect terms in $\mathbf{w}$, do the Gaussian integral and are left with a Gaussian in $\mathbf{y}$. Starting with the full integral expression:

$$\int N_{\mathbf{y}}(\boldsymbol{\Phi}_b(\mathbf{X})\mathbf{w}, \sigma^2 \mathbf{I}_n) N_{\mathbf{w}}(\mathbf{0}, \sigma_{\mathbf{w}}^2 \mathbf{I}_b) d\mathbf{w}$$

Expanding the squares, dropping arguments and subscripts (for brevity), collecting terms and only writing out the terms in the exponentials:

$$\begin{aligned} & -\mathbf{w}^T \underbrace{\frac{1}{\sigma^2} (\boldsymbol{\Phi}^T \boldsymbol{\Phi} + \frac{\sigma^2}{\sigma_{\mathbf{w}}^2} \mathbf{I}_b)}_{\boldsymbol{\Sigma}^{-1}} \mathbf{w} + 2\mathbf{y}^T \boldsymbol{\Phi} \mathbf{w} - \frac{1}{\sigma^2} \mathbf{y}^T \mathbf{y} \\ &= -\frac{1}{\sigma^2} \mathbf{y}^T \mathbf{y} - (\mathbf{w} - \boldsymbol{\Sigma} \boldsymbol{\Phi}^T \mathbf{y})^T \boldsymbol{\Sigma}^{-1} (\mathbf{w} - \boldsymbol{\Sigma} \boldsymbol{\Phi}^T \mathbf{y}) + \mathbf{y}^T \boldsymbol{\Phi} \boldsymbol{\Sigma} \boldsymbol{\Phi}^T \mathbf{y}. \end{aligned}$$

We then do the $\mathbf{w}$ integral and collect the remaining terms in $\mathbf{y}$ (ignoring constant prefactors) and then rewrite the Woodbury Identity [2]:

$$A^{-1} U (B^{-1} + V^T A^{-1} U)^{-1} V^T A^{-1} = A^{-1} - (A + UBV^T)^{-1}$$

from which we can see that

$$-\frac{1}{\sigma^2} \mathbf{I} + \boldsymbol{\Phi} (\boldsymbol{\Phi}^T \boldsymbol{\Phi} + \lambda \mathbf{I})^{-1} \boldsymbol{\Phi}^T = -(\mathbf{I} + \frac{1}{\lambda} \boldsymbol{\Phi} \boldsymbol{\Phi}^T)^{-1}$$

by identifying $A^{-1} = \frac{1}{\sigma^2}\mathbf{I}$, $U = \mathbf{\Phi}$, $V = \mathbf{\Phi}$ and $B^{-1} = \frac{1}{\sigma^2_{\mathbf{w}}}\mathbf{I}$ we get the final result for the covariance:

$$-(\sigma^2\mathbf{I} + \sigma^2_{\mathbf{w}}\mathbf{\Phi}\mathbf{\Phi}^T)^{-1}$$

References

[1] Christopher M Bishop. *Pattern recognition and machine learning.* Information science and statistics. Springer, New York, NY, 2006. Softcover published in 2016.

[2] K. B. Petersen and M. S. Pedersen. The matrix cookbook, October 2008. Version 20081110.