In this portfolio, we provide a more robust justification for previously used testing methods and we explore linear functions $f(\mathbf{x}; \mathbf{w})$ beyond polynomial transforms that might approximate the relation between our input vector $\mathbf{x}$ and output y. A description of the general model used for this portfolio is provided in Appendix A.

1 Creating a Generalisable Model

Returning to a polynomial transform setting, we again split our dataset D into testing and training sets, but instead of considering the testing error, we focus on its expectation:

$$\mathbb{E}_D[\mathbb{E}(D_1, \mathbf{w}_{LS})] = \sum_i \mathbb{E}_D[(y_i - f(\mathbf{x}_i; \mathbf{w}_{LS}))^2 | \mathbf{x_i}] \tag{1}$$

The term inside the sum is called the expected loss, which we explore in the next section.

1.1 Expected Loss Function

We find the expected loss can be decomposed into a sum of three terms:

$$\mathbb{E}_D[(y_i - f(\mathbf{x}_i; \mathbf{w}_{LS}))^2 | \mathbf{x}_i] = \text{var}(\epsilon) + [g(\mathbf{x}) - \mathbb{E}_D(f_{LS}(\mathbf{x}_i)|\mathbf{x}_i)]^2 + \text{var}[f_{LS}(\mathbf{x}_i)|\mathbf{x}_i]$$

The proof of this is given in Appendix B. We describe these terms as follows:

- **Irreducible Error:** this is the error inherent in our data generation process, and so is fixed for any model we choose.
- **Bias:** This measures the accuracy of our prediction function.
- **Variance:** this shows how much our prediction function varies when our random data changes

1.2 Bias-Variance Trade-Off and Overfitting

It then makes sense to consider how the expected loss changes with respect to b. As b increases:

- f_{LS} is more flexible so the bias decreases
- f_{LS} also becomes more sensitive to the data so the variance term increases

It then makes sense to choose a b that balances these two factors by minimising the expected loss function. Considering this, the practice of choosing the b which results in the smallest testing error is mathematically sensible, as well as intuitive.

1.3 Out-Sample Error and Cross-Validation

Another quantity to consider is the error over the entire distribution of $\mathbf{x}$ (the out-sample error:

$$\mathbb{E}_{\mathbf{x}}\mathbb{E}\left[(y - f_{LS}(\mathbf{x}))^2 | \mathbf{x}\right] = \mathbb{E}_{D_0}\mathbb{E}_{p(y,\mathbf{x})}\left[(y - f_{LS}(\mathbf{x}))^2\right]$$

We can split D into disjoint $D^{(1)}, \ldots, D^{(K)}$ and further split these into testing and training sets $D_0^{(k)}, D_1^{(k)}$ for each k. We can then fit $f_{LS}^{(k)}$ trained on $D_0^{(k)}$. Since the out-sample error is a theoretical quantity, under mild conditions, we can approximate it by

$$\mathbb{E}_{D_0}\mathbb{E}_{p(y,\mathbf{x})}\left[(y - f_{LS}(\mathbf{x}))^2\right] \approx \frac{1}{K}\sum_k \frac{1}{n'} \sum_{i \in D_1^{(k)}} \left(y_i - f_{LS}^{(k)}(\mathbf{x}_i)\right)^2$$

We can see this is the objective function for k-fold cross-validation, thus if we want to minimise this error, we need to use k-fold cross-validation.

2 Feature Transforms

We can consider the Taylor Expansion near 0 and compare it to a polynomial feature transform to see why this type of transform might be appropriate:

$$\textbf{Taylor Expansion: } g(x) = g(0) + g'(0)x + \frac{g''(0)}{2!}x^2 + \dots$$
$$\textbf{Polynomial Transform: } [x, x^2, \dots, x^b] \in \mathbb{R}^b$$

A similar argument can be provided for a trigonometric transform as shown in Appendix C.

These are examples of a linear expansion of a basis function, where we approximate a function by

$$g(\mathbf{x}) \approx f(\mathbf{x}; \mathbf{w}) = \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}) \rangle = \sum_i w^{(i)} \phi^{(i)}(\mathbf{x})$$

where $\phi^{(i)}$ are called the basis functions. An interesting set of basis functions is described below.

2.1 Radial Basis Functions

These basis functions are commonly used for regression:

$$\phi^{(i)}(\mathbf{x}) \coloneqq \exp\left(-\frac{||\mathbf{x} - \mathbf{x}_i||}{2\sigma^2}\right)$$

where $\sigma > 0$ is called the bandwidth, to be determined before fitting and $\mathbf{x}_i$ are called RBF centroids and can be chosen randomly from the dataset. We can see each $\phi^{(i)}$ defines a ball centered at $\mathbf{x}_i$ with radius σ, and so f is supported over the union of these balls:

$$\text{support}(f) = \{\mathbf{x} : f(\mathbf{x}; \mathbf{w}) \neq \mathbf{0}\}$$

We can see that for f to have wider support, we need more centroids. Hence we encounter the curse of dimensionality as our number of basis functions increases with the dimension.

2.2 Feature Space

In our context, $\boldsymbol{\phi}(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}^b$ and we call $\mathbb{R}^b$ our feature space. To be able to increase the flexibility of f, we might wonder if it is possible to consider a $\boldsymbol{\phi}$ that produces an infinite dimensional feature space.

While the expression for the regularised least squares solution would then contain the intractable $\boldsymbol{\Phi}\boldsymbol{\Phi}^T$, by the proof given in Appendix C.2, we can rewrite the solution as

$$\mathbf{w}_{LS-R} = \boldsymbol{\Phi}\left(\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \lambda\mathbf{I}\right)^{-1}\mathbf{y}^T$$

2.3 Kernel Functions

We can define a function $k(\mathbf{x}, \mathbf{y}) \coloneqq \langle \boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{y}) \rangle$ and a matrix $\mathbf{K}$ where $\mathbf{K}^{(i,j)} = k(\mathbf{x}_i, \mathbf{x}_j)$, i.e. $\mathbf{K} = \boldsymbol{\Phi}^T\boldsymbol{\Phi}$. Then our prediction function becomes:

$$\begin{aligned} f(\mathbf{x}; \mathbf{w}_{LS-R}) &= \left\langle \boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\Phi}(\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{y}^T \right\rangle = \left\langle \boldsymbol{\phi}(\mathbf{x})\boldsymbol{\Phi}, (\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{y}^T \right\rangle \\ &= \mathbf{k}(\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{y}^T \end{aligned}$$

and so $\boldsymbol{\phi}(\mathbf{x})$ only appears inside the inner product vector $\mathbf{k}$.

This means if we can construct a function k that behaves an inner product on our feature space, we do not have to compute $\boldsymbol{\phi}(\mathbf{x})$ explicitly. Such a function k is called a kernel function, and we say k induces

$\boldsymbol{\phi}(\mathbf{x})$. The choice of such a kernel function will depend on the task and dataset, so we usually rely on subject knowledge. Some examples of kernel functions one might use provided in Appendix C.3, as well as an example of how we might find the induced feature transform when provided with a kernel function.

While we have the benefit of flexibility, it is important to note the major drawback, which is the computational cost. The computation time of $(\mathbf{K}+\lambda\mathbf{I})^{-1}$ is of order n^3, so is computationally expensive.

3 Bayesian Model Selection

In this approach, we build use priors on parameters and obtain posteriors which capture the uncertainty of the model and data while still accounting for our beliefs about the model:

$$p(m|D) \propto p(D|m)p(m)$$

where $p(m)$ is the model prior and $p(D|m)$ is the model evidence.

3.1 Marginalisation

Rather than selecting one most likely model, we take a weighted average of plausible models:

$$p(\hat{y}|D) = \sum_m p(\hat{y}|D,m)p(m|D)$$

This means if two models give different predictions, we choose a compromise of the two predictions, using both to inform decision making.

As shown above, we need to calculate the model evidence to obtain the posterior. If our model contains parameters $\mathbf{w}$, then

$$p(D|m) = \int p(D|\mathbf{w},m)p(\mathbf{w}|m)d\mathbf{w}$$

While in practice the following assumptions and approximations won't hold, they are helpful in motivating the problem.

3.2 Approximations

Posterior Step Function We first consider the case where $p(\mathbf{w}|D,m)$ is a step function with a plateau at $\mathbf{w}_{MAP}$. We can then call the length of this step $\Delta_{\text{posterior}}$. We can now approximate the integral by:

$$\int p(D|\mathbf{w},m)p(\mathbf{w}|m)d\mathbf{w} \approx p(D|\mathbf{w}_{MAP},m)p(\mathbf{w}_{MAP}|m)\cdot\Delta_{\text{posterior}}$$

Prior Step Function If similarly $p(\mathbf{w}|m)$ has a plateau, the length of which we will call Δ_{prior},

$$p(D|m) \approx p(D|\mathbf{w}_{MAP},m)\frac{\Delta_{\text{posterior}}}{\Delta_{\text{prior}}}$$

Since the posterior is generally more specific and informed by data, it is sharper so it's reasonable to assume $\Delta_{\text{posterior}} < \Delta_{\text{prior}}$.

3.2.1 Model With b Parameters

Suppose we have b and if we assume $\frac{\Delta_{\text{posterior}}}{\Delta_{\text{prior}}}$ is the same for all w_i, with all w_i independent of each other,

$$\log p(D|m) \approx \log p(D|\mathbf{w}_{MAP},m) + b\log\frac{\Delta_{\text{posterior}}}{\Delta_{\text{prior}}}$$

with the proof of this given in Appendix D.1 If b increases:

- $b \log \frac{\Delta_{\text{posterior}}}{\Delta_{\text{prior}}}$ decreases
- $\log p(D|\mathbf{w}_{MAP}, m)$ increases

Hence the model evidence balances these two quantities and chooses an intermediate model complexity.

3.3 Tuning Hyper Parameters

If our model includes a hyper parameter α, the predictive distribution is

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

which is usually intractable.

Again, we can make a couple of assumptions on α to obtain a workable expression for α.

"Pointy" Posterior If $p(\alpha|D)$ is pointy at $\hat{\alpha}$, we can approximate this by

$$p(\hat{y}|D) \approx \int p(\hat{y}|\mathbf{w},\hat{\alpha})p(\mathbf{w}|D,\alpha)d\mathbf{w}$$

Then by definition $\hat{\alpha}$ is the maximiser of

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

Flat Prior If we also assume $p(\alpha)$ is sufficiently flat, $\hat{\alpha}$ is

$$\hat{\alpha} \coloneqq \arg\max_{\alpha} \int p(D|\mathbf{w},\alpha)p(\mathbf{w}|\alpha)d\mathbf{w}$$

3.4 Linear Regression Example

We have the following model:

$$p(y_1,\dots,y_n|\mathbf{x}_1,\dots,\mathbf{x}_n;\mathbf{w},b) \coloneqq \prod_{i\in D}\mathcal{N}_{y_i}\left(\left\langle \mathbf{w}, \boldsymbol{\phi}^b(\mathbf{x}_i)\right\rangle, \sigma^2\mathbf{I}\right)$$
$$p(\mathbf{w};\sigma_{\mathbf{w}},b) \coloneqq \mathcal{N}_{\mathbf{w}}(\mathbf{0},\sigma_{\mathbf{w}}^2\mathbf{I}_b)$$

Then the marginalised likelihood is

$$p(y_1,\dots,y_n|\mathbf{x}_1,\dots,\mathbf{x}_n;b,\sigma,\sigma_{\mathbf{w}}) = \int p(y_1,\dots,y_n|\mathbf{x}_1,\dots,\mathbf{x}_n;\mathbf{w},b,\sigma,\sigma_{\mathbf{w}})p(\mathbf{w})d\mathbf{w}$$

Rather than evaluating this integral though, we can instead use the Gaussian identity, provided in Appendix D.2, as the mean of $\mathbf{y}$ is a linear combination of $\mathbf{w}$. Doing this, we see

$$\boldsymbol{\mu}_{\mathbf{y}} = \boldsymbol{\Phi}^T\mathbf{0} + \mathbf{0} = \mathbf{0}$$
$$\boldsymbol{\Sigma}_{\mathbf{y}} = \sigma_{\mathbf{w}}^2\mathbf{I}_b + \boldsymbol{\Phi}^T\sigma^2\mathbf{I}\boldsymbol{\Phi} = \sigma_{\mathbf{w}}^2\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \sigma^2\mathbf{I}$$

and so we get

$$p(y_1,\dots,y_n|\mathbf{x}_1,\dots,\mathbf{x}_n;b,\sigma,\sigma_{\mathbf{w}}) = \mathcal{N}_{\mathbf{y}}(\mathbf{0},\sigma_{\mathbf{w}}^2\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \sigma^2\mathbf{I})$$

A Model

Suppose we have covariates $\mathbf{x}_i$ which we believe to contribute to the output y_i. More precisely we write

$$y_i = g(\mathbf{x}_i) + \epsilon_i; \text{ where}$$

- $g(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}$ is a deterministic function
- ϵ_i is independent of $\mathbf{x}_i$ for each i and $\mathbb{E}(\epsilon) = 0$ (called additive noise)

For simplicity we also assume $\mathbf{x}_i$ is fixed.

B Generalisable Model

Proof of Bias-Variance Decomposition

We can expand the square term to get:

$$\mathbb{E}[(y_i - f(\mathbf{x}_i; \mathbf{w}_{LS}))^2 | \mathbf{x_i}] = \mathbb{E}[y_i^2 | \mathbf{x}_i] - 2\mathbb{E}[y_i f_{LS}(\mathbf{x}_i) | \mathbf{x}_i] + \mathbb{E}[f_{LS}(\mathbf{x}_i)^2 | \mathbf{x}_i]$$

where the D subscript has been dropped but is still implicitly being considered. We can consider the first two terms:

$$\begin{aligned}
\mathbb{E}[y_i^2 | \mathbf{x}_i] &= \mathbb{E}[(g(\mathbf{x}_i) + \epsilon)^2] = g(\mathbf{x}_i)^2 + 2g(\mathbf{x}_i)\mathbb{E}[\epsilon] + \mathbb{E}[\epsilon^2] = g(\mathbf{x}_i)^2 + \text{var}[\epsilon] \\
\mathbb{E}[y_i f_{LS}(\mathbf{x}_i) | \mathbf{x}_i] &= \mathbb{E}[(g(\mathbf{x}_i) + \epsilon) f_{LS}(\mathbf{x}_i) | \mathbf{x}_i] = g(\mathbf{x}_i)\mathbb{E}[f_{LS}(\mathbf{x}_i) | \mathbf{x}_i] + \mathbb{E}[\epsilon]\mathbb{E}[f_{LS}(\mathbf{x}_i) | \mathbf{x}_i] \\
&= g(\mathbf{x}_i)\mathbb{E}[f_{LS}(\mathbf{x}_i) | \mathbf{x}_i]
\end{aligned}$$

So the expected loss function becomes

$$\begin{aligned}
\mathbb{E}[(y_i - f(\mathbf{x}_i; \mathbf{w}_{LS}))^2 | \mathbf{x_i}] &= \text{var}[\epsilon] + g(\mathbf{x}_i)^2 - 2g(\mathbf{x}_i)\mathbb{E}[f_{LS}(\mathbf{x}_i) | \mathbf{x}_i] + \mathbb{E}[f_{LS}(\mathbf{x}_i)^2 | \mathbf{x}_i] \\
&= \text{var}[\epsilon] + g(\mathbf{x}_i)^2 - 2g(\mathbf{x}_i)\mathbb{E}[f_{LS}(\mathbf{x}_i) | \mathbf{x}_i] + \mathbb{E}[f_{LS}(\mathbf{x}_i)^2 | \mathbf{x}_i] \\
&\quad - \mathbb{E}[f_{LS}(\mathbf{x}_i)^2 | \mathbf{x}_i] + \mathbb{E}[f_{LS}(\mathbf{x}_i)^2 | \mathbf{x}_i] \\
&= \text{var}(\epsilon) + [g(\mathbf{x}) - \mathbb{E}_D(f_{LS}(\mathbf{x}_i) | \mathbf{x}_i)]^2 + \text{var}[f_{LS}(\mathbf{x}_i) | \mathbf{x}_i]
\end{aligned}$$

as desired.

C Feature Transforms

C.1 Basis Functions

Trigonometric Transform

Similarly to the Taylor Expansion, we can consider the Fourier Series for a function with a time domain to justify the use of a trigonometric transform:

$$\textbf{Fourier Series: } g(x) = a_0 + \sum_{i=1}^{\infty} [a_i \sin(ix) + b_i \cos(ix)]$$

$$\textbf{Trigonometric Transform: } \boldsymbol{\phi}(x) = [\sin(x), \cos(x), \sin(2x), \cos(2x), \ldots, \sin(bx), \cos(bx)] \in \mathbb{R}^{2b}$$

C.1.1 Radial Basis Functions

These basis functions are commonly used for regression:

$$\phi^{(i)}(\mathbf{x}) \coloneqq \exp\left(-\frac{||\mathbf{x}-\mathbf{x}_i||}{2\sigma^2}\right)$$

where $\sigma > 0$ is called the bandwidth, to be determined before fitting and $\mathbf{x}_i$ are called RBF centroids and can be chosen randomly from the dataset.

We can see each $\phi^{(i)}$ defines a ball centered at $\mathbf{x}_i$ with radius σ, and so f is supported over the union of these balls:

$$\text{support}(f) = \{\mathbf{x} : f(\mathbf{x};\mathbf{w}) \neq \mathbf{0}\}$$

We can see that for f to have wider support, we need more centroids. Hence we encounter the curse of dimensionality as our number of basis functions increases with the dimension.

C.2 Feature Space

Alternative Regularised Least Squares

For this, we can make use of the Woodburyuse of the Woodbury Identity

$$(\mathbf{P}^{-1}+\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T = \mathbf{P}\mathbf{B}^T(\mathbf{B}\mathbf{P}\mathbf{B}^T+\mathbf{I})^{-1},$$

setting $\mathbf{P} = \frac{1}{\lambda}\mathbf{I}$ and $\mathbf{B} = \mathbf{\Phi}^T$ this becomes

$$\begin{aligned}(\lambda\mathbf{I}+\mathbf{\Phi}\mathbf{\Phi}^T)^{-1}\mathbf{\Phi} &= \frac{1}{\lambda}\mathbf{I}\mathbf{\Phi}\left(\frac{1}{\lambda}\mathbf{\Phi}^T\mathbf{\Phi}+\mathbf{I}\right)^{-1}\\ &= \mathbf{\Phi}\left(\mathbf{\Phi}^T\mathbf{\Phi}+\lambda\mathbf{I}\right)^{-1}\end{aligned}$$

Then we can easily see

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

C.3 Kernel Functions

Examples of Kernel Functions

- **Linear Kernel:** $k(\mathbf{x}_i,\mathbf{x}_j) = \langle\mathbf{x}_i,\mathbf{x}_j\rangle$ induces $\boldsymbol{\phi}(\mathbf{x}) = \mathbf{x}$
- **Polynomial Kernel:** $k(\mathbf{x}_i,\mathbf{x}_j) = (\langle\mathbf{x}_i,\mathbf{x}_j\rangle+1)^b$ (see example below for induced transform)
- **Radial Basis Function (RBF) Kernel:** $k(\mathbf{x}_i,\mathbf{x}_j) = \exp\left(-\frac{||\mathbf{x}_i-\mathbf{x}_j||}{2\sigma^2}\right)$ which induces infinite dimensional $\boldsymbol{\phi}(\mathbf{x})$

Example: Finding Induced Feature Transform

In this example we consider the polynomial kernel function for order $b = 2$:

$$\begin{aligned}
k(\mathbf{x}_i, \mathbf{x}_j) &= (\langle \mathbf{x}_i, \mathbf{x}_j \rangle + 1)^2 \\
&= \left(\sum_k x_i^{(k)} x_j^{(k)} \right)^2 + 2 \sum_k x_i^{(k)} x_j^{(k)} + 1 \\
&= \sum_k \left(x_i^{(k)} \right)^2 \left(x_j^{(k)} \right)^2 + 2 \sum_{k>l} x_i^{(k)} x_i^{(l)} x_j^{(k)} x_j^{(l)} + 2 \sum_k x_i^{(k)} x_j^{(k)} + 1 \\
&= \sum_k \left(x_i^{(k)} \right)^2 \left(x_j^{(k)} \right)^2 + \sum_{k>l} \sqrt{2} x_i^{(k)} x_i^{(l)} \sqrt{2} x_j^{(k)} x_j^{(l)} + \sum_k \sqrt{2} x_i^{(k)} \sqrt{2} x_j^{(k)} + 1
\end{aligned}$$

We want to find a $\boldsymbol{\phi}$ such that $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \boldsymbol{\phi}(\mathbf{x}_i, \mathbf{x}_j) \rangle$. We can see that we need to consider first and second order terms of each element, as well as first order interaction terms. Taking into account the $\sqrt{2}$ terms, we get:

$$\boldsymbol{\phi}(\mathbf{x})^T = \left[1, \sqrt{2}x^{(1)}, ..., \sqrt{2}x^{(d)}, (x^{(1)})^2, ..., (x^{(d)})^2, \left[\sqrt{2} x^{(k)} x^{(l)} \right]_{k>l} \right]$$

We can check this works below:

$$\langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) \rangle = \sum_k \left(x_i^{(k)} \right)^2 \left(x_j^{(k)} \right)^2 + 2 \sum_{k>l} x_i^{(k)} x_i^{(l)} x_j^{(k)} x_j^{(l)} + 2 \sum_k x_i^{(k)} x_j^{(k)} + 1 = k(\mathbf{x}_i, \mathbf{x}_j)$$

as desired.

D Bayesian Model Selection

D.1 Model with b parameters

Proof of Model Evidence

We have

$$\begin{aligned}
\log p(D|m) &\approx \log \left(p(D|\mathbf{w}_{MAP}, m) \prod_{i=1}^{b} \frac{\Delta^i_{\text{posterior}}}{\Delta^i_{\text{prior}}} \right) \\
&= \log \left(p(D|\mathbf{w}_{MAP}, m) \left(\frac{\Delta_{\text{posterior}}}{\Delta_{\text{prior}}} \right)^b \right) \\
&= \log p(D|\mathbf{w}_{MAP}, m) + b \log \frac{\Delta_{\text{posterior}}}{\Delta_{\text{prior}}}
\end{aligned}$$

with the first line being valid due to the independence of the w_i and the second step due to assumption of $\frac{\Delta^i_{\text{posterior}}}{\Delta^i_{\text{prior}}}$ being the same for all w_i.

D.2 Linear Regression Example

Gaussian Identity

If we are given distributions in the following form:

$$\begin{aligned}
p(\mathbf{x}) &= \mathcal{N}(\mathbf{x}|\mu, \mathbf{\Lambda}^{-1}) \\
p(\mathbf{y}|\mathbf{x}) &= \mathcal{N}(\mathbf{A}\mathbf{x} + \mathbf{b}, \mathbf{L}^{-1})
\end{aligned}$$

we can obtain the marginal distribution of $\mathbf{y}$ as

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{L}^{-1} + \mathbf{A}\boldsymbol{\Lambda}^{-1}\mathbf{A}^T)$$