Robot Learning

Winter Semester 2020/2021, Homework 3

Prof. Dr. J. Peters, J. Watson, J. Carvalho, J. Urain and T. Dam

TECHNISCHE UNIVERSITÄT DARMSTADT

Total points: 35
Due date: Midnight, 04 January 2021
Übungsblatt 3

Aufgabe 3.1: Machine Learning in a Nutshell (35 Punkte)

For this exercise you will use a dataset, divided into training set and validation set (download them in Moodle). The first row is the vector x and the second row the vector y.
Based on this data, we want to learn a function mapping from x values to y values, of the form $y = \theta^T \phi(x)$.
Please upload the code you developed in the corresponding section in Moodle.

3.1a) Supervised vs Unsupervised Learning (2 Punkte)

Briefly explain the differences between supervised and unsupervised learning. Is the above a supervised or unsupervised learning problem? Why?

Lösungsvorschlag:

The main difference between supervised and unsupervised learning base on, whether the dataset is labeled (or using a ground truth).[1]

* Supervised learning: has a sample of data and desired outputs
* Unsupervised learning: does not have labeled outputs

Therefore, above task is a supervised problem. (Because we have the desired output y in datasets, in other words the dataset is labeled).

3.1b) Regression vs Classification (2 Punkte)

Supervised learning is typically divided into regression and classification tasks. Briefly explain what are the differences between regression and classification.

Lösungsvorschlag:

The main difference between them is that the output variable in regression is numerical (or continuous) while that for classification is categorical (or discrete).[2]

* Regression is the task of predicting a continuous quantity.[3]
* Classification is the task of predicting a discrete class label.[3]

Robot Learning Homework 3

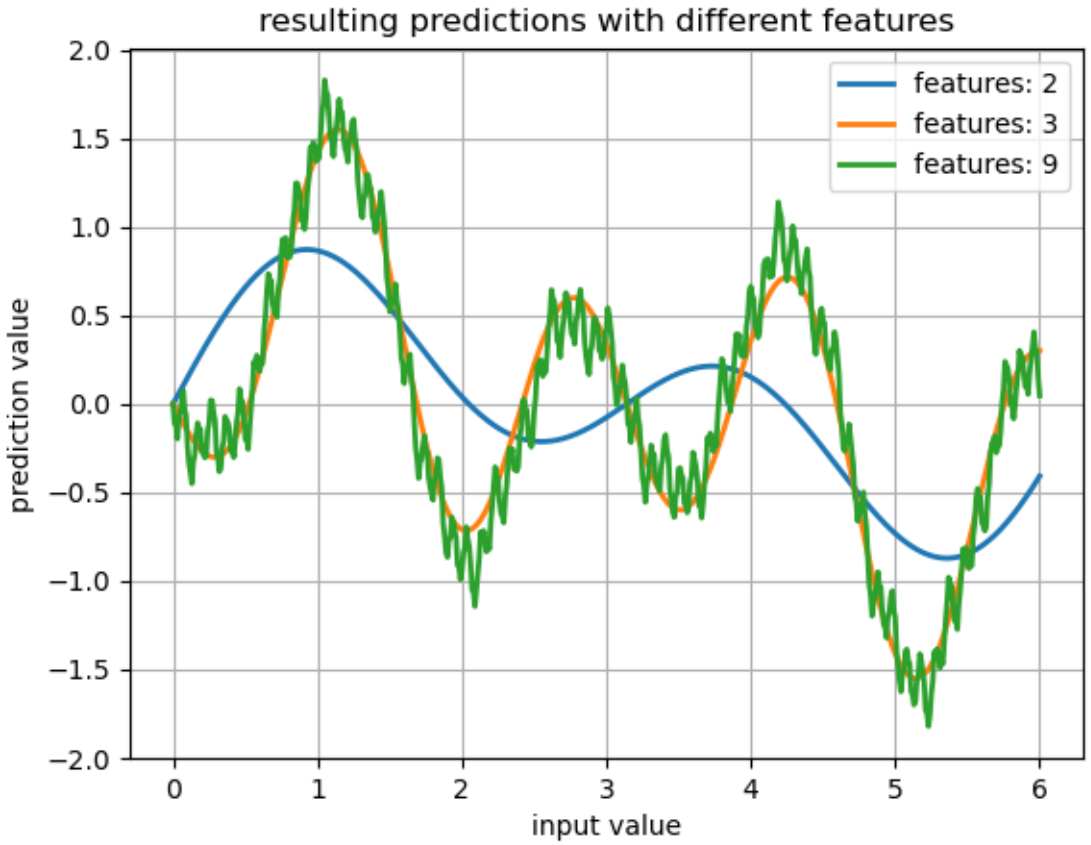
Group 200:

Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Liaotian	Zhihao	2897965

3.1c) Linear Least Squares (4 Punkte)

Consider the training set above to calculate features $\phi(x)$ of the form $\left[\sin\left(2^i x\right)\right]_{i=0\ldots n-1}$. Compute the feature values when n is 2,3 and 9 (i.e., when using 2,3 and 9 features). Use the linear least squares (LLS) method to predict output values y for input values $x \in \{0, 0.01, 0.02, \ldots, 6\}$ using the different numbers of features. Attach a single plot showing the three resulting predictions when using 2,3 and 9 features (i.e., having x and y as axes)

Lösungsvorschlag:



Robot Learning Homework 3

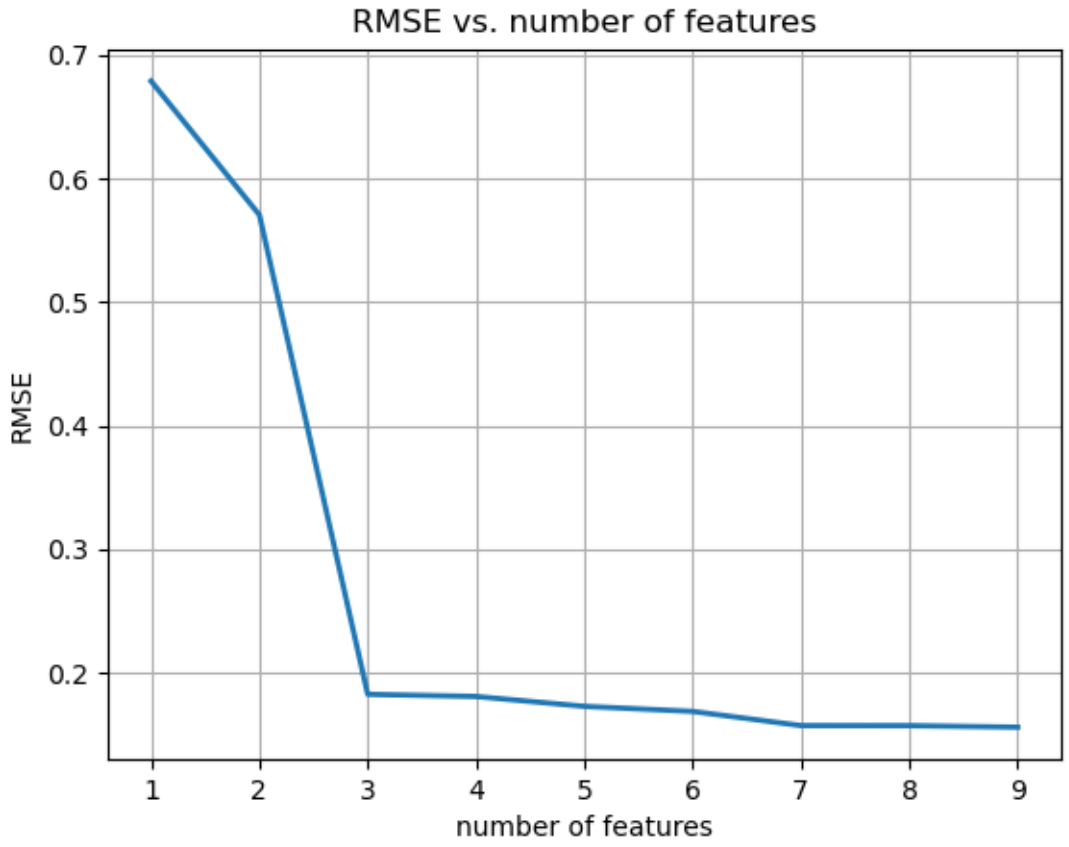
Group 200:

Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Liaotian	Zhihao	2897965

3.1d) Training a Model (2 Punkte)

The root mean square error (RMSE) is defined as RMSE $= \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_i^{\text{true}} - y_i^{\text{predicted}}\right)^2}$, where N is the number of data points. Using the LLS algorithm implemented in the previous exercise, train a different model for each of the number of features between 1 and 9, i.e., $[1, 2, 3 \ldots, 9]$. For each of these models compute the corresponding RMSE for the training set. Attach a plot where the x-axis represents the number of features and the y-axis represents the RMSE.

Lösungsvorschlag:



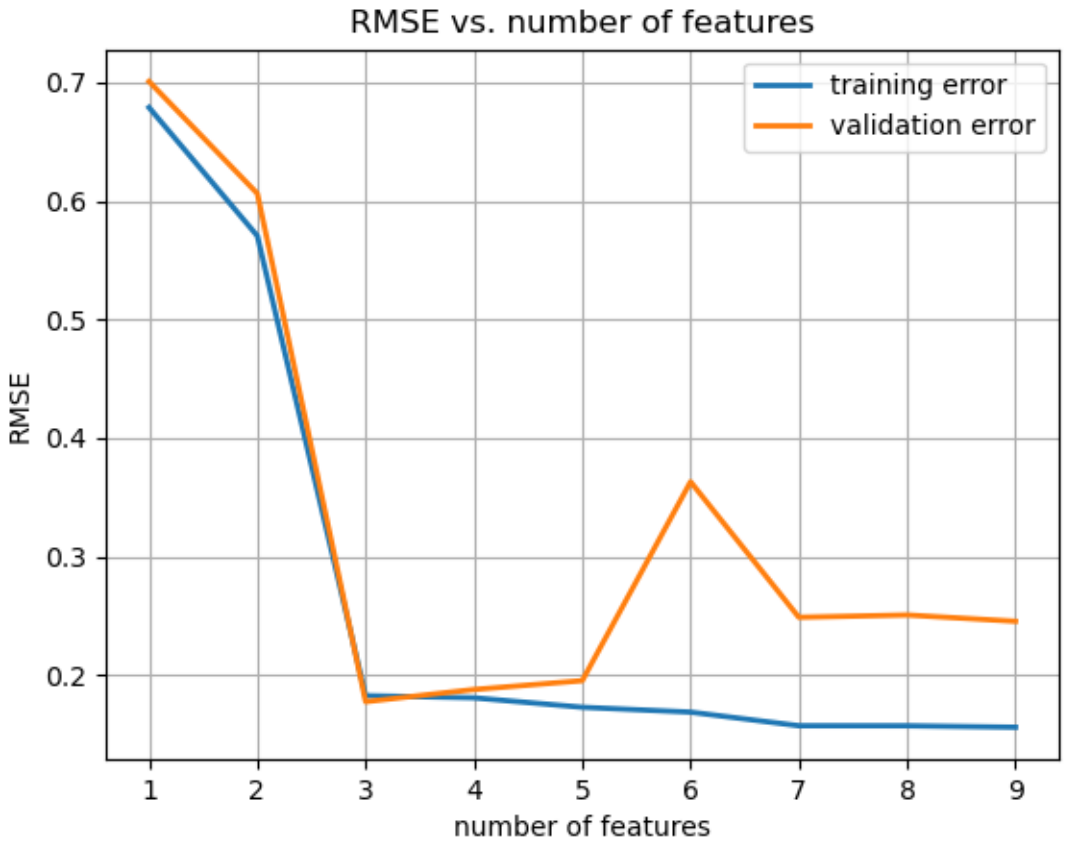
Robot Learning Homework 3 Group 200:

Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Liaotian	Zhihao	2897965

3.1e) Model Selection (4 Punkte)

Using the models trained in the previous exercise, compute the RMSE of each of these models for the validation set. Compare in one plot the RMSE on the training set and on the validation set. How do they differ? Can you explain what is the reason for these differences? (Hint: remember the plot from Exercise c)) What is the number of features that you should use to achieve a proper modeling?

Lösungsvorschlag:



Difference of RMSE:

* training set:
  With increasing of features the RMSE on training set reduce significantly.

* validation set:
  The RMSE on validation set decreases while the features increases up to 3.
  After that the RMSE will increase again, especially when features more than 5.

The over-fitting in training could be the reason of those differences, which means that: with increasing features' number the trained model will be more and more appropriate with training dataset; however this fitting could include too many details in training dataset, which is not suitable for validation dataset and brings the increasing of RMSE in validation phase.
In our opinion the appropriate should be 3.

Robot Learning Homework 3 Group 200:

Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Liaotian	Zhihao	2897965

3.1f) Cross Validation (8 Punkte)

K-fold cross validation is a common approach to estimate the test error when the dataset is small. The idea is to randomly divide the training set into K different datasets. Each of these datasets is then used as validation set for the model trained from the remaining $K-1$ datasets. The resulting vector of errors $E = [e_1 \dots e_K]$ can now be used to compute a distribution (typically by fitting a Gaussian distribution). When K is equal to the number of data points, K -fold cross validation takes the name of leave-one-out cross validation (LOO).
Apply LOO using only the training set and compute the mean/variance of the RMSE for the learned models. Repeat for the models with the number of features between 1 and 9, i.e., $[1, 2, 3 \dots, 9]$
Attach a plot showing the mean/variance (as a distribution) of the RMSE computed using LOO and having on the x-axis the number of features and on the y-axis the RMSE. Which is the optimal number of features now? Discuss the results obtained and compare against model selection using train/validation set.

Lösungsvorschlag:

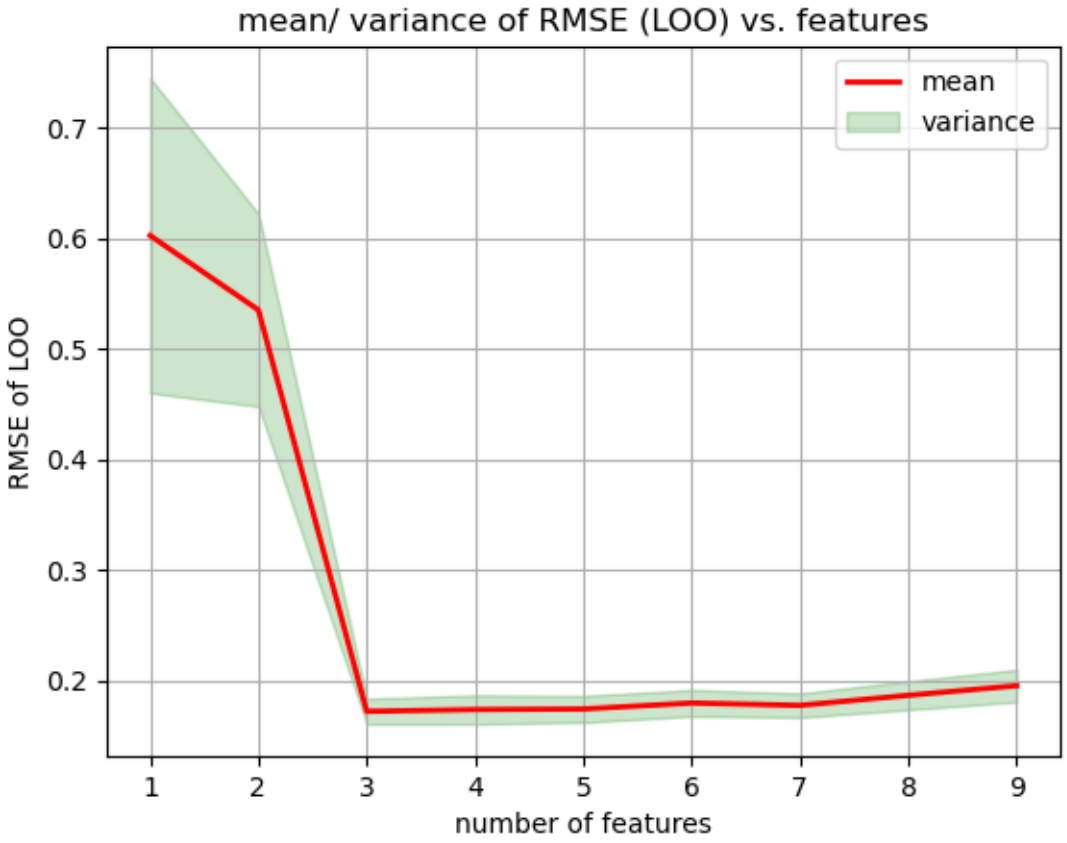


Abbildung 1: Cross Validation in train set

From cross validation in train set, the optimal number of features is 3.

The optimal feature number is identical with model selection result. In cross validation result, the increasing of RMSE (LOO) with features number more than 3 is significant, which is relevant to validation RMSE of model selection result.

In summary the optimal number of features in this model should be 3.

Robot Learning Homework 3 Group 200:

Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Liaotian	Zhihao	2897965

3.1g) Kernel Functions (2 Punkte)

A kernel function $k(\mathrm{x}_i, \mathrm{x}_j)$ is given by the inner product of two feature vectors. Write out the kernel function for the previous set of features where $n = 3$.

Lösungsvorschlag:

When $n = 3$, the feature vector of x_i is: $\phi(\mathrm{x}_i) = \begin{pmatrix} \sin\left(2^0 \mathrm{x}_i\right) \\ \sin\left(2^1 \mathrm{x}_i\right) \\ \sin\left(2^2 \mathrm{x}_i\right) \end{pmatrix} = \begin{pmatrix} \sin\left(\mathrm{x}_i\right) \\ \sin\left(2\mathrm{x}_i\right) \\ \sin\left(4\mathrm{x}_i\right) \end{pmatrix}$

The kernel function is:

$$\begin{aligned} k(\mathrm{x}_i, \mathrm{x}_j) = & < \phi(\mathrm{x}_i), \phi(\mathrm{x}_i) > \\ = & \sin(\mathrm{x}_i) * \sin(\mathrm{x}_j) + \sin(2\mathrm{x}_i) * \sin(2\mathrm{x}_j) + \sin(4\mathrm{x}_i) * \sin(4\mathrm{x}_j) \end{aligned}$$

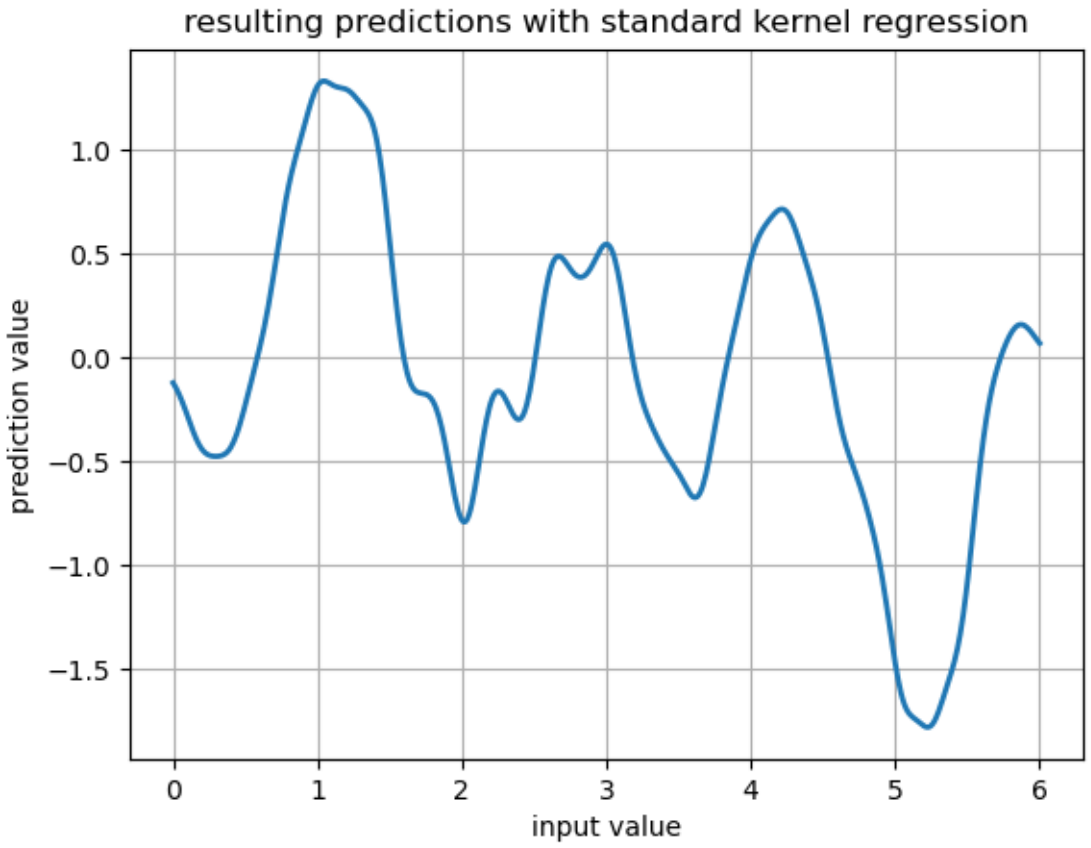
Robot Learning Homework 3 Group 200:

Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Liaotian	Zhihao	2897965

3.1h) Kernel Regression (6 Punkte)

The kernel function in the previous question required explicit definition of the type and number of features, which is often difficult in practice. Instead, we can use a kernel that defines an inner product in a (possibly infinite dimensional) feature space.
Using the training set and an exponential squared kernel $k(x_i, x_j) = \exp\left(-\frac{1}{\sigma^2}\|x_i - x_j\|^2\right)$ with $\sigma = 0.15$ predict output values y for input values $x \in \{0, 0.01, \ldots, 6\}$. Attach a plot of your results.
(Hint: use standard kernel regression: $f(x) = k^\top K^{-1} y$ with $K_{ij} = k(x_i, x_j)$ and $k_i = k(x, x_i)$)
Compute the RMSE on the validation set for the kernel regression model. Compare it with the RMSE of the best LLS model you found.

Lösungsvorschlag:



comparison between kernel regression model and LLS model in validation set:

* RMSE of Kernel Regression Model: 0.2422471046288512
* RMSE of linear least squares (LLS) Model: 0.17748780353358695

Robot Learning Homework 3 Group 200:

Vorname	Name	Matrikel-Nr.
Yi	Cui	2758172
Yuting	Li	2547040
Liaotian	Zhihao	2897965

3.1i) Derivation (5 Punkte)

Explain the concept of ridge regression and why / when it is used. Derive its final equations presented during the lecture.
(Hint: remind that for normal linear regression the cost function is $J = \frac{1}{2}\sum_{i=1}^{N}\left(f\left(x_i\right) - y_i\right)^2\Big)$
(Hint 2 : use matrix notation)

Lösungsvorschlag:

* **Concept of Ridge Regression (or Tikhanov Regularized Regression):**

Ridge Regression is a special Variation of Simple Regression, in which the weighting of input data is no more "unbiased".

With the increasing of input data dimensions, the unbiased weighting of input data (such as Linear Least Squares) will lead to a "curse of model complexity". In other words, due to unavoidable data noise, those unbiased weighting will cause the variance of prediction, especially in validation phase.

An example of unbiased Simple Regression Coefficient(Bayesian Point Estimate) is shown as following:

$$\boldsymbol{\theta}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} J_{\text{MAP}}(\boldsymbol{\theta}) = \left(\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \sigma^2\mathbf{W}^{-1}\right)^{-1}\boldsymbol{\Phi}^\top\mathbf{y} \tag{1}$$

Instead of the measured variance of input data $\sigma^2 W^{-1}$, a simplified regularization coefficient λ is imported in Ridge Regression. The purpose is to reduce the data-noise caused variance in prediction (it is helpful to avoid the over-fitting).

Following is the formulas expression of Ridge Regression Coefficient:

$$\boldsymbol{\theta}_{\text{RidgeRegression}} = \mathbf{N}\left(\mathbf{N}\Phi^\top\Phi\mathbf{N} + \lambda\mathbf{I}\right)^{-1}\mathbf{N}\Phi^\top\mathbf{y} \tag{2}$$

where

$$\mathbf{N}_{ii} = \begin{cases} 1 & \text{if } \operatorname{var}\phi_i(\mathbf{x}) = 0 \\ \frac{1}{\sqrt{\operatorname{var}\phi_i(\mathbf{x})}} & \text{otherwise} \end{cases}$$

and for off-diagonal elements $\mathbf{N}_{ij} = 0$ for $j \neq i$

* **why / when it is used:**

As mentioned above, the measured data noise is unavoidable, which could decrease the accuracy of the model (cause of possible huge variance in prediction).
Moreover, the noise in high-dimensional data will enhance the difficulty in model fitting and increase the model complexity.

In practical scenario, to reduce the effect of data noise, the ridge regression is recommended to implement.

* **Derive its final equations:**

The cost function of Maximum A Posteriori in Bayesian Point Estimate is written as below:

$$\begin{aligned} J_{\text{MAP}}(\boldsymbol{\theta}) &\propto \frac{1}{2\sigma^2}\sum_{i=1}^{N}\left(y_i - \boldsymbol{\Phi}^\top(x)\theta\right)^2 + \frac{1}{2}\boldsymbol{\theta}^\top \boldsymbol{W}^{-1}\boldsymbol{\theta} \\ &= \frac{1}{2\sigma^2}(\boldsymbol{y}-\boldsymbol{\Phi\theta})^\top(\mathbf{y}-\boldsymbol{\Phi\theta}) + \frac{1}{2}\boldsymbol{\theta}^\top \boldsymbol{W}^{-1}\boldsymbol{\theta} \end{aligned}$$

s.t. 1.st derivation of $J_{\text{MAP}}(\boldsymbol{\theta})$ identical with 0:

$$\begin{aligned} \frac{\partial J_{\text{MAP}}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} &= -\frac{2}{2\sigma^2}\boldsymbol{\Phi}^\top(\boldsymbol{y}-\boldsymbol{\Phi\theta}) + \frac{1}{2}\left(\boldsymbol{W}^{-1} + (\boldsymbol{W}^{-1})^\top\right)\boldsymbol{\theta} \\ &= \frac{1}{\sigma^2}\boldsymbol{\Phi}^\top(\boldsymbol{\Phi\theta}-\boldsymbol{y}) + \boldsymbol{W}^{-1}\boldsymbol{\theta} \\ &\quad (\boldsymbol{W}^{-1} = (\boldsymbol{W}^{-1})^\top \text{covariance matrix is diagonal}) \\ &= \frac{1}{\sigma^2}\left(\boldsymbol{\Phi}^\top\boldsymbol{\Phi\theta} + \sigma^2\boldsymbol{W}^{-1}\right)\boldsymbol{\theta} - \frac{1}{\sigma^2}\boldsymbol{\Phi}^\top\boldsymbol{y} \\ &= 0 \end{aligned}$$

$\Rightarrow$

$$\left(\boldsymbol{\Phi}^\top\boldsymbol{\Phi\theta} + \sigma^2\boldsymbol{W}^{-1}\right)\boldsymbol{\theta} = \boldsymbol{\Phi}^\top\boldsymbol{y}$$

we can obtain the optimal coefficient $\boldsymbol{\theta}_{\text{MAP}}$ of $J_{\text{MAP}}(\boldsymbol{\theta})$:

$$\begin{aligned} \boldsymbol{\theta}_{\text{MAP}} &= \arg\max_{\boldsymbol{\theta}} J_{\text{MAP}}(\boldsymbol{\theta}) \\ &= \left(\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \sigma^2\mathbf{W}^{-1}\right)^{-1}\boldsymbol{\Phi}^\top\mathbf{y} \end{aligned}$$

To get the ridge regression coefficient, here $\sigma^2\mathbf{W}^{-1}$ need to be replaced by $\lambda\mathbf{I}$:

$$\boldsymbol{\theta}_{\text{RR}} = \left(\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \lambda\mathbf{I}\right)^{-1}\boldsymbol{\Phi}^\top\mathbf{y}$$

Literatur

[1] ISHA SALIAN, August 2, 2018, SuperVize Me: What's the Difference Between Supervised, Unsupervised, Semi-Supervised and Reinforcement Learning?

[2] Dr. Michael J. Garbade, August 11, 2018, Regression Versus Classification Machine Learning: What's the Difference?

[3] Jason Brownlee, December 11, 2017, Difference Between Classification and Regression in Machine Learning