1 Linear Regression models in 1D

1.1 Conceptual Questions

1.1.1

We are trying to learn an arbitrary function $f := \mathbf{x} \rightarrow y$, with some observed data pairs $\mathcal{D} : \{\mathbf{x}_i, y_i\}_{i=1}^{N}$. Note that this observation might be noisy as well $y = f(\mathbf{x}) + \varepsilon$, for Gaussian noise model $\varepsilon \sim \mathcal{N}(0, \sigma^2)$.

Our aim is to learn f via a model in the form,

$$y = \mathbf{w}^{\mathrm{T}}\mathbf{x} \tag{1}$$

where, $y \in \mathbb{R}$, $\mathbf{w} \in \mathbb{R}^D$ and $\mathbf{x} \in \mathbb{R}^D$. As can be seen the assumed model can only find the linear relation ships between $\mathbf{x}$ and y. In order to, find nonlinear relationships it is a good idea to include certain basis functions. To another high dimensional space where there exist linear relationships. Note that this mapping to another space does not change the linearity of the assumed model with respect to $\mathbf{w}$, which makes life easier for us. Then for M basis functions $\boldsymbol{\phi} := (\phi_0, \cdots, \phi_{M-1})^{\mathrm{T}}$. The assumed model takes the form,

$$y = \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}). \tag{2}$$

Now our model is capable of dealing with nonlinear relationship as well. Again, we have assumed that we have have seen the $\mathcal{D}$ and we would like to learn somehow $\mathbf{w}$ which will give us the $\hat{f}$, which is our estimate of the f that we are trying to find.

In a Bayesian setting, one would assume a prior over the parameters of the model. In this case the parameters of our model is $\mathbf{w}$ and the first thing a Bayesian cult follower would do is to assume a prior over the weights. But, the way to select a prior is tricky for obtaining analytical solutions for the derivations. For our problem statement, the likelihood is given by,

$$p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \beta) = \prod_{i}^{N} \mathcal{N}(y_i|\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x_i}), \beta^{-1}) \tag{3}$$

where $\mathbf{X} := [\mathbf{x_1}^{\mathrm{T}}, \cdots, \mathbf{x_1}^{\mathrm{T}}] \in \mathbb{R}^{N \times D}$, $\mathbf{y} := [y_1, \cdots, y_N]^{\mathrm{T}} \in \mathbb{R}$ and $\beta^{-1} = \sigma^2$. For simplicity we can assume that we know β from now on. One can see that the likelihood is actually Gaussian and the conjugate prior for the Gaussian likelihood is again a Gaussian distribution. So, we will choose our prior as $p(\mathbf{w}) \sim \mathcal{N}(\mathbf{m}_0, \mathbf{S}_0)$ with $\mathbf{m}_0 \in \mathbb{R}^D$, and $\mathbf{S}_0 \in R^{D \times D}$. Knowing that the posterior is given by the product of likelihood and the prior we can "easily" (just kidding it is not that easy, just check the Chapter 2 of Bishop, where the conditional and marginal Gaussians are explained) obtain the posterior for our parameters $\mathbf{w}$ as,

$$p(\mathbf{w}|\mathbf{y}) \sim \mathcal{N}(\mathbf{m}_N, \mathbf{S}_N) \tag{4}$$

where, $\mathbf{m}_N = \mathbf{S}_N(\mathbf{S}_0^{-1}\mathbf{m}_0 + \beta\mathbf{\Phi}\mathbf{y})$, and $\mathbf{S}_N = (\mathbf{S}_0^{-1} + \beta\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi})^{-1}$. I know we have assumed a lot of stuff, but just assume for the sake of "simplicity" (it will never be simple, but anyway...) our prior has the variance as a spherical covariance $\mathbf{S}_0 = \alpha^{-1}\mathbf{I}$, with $\mathbf{I} \in \mathbb{R}^{D \times D}$.

Then, remembering that the marginalization of the parameters of the model $\mathbf{w}$ gives the predictive distribution.

$$p(y|\alpha, \beta, \mathbf{y}) = \int p(y|\mathbf{w}, \beta) p(\mathbf{w}|\mathbf{y}, \alpha, \beta) d\mathbf{w} \tag{5}$$

Then one can "easily" (again, it is not that easy just check out the same place in Chapter 2) show that the predictive distribution is given by,

$$p(y|\mathcal{D}, \alpha, \beta) = \mathcal{N}(y|\mathbf{m}_N^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}, \sigma_N^2(\mathbf{x})) \tag{6}$$

where, $\sigma_N^2(\mathbf{x}) = 1/\beta + \boldsymbol{\phi}(x)^T \mathbf{S}_N \boldsymbol{\phi}(\mathbf{x})$.

Okay Okay, the long long long derivation can be found on the internet too but the gist of it is this and anyone getting this idea this far has pretty good understanding of the concepts that constitutes the foundation of Bayesian Linear Regression.

Now, one can observe that the relationship between LS (Least-Squares) and BR (Bayesian Regression) is not that obvious. We utilize the Bayes rule to come up with the Bayes optimal parameters for BR, whereas in the LS we are trying to minimize the residual of our assumed model for a given $\mathcal{D}$. Note that, due to the obtained posterior being a Gaussian Distribution (our mean is exactly our mode in this case) the maximum posterior is nothing but our $\mathbf{m}_N$ vector. Now, if we assume (again assumption on top of another assumption, but promise this is the last one...) $\alpha \to 0$ for $\mathbf{S}_N$ we end up with the $\mathbf{m}_N$ being your maximum likelihood estimate $((\mathbf{\Phi}^T \mathbf{\Phi})^{-1} \mathbf{\Phi}^{\mathrm{T}} \mathbf{y})$. (You can see this from the definitions of $\mathbf{m_N}$ and $\mathbf{S_N}$.) Keeping this in mind, lets remember how do we get the maximum likelihood, estimation. Given the same likelihood term in Equation 3, we need to maximize this expression to get our point estimates of our linear models. Observe that maximization of this term in can be easier if we take the negative log of it and minimize it instead. Remembering the definition of a Gaussian pdf definition and putting the related terms in Equation **??** in place, then taking the negative log, thanks to our likelihood being Gaussian distribution it is "easy" (it is easy too see this time) that the following minimization problem, gives us the maximum likelihood estimation of our parameters $\mathbf{w}$.

$$\arg\min_{\mathbf{w}} = \frac{1}{2} \sum_{i}^{N} (y_i - \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_i))^2 + \text{const.} \tag{7}$$

where, the const. are the trash terms (in terms of minimization problem at hand). Yeayyyy, now we see something that we now from the LS solutions the term $(y_i - \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_i))^2$ is nothing but the thing (sum of the squared residuals) that we are trying to minimize for the LS problem. Hence, we can see that the infinitely wide Gaussian prior that hugs all the parameters in the parameter space gives us nothing but the Least squares solution.

THE END

1.1.2

Generally, if you have enough data you would like to split it to 3 parts as train, validation and test set. On train set you try to condition your modal, in other words learn or train your model parameters. You might have slightly varying models, that is where validation set comes in to the picture (or you might want to tune your models hyper-parameters.) Then, you would like to test your unbiased error which is given by a different set of parameters that you have not showed to you model before hand this set is called as test set.

THE END

1.1.3

- MSE (Mean Squared Error) is an error measure for a given input, target pairs. Considering the Question 1.1, this is represented by $\mathcal{D}_N : \{\mathbf{x}_i, y_i\}_{i=1}^N$ and for a given estimator let's say this is represented by $\hat{f}$ again. Then, the MSE for a dataset Dis defined as follows,

  $$MSE := \frac{1}{N}\sum_i^N(\hat{f}(\mathbf{x}_i) - y_i)^2. \tag{8}$$

  As can be seen this is nothing but the expectation of the squared error at each point that we are looking at.

- R^2 is not an error measure like the MSE. It is simply a statistical measure that tells you how good of a fit you have some might say, but lets not believe them, and go a bit deeper you can find a really good and deep dive in to this magical measure here. Moreover, there are many other definitions, but the most general one is given as;

  $$R^2 := 1 - \frac{\sum_i^N(\hat{f}(\mathbf{x}_i) - y_i)^2}{\sum_i^N(\sum_j^N(y_j) - y_i)^2}. \tag{9}$$

THE END

1.1.4

k-fold cross-validation is a procedure that is employed generally for model selection purposes. Lets say we have observed again $\mathcal{D}_N : \{\mathbf{x}_i, y_i\}_{i=1}^N$. Then, we split this data into k partitions and 1 group is always left-out for testing our trained model $\hat{f}$ and the rest $k-1$ partitions are used for training, this process is repreated k times then looking at the statistical measures obtained from the error measures of every run is compared to select a model. This method allows us to, squeeze every bit of juice from our data, help us select a model,and fine-tune hyper-parameters, which is some sort of model selection as well...

THE END

1.1.5

According to Miguel's definition a hyper-parameter is any parameter that is not being learned in the training procedure and the parameters are the ones that are tuned in the training phase. However, one might come up other definitions and some other people calling what in the lecture we would call parameters hyper-parameters and so on. Thus, it is important to not stick to one formal definition for the literature, but for the sake of completeness in this lecture you can refer to Miguel's definition. My question to you would be is the model itself a hyper-parameter in this case?

THE END

Look at the below link for the answers of the coding questions.

1.2

Remember the ridge regression solution as $(\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi} + \lambda \mathbf{I})^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{y}$. Now, remembering the prior assumption over the parameters defined in the conceptual questions (it was one of the simple assumptions) to be $p(\mathbf{w}) \sim \mathcal{N}(\mathbf{0}, \alpha^{-1}\mathbf{I})$, one can see that the MAP estimate which is given by the $\mathbf{m}_N$ in this case the mean/mode of our posterior for our parameters becomes equal to ridge regression optimal solution for $\lambda = \alpha^{-1}$.

THE END

1.3

A kernel function is the function is a tool that allow us to go to high-dimensional spaces where dot product is defined. In the setting of the GPRs, it determines the relationship between two different data points $\mathbf{x}$ and $\mathbf{x}'$. According to the definitions given in the class the parameters are the parameters of the kernel and the kernel becomes the hyper–parameter in a sense. The additional noise term in the diagonal increases the stability of the covariance matrix as sometimes the covariance matrix gets ill-conditioned and the inversion needed for the posterior predictive cannot be obtained. In addition, to stability it acts as another parameter where we can introduce noise to our GPR model.

THE END