**Machine Learning 2021 Homework 1**

310511048 張奕廷

1. **Bayesian Linear Regression**
   1. and are said to be conditionally independent if

Since is a new test point and label, are the training data and corresponding label, and is only trained on ,we can conclude that and are independent given . Hence, we can obtain:

Furthermore, is only dependent on its training data , hence:

* 1. The equations from textbook page 93 are list below.

From 1.1, we know that,

**First Step**

To derive , we first derive , from hint:

By equation (2.114),

and by equation (2.113), we obtain the prior distribution as follows,

Then, by equation (2.117) and substitute the results above,

Hence, by equation (2.116),

**Second Step**

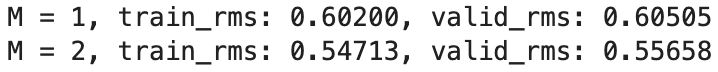
Now we derive . Again, by equation (2.114),

By equation (2.113) and the derivation from first step,

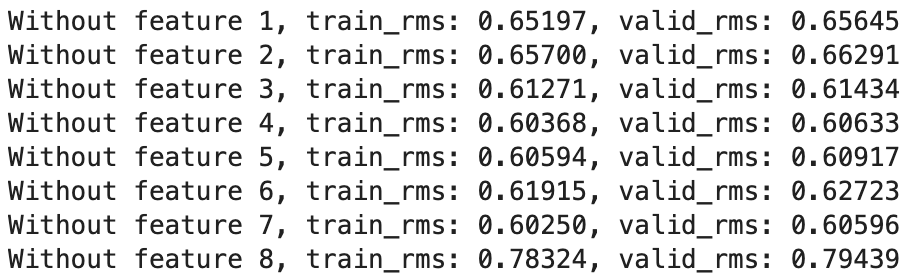
Finally, by equation (2.115) and substitute the results above,

1. **Linear Regression**
   1. Feature Selection

(a) **Code Result.**



(b) **Code Result. Explain.** I remove one feature from the dataset at a time and observe the RMS error. As we can see from the code result, both training and validation RMS error are greatest when feature 8 (*median income*) is removed. As a result, the feature *median income* is the most contributive one in this dataset.

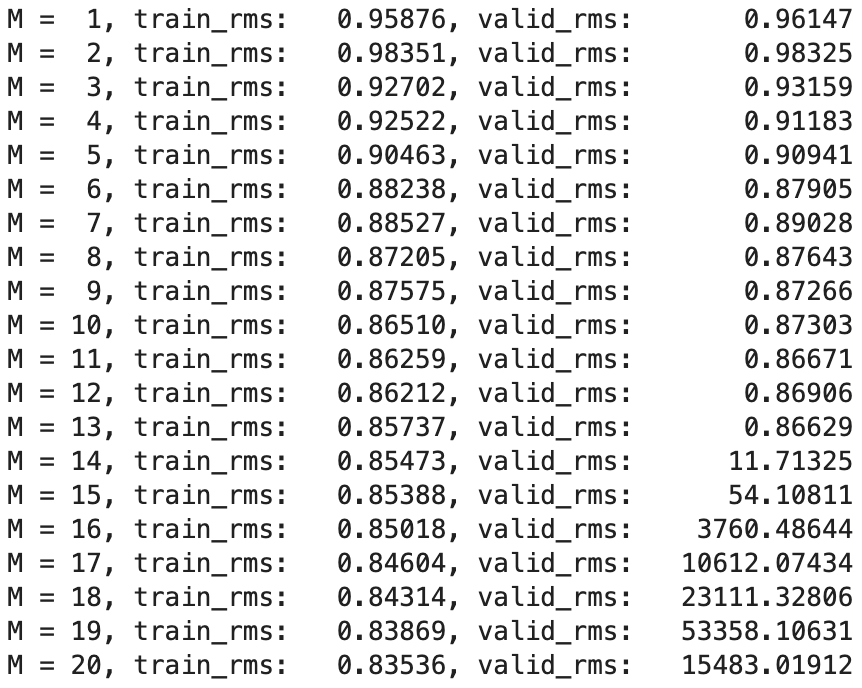


* 1. Maximum Likelihood Approach

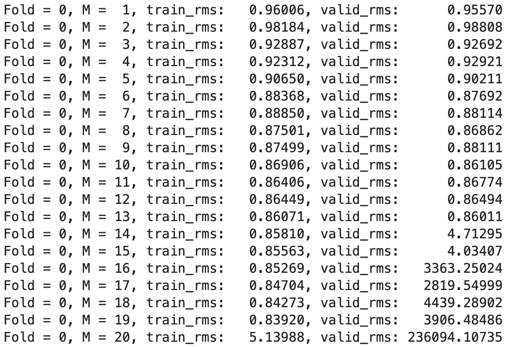
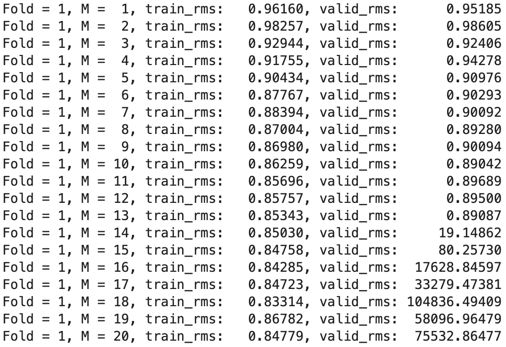
(a) **Explain.** In problem 2.1, I already used polynomial as the basis function for my regression model. However, polynomials are *global* basis functions, each affecting the prediction over the whole input space. *Local* basis functions are often more appropriate, so I choose Gaussian distribution as the basis function.

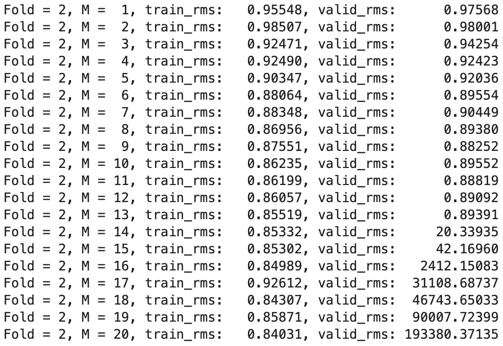
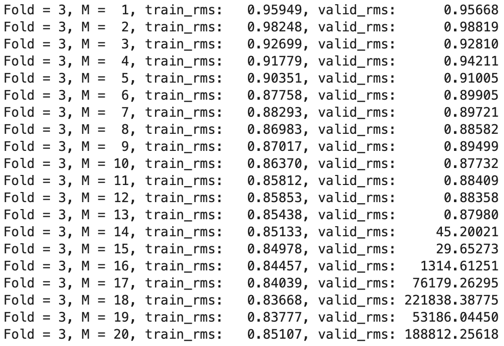
(b) **Code Result. Explain.** Since we’re not required to find the best parameters for basis functions in this homework, I choose the Gaussian distribution as follows,

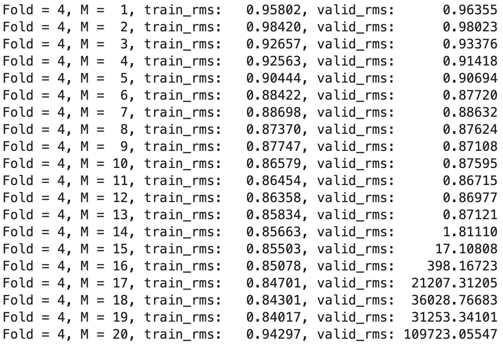
where is the order of basis function and is all positive integers less than . Below is the code result, we can see that as increases, the training error slightly decreases yet the validation error increases significantly after , which means that the model is overfitting. Also, I discover that changing the basis function to Gaussian doesn’t make the RMS error better than in problem 2.1. I think this is because I didn’t choose the best parameters for Gaussian.



(c) **Code Result. Explain.** Code result for N-fold cross validation with N set as 5 are as follows. Obviously, we can see that every fold has similar behavior. When the order is low (), both training and validation RMS error are still high, demonstrating that the model is underfitting, but when the order grows higher (), the error seems to converge. However, the validation RMS error starts to increase significantly when yet the training RMS error keep on decreasing, indicating the model is overfitting.



* 1. Maximum A Posterior Approach

(a) **Explain.**

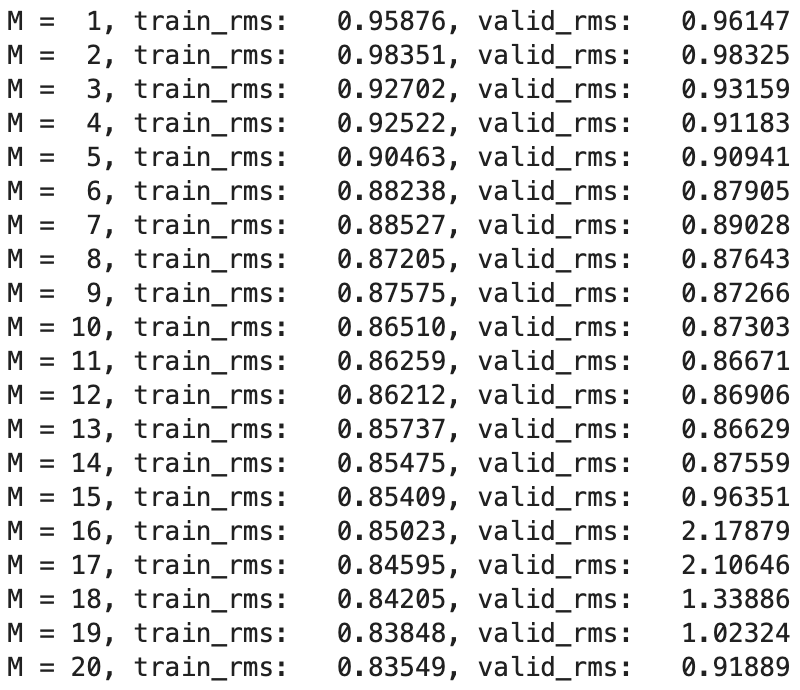
**Maximum Likelihood**

**Maximum a Posterior**

If we choose Gaussian distribution as the prior, then

The difference between *maximum likelihood* and *maximum a posterior approach* is highlighted in red in the above equations, which is the prior distribution. In *maximum likelihood approach*, there is no this term. Also, the final derivation of *maximum likelihood approach* can be viewed as **L**east **S**quares (**LS**), while in *maximum a posterior approach*, if we choose Gaussian distribution as the prior, the final derivation can be viewed as **R**egularized **L**east **S**quares (**RLS**). Due to the regularized term, we can avoid models from overfitting to training data to some extent with *maximum a posterior approach*.

(b) **Code Result.**

****

(c) **Explain.** Compared to the results from *maximum likelihood approach*, applying *maximum a posterior approach* can indeed ease the effect of overfitting. As we can see in the code result above, with the regularized term parameter set as , the validation RMS error is much smaller compared to previous results in problem 2.2.