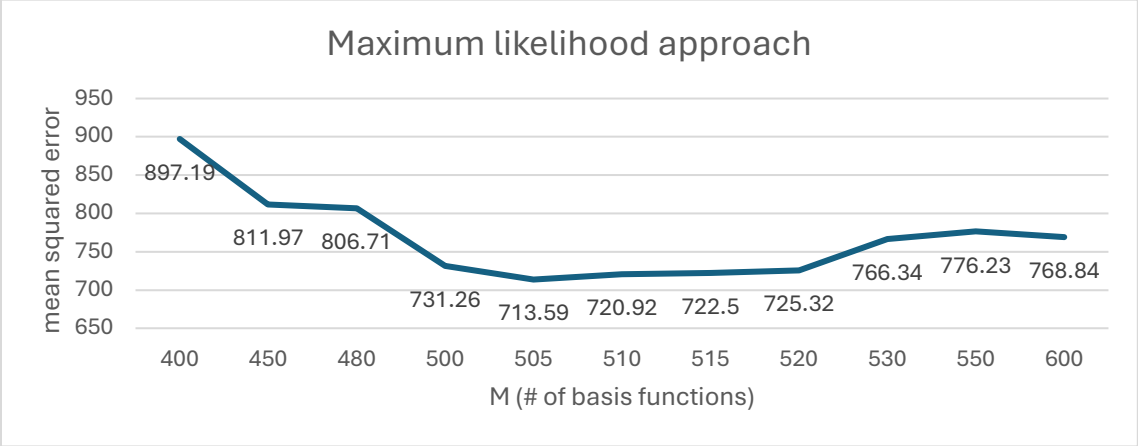
Hw1_111705011 陳信愷

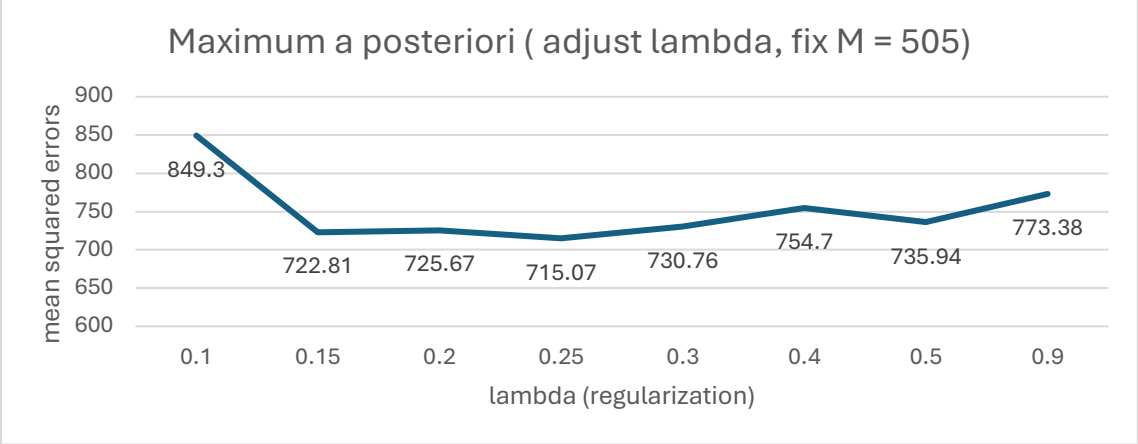
- Maximum likelihood
  - Basis function: Gaussian
  - Likelihood: Gaussian
  - Results (note that the x-axis is not evenly spaced)

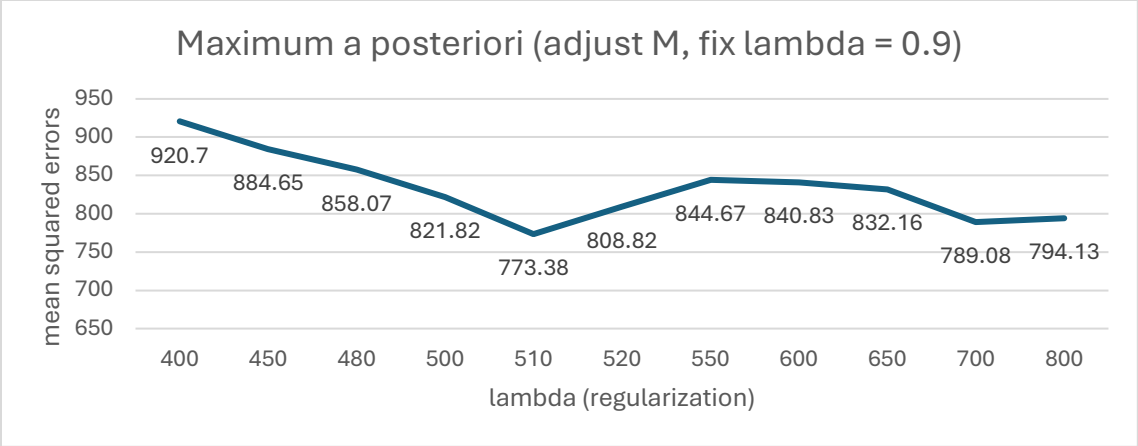


  - Hyperparameters tuning (hyperparameter: M)

    There is a clear trend plotting M. When $M < M_{opt} = 505$, shows underfitting. When $M > M_{opt} = 505$, shows overfitting. It is impossible to calculate $M > 650$ because it shows that $\Phi^{T}\Phi$ is not invertible, which pseudo inverse calculation cannot be done (details written in "Additional details").

- Maximum a posteriori
  - Basis function: Gaussian
  - Prior: Gaussian
  - Likelihood: Gaussian
  - Results (note that x-axis is not evenly spaced)



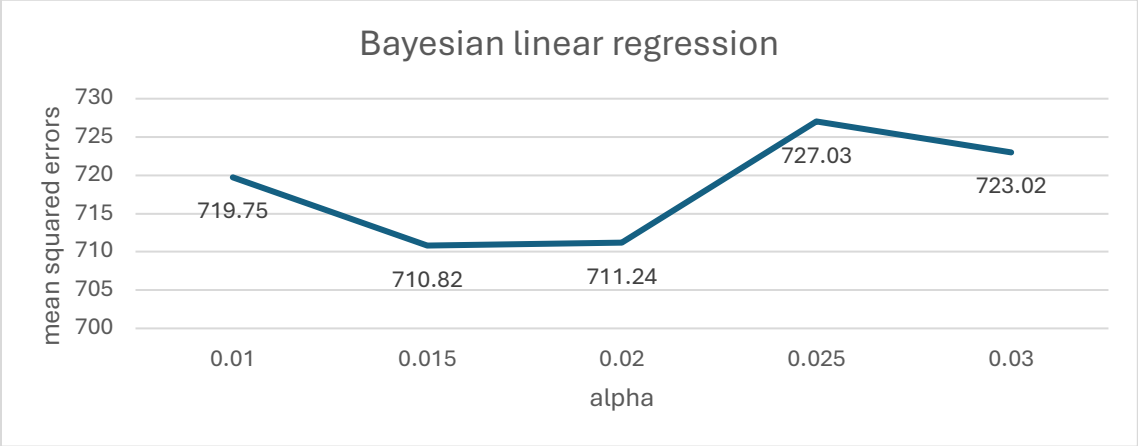


- Hyperparameters tuning (hyperparameter: M, λ)

  M and λ must be tuned together, a better way of tuning M, λ is grid search. However, it takes too long to go through all the tests. So, I take $M_{opt} = 505$ from maximum likelihood approach to tune the $\lambda_{opt} = 0.2$.

  I also arbitrarily choose a $\lambda_0 = 0.9$ and calculate M's. We can see that there exist at least two low points in MSE given λ_0. Model with $M > 650$, however, in maximum likelihood approach shows $\Phi^T\Phi$ is not invertible. With a large M, the model is not robust enough. Since the random seed is not fully fixed, it could lead to error if some random seed causes two centroids to be too close. (I fix random seed number at a function scale, but not for each iteration, details written in "Additional details".). Hence, I pick $M_{opt} = 505, \lambda_{opt} = 0.25$.

- Bayesian linear regression
  - Basis function: Gaussian
  - Prior: Gaussian
  - Likelihood: Gaussian
  - Results



- Hyperparameters tuning (hyperparameter: M, α)

  Like MAP approach, M and α must be tuned together. M and α are even more correlated since the number of clusters M directly changes the variances. So, I fix $M_{opt} = 505$ from maximum likelihood approach to tune the α_{opt}.

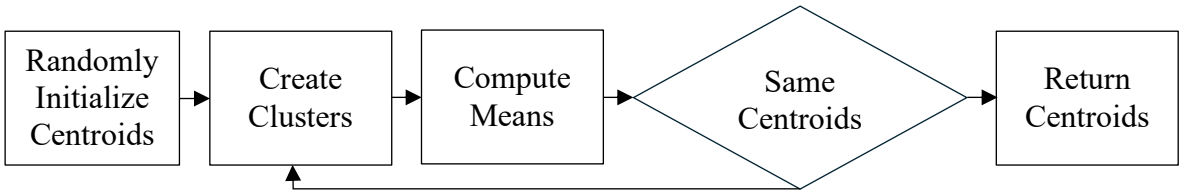
- Additional details
  - K-means to find μ_i, σ_i of clusters for Gaussian basis function

    Since the training dataset contains 6000 (x, y) coordinates, given M basis functions (also M clusters), M must not exceed 6000. From the Performance Score metric given by the TAs, we can conjecture that M is close to the resulting mean squared errors, which is set as lower than 900.

    Due to large M and application of Gaussian basis function, it is impractical to guess μ_i, σ_i and manually adjust them by oneself. Therefore, I use K-means to find μ_i, σ_i, a method of clustering to find centroids μ_i and their shapes σ_i (though not necessarily Gaussian). Using K-means have two main benefits:

    1. Reduce the # of hyperparameters.
    2. Error from large M (An error would show up if two clusters' centroids are too close that a data point cannot decide which cluster to join. This creates an upper bound of M, which is useful.)

    Having a vague upper limit and an approximate range of M helps us tune the hyperparameters. In maximum likelihood approach, M is the only hyperparameter, which can help us tune M well.



Flow chart of K-means implementation

  - Large M causes linearly dependent features

    Calculating the weight involves pseudo inverse computation, where $\Phi^T\Phi$ must be invertible. If M is too large, the column (feature) vectors could become linearly dependent, it happens when two centroids are located too close to each other, the computation would fail and therefore create an upper bound for M.

- Fix random seed

  Even though tuning $M = M_{max}$, which finds every possible cluster, seems to be promising. However, large M in my experiments shows sometimes overfitting (see results above). In my case of using K-means clustering, overfitting happens when a cluster does not include enough data points that predictions become largely biased due to lack of data.

  But if $M \neq M_{max}$, there is a drawback. The random clustering in the first place would have great influence over the result. Hence, I tried to find a good seed number. To improve replication of results, I fixed the seed number of randomly initializing centroids to 44. The choice of seed number is purely by the performance from s-fold cross validation. I tried out some seeds, and 44 was the better one.

  Though this still does not always yield the same result because I do not determine every seed when initializing every centroid, but if to set fixed the seed in every iteration of initialization, the code would be very inflexible.

- Plot out data

  From the plots, we can tell that there are many zero values (perhaps the sea) around southern Taiwan. This encourages me to set the prior mean of Bayesian linear regression model to zeros.

  We also see that there exist many clusters within the graph. Due to the characteristics of altitude, there should not be many outliers, so I use Gaussian likelihood. Within a cluster, there shouldn't be much fluctuation between data points, so I decided to use Gaussian prior. Between Gaussian and polynomial basis function, Gaussian significantly performs better because Gaussian basis function uses more information than polynomial basis function, such as μ_i, σ_i. And it is also less prone to mistakes to implement closed-form solutions.