Homework 1 Part 1

This is an individual assignment.

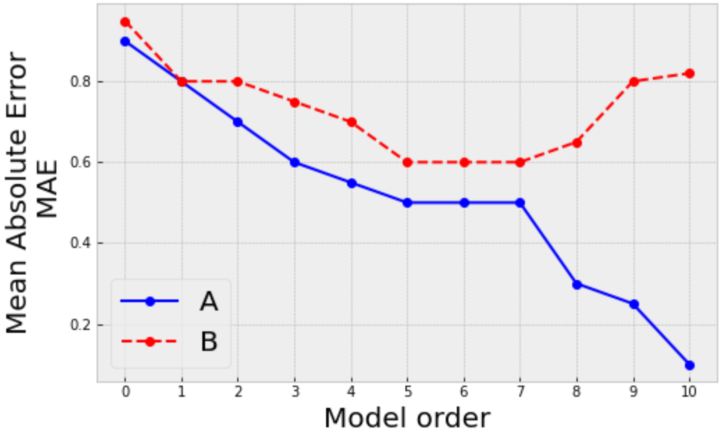
Write your answers as markdown cells.

Exercise 1 (4 points)

The figure below illustrates the hyperparameter tuning performance in two datasets (training and validation) as a function of the model order, M, in the polynomial regression mapper function (or model).

In [1]:
```
from IPython.display import Image
Image(’figures/performance.png’, width=400)
```

Out[1]:



Based on these graph, answer the following questions:

1. (2 points) **Which line (A or B) correspond to the train and validation sets?**
2. (2 points) **Based on these results, which model order M would you select for the final model?**

Justify your answers.

1. Line A corresponds to training set because the value of error reduces towards zero as moder order increases, namely overfitting has happened, which can be verified by the surge of error value on Line B. Therefore, Line B corresponds to testing set.

1. I would go for 5, 6, 7 because we can see the value of error significantly increases when model order is 7 or greater. It indicates that our model is overfitting and we have uncessary complexity.

In [ ]:

Exercise 2 (4 points)

In practice, what strategies can you apply to avoid overfitting? List at least 4 distinct strategies and explain why they are effective at mitigating overfitting.

1. Add proper regularizer to the error function so that we constrain the magnitude of w.
2. Increase the amount of data used for trainning, because poor in data can promote the impact caused by noises.
3. Use cross-validation. When given limited data, it can divide the data in to k subsets and validate the model for k times with different subsets each time.
4. For some simple tasks, we can use simpler models because sometimes over-complicated models only bring unnecessary complexity and thus leads to over-fitting. For instance, we can reduce the order of polynomial.

In [ ]:

In [ ]:

Exercise 3 (2 points)

In practice, how can you determine whether you have overfitted your machine learning system?

We can determine it by looking up graph that shows error of both trainning set and testing set. Abstractly speaking, when the the value of error for trainning set closes to zero while the value for testing set becomes very large, we consider that over-fitting has happened. However, it depends on requirement of different tasks.

In [ ]:

In [ ]:

Exercise 4 (4 points)

Suppose you have 100 training samples that you are using to train a classifier to distinguish between four classes. The training data has 50 samples of class 1, 25 samples of class 2, 20 samples of class 3 and 5 samples of class 4. To evaluate the stability and performance of your classifier on each class, you use 10-fold cross-validation. Is it a good strategy to randomly partition the data into 10 folds? Why or why not? If yes, fully justify why. If no, state why not, provide an alternate cross-validation scheme and justify the new scheme.

It's not reasonable to randomly partition the data into 10 folds. We may lose some part of data(say class 4 which has minor samples). In my opinion, we should take advantage of stratification, which is in this case spliting the dataset into 5 classes and each one of them has 10 samples of class 1, 5 samples of class 2, 4 samples of class 3 and 1 sample of class 4. In this way, classes are represented in the right proportions when subsets of data are held out, not to disturb the class prior probabilities.(Alpaydin, Ethem. Introduction to Machine Learning, MIT Press, 2014. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/ufl/detail.action?docID=3339851.)

In [ ]:

In [ ]:

Exercise 5 (2 points)

Suppose that you split your data into training and test sets, and perform cross-validation with the training set to select the best set of parameters for the model. Can the model still overfit? Justify your answer.

Yes, it's still possible. First, it is possible that we don't have enough data in the first place so our models are greatly impacted by noises and get overfit. Second, if we pick model of imporper complexity for our task, model can be easily overfit.

In [ ]:

In [ ]:

Exercise 6 (4 points)

Answer the following questions:

1. (2 points) **Name one advantage and one disadvantage of using regularization.**

2. (2 points) **In practice, how criteria would you use to decide between ridge, lasso and elastic net regularization?**

1. Advantage: using regularization helps us avoid overfitting because it adds a penalty term to the loss function, discouraging the model from fitting the training data too closely. Disadvantage: if we aggressively use regularization, it may lead to the loss of complexity of model, making the data underfit to training data.

1. I would go for ridge as default, but when training data is limited, it is better to choose lasso regularization because they can reduce some features' weight to zero. In addition, when we have data with many features, it is better to use elastic net regularization becuase it combines ridge and lasso regularization, allowing us to benefit from feature selection while handling multicollinearity.

In [ ]:

```
```

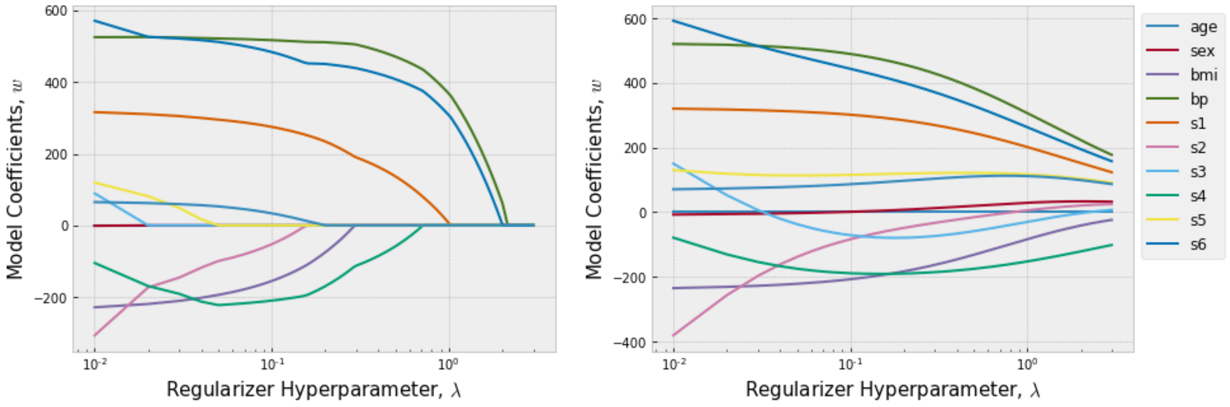
Exercise 7 (4 points)

The figure below, shows how the weights associated with each with the 10 attributes/features change as a function of the regularizer parameter λ in a linear regression model.

In [2]:

```
Image('figures/Regression_with_Regularizer.png', width=900)
```

Out[2]:



Based on this plots, which one (left or right) corresponds to the Ridge Regression and Lasso Regression? Justify your answer.

The left one shows lasso regularization because all the these features' weights reduced down to zero as lamda increases.
The right one shows ridge rigularization because the features' weights are continuously shrunk towards zero as the regularization strength increases, but they are never exactly zero.

In [ ]:

In [ ]:

Exercise 8 (4 points)

In practice, when you are implementing your regression or classification tasks with your feature matrix $\mathbf{X}$ of size $N \times M$, N is the number of training samples and M is the number of dimensions/features.

If you encounter the computational error "matrix is singular":

1. (2 points) **What does this mean about the feature matrix $\mathbf{X}$?**
2. (2 points) **What should you do to solve the problem?**

In [ ]:

In [ ]:

In [ ]:

Exercise 9 (4 points)

Before feeding the data to a mapper function, we must carry any necessary preprocessing. This may include encoding features, dealing with missing values, and scaling. In which order should the data be processed:

- (2 points) **Option 1: Partition the data into training-test sets, then apply preprocessing based on training set.**

- (2 points) **Option 2: Apply preprocessing on entire data, then partition into training-test sets.**

I will go for option 1 because we can ensure our preprocessing steps are adapted to the distribution of training data. We then can use the same steps to preprocess test data, helping the data to be consistently processed.

If we choose option 2, we risk data leakage from the test set into the training set. In this way, we are very likely to be overly optimistic to testing consequence.

```
In [ ]:
```

Exercise 10 (9 points)

Suppose that you are training a linear polynomial regression model of order M for a training set with N data points $\{x_i\}_{i=1}^N$, where $x_i \in \mathbb{R}$, and its corresponding target labels $\{t_i\}_{i=1}^N$. Answer the following questions:

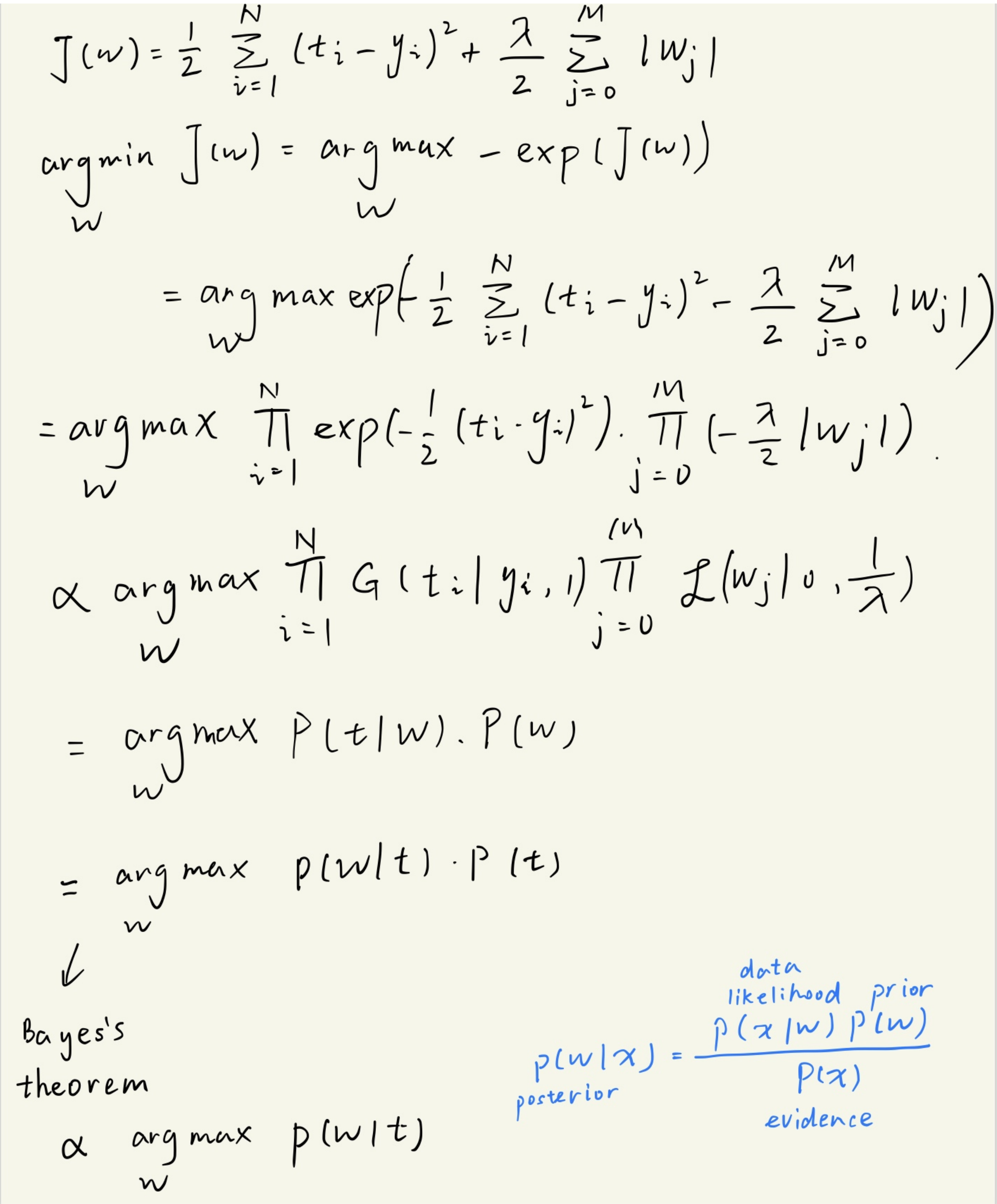
1. (3 points) **Write down the mapper function. Use proper notation to avoid ambiguity.**
2. (3 points) **Suppose you want to minimize the absolute error with the Lasso regularizer. Write down the objective function.**
3. (3 points) **What is the Bayesian interpretation of this objective function? Show your work using parametric expressions and justify it further using words.**

1.
$$y(x, w) = \sum_{j=0}^{M} w_j x^j$$

   Here, w is the vetcor of parameters, and $y(x, w)$ is the predicted value for input x using the polynomial regression model of order M.

1.
$$J(w) = \frac{1}{2}\sum_{i=1}^{N}(t_i - y(x_i, w))^2 + \frac{\lambda}{2}\sum_{j=0}^{M}|w_j|$$

   Here, x_i and t_i respectively means the input and target value of data point i

1. 

$$J(w) = \frac{1}{2}\sum_{i=1}^{N}(t_i - y_i)^2 + \frac{\lambda}{2}\sum_{j=0}^{M}|w_j|$$

$$\underset{w}{\arg\min}\, J(w) = \underset{w}{\arg\max} - \exp(J(w))$$

$$= \underset{w}{\arg\max}\, \exp\left(-\frac{1}{2}\sum_{i=1}^{N}(t_i - y_i)^2 - \frac{\lambda}{2}\sum_{j=0}^{M}|w_j|\right)$$

$$= \underset{w}{\arg\max} \prod_{i=1}^{N} \exp\left(-\frac{1}{2}(t_i - y_i)^2\right) \cdot \prod_{j=0}^{M}\left(-\frac{\lambda}{2}|w_j|\right).$$

$$\propto \underset{w}{\arg\max} \prod_{i=1}^{N} G(t_i | y_i, 1) \prod_{j=0}^{M} \mathcal{L}\left(w_j | 0, \frac{1}{\lambda}\right)$$

$$= \underset{w}{\arg\max}\, P(t|W) \cdot P(w)$$

$$= \underset{w}{\arg\max}\, p(w|t) \cdot p(t)$$

↙

Bayes's theorem

$$\propto \underset{w}{\arg\max}\, p(w|t)$$

$$\underset{\text{posterior}}{p(w|x)} = \frac{\overset{\text{data likelihood}}{p(x|w)}\ \overset{\text{prior}}{p(w)}}{\underset{\text{evidence}}{p(x)}}$$

Exercise 11 (4 points)

Answer the following questions:

1. (1 point) **What is the *Bayesian interpretation* of a constrained objective function, i.e. objective function with a regularizer?**
2. (1 point) **Why is it useful in machine learning?**
3. (1 point) **What advantages, if any, does it bring when performing parameter estimation?**

4. (1 point) **Suppose you used the Bayesian approach to estimate a data likelihood. Provide a practical example on how you can use that data likelihood estimation.**

1. It treats the model parameters as random variables and applying Bayesian principles to incorporate prior knowledge and uncertainty into the model.

1. Sometimes we have uncertain event, for example whether there once have lives on Pluto, which cannot be repeated through experiment to define a probablility. In this cases we can use Bayesian interpretation, allowing us to incorporate prior knowledge to solve the problem.

1. As my answer in Q2, it can incorporate prior belief into parameter estimation. In addition, Bayesian parameter estimation provides a reasonable approach to quantify uncertainty in parameter estimates.

1. We can use it to estimate the efficacy of a drug.First, we can express our belief about the efficacy of a drug as a prior distribution, which is based on previous research or the insights of domain experts. We then conduct clinical trials to observe the patient's treatment outcomes, such as success or failure.

On-Time (5 points)

Submit your assignment before the deadline.

Submit Your Solution

Confirm that you've successfully completed the assignment.

Along with the Notebook, include a PDF of the notebook with your solutions.

`add` and `commit` the final version of your work, and `push` your code to your GitHub repository.

Submit the URL of your GitHub Repository as your assignment submission on Canvas.
