Machine Learning

Homework 1

309551059 羅文慧

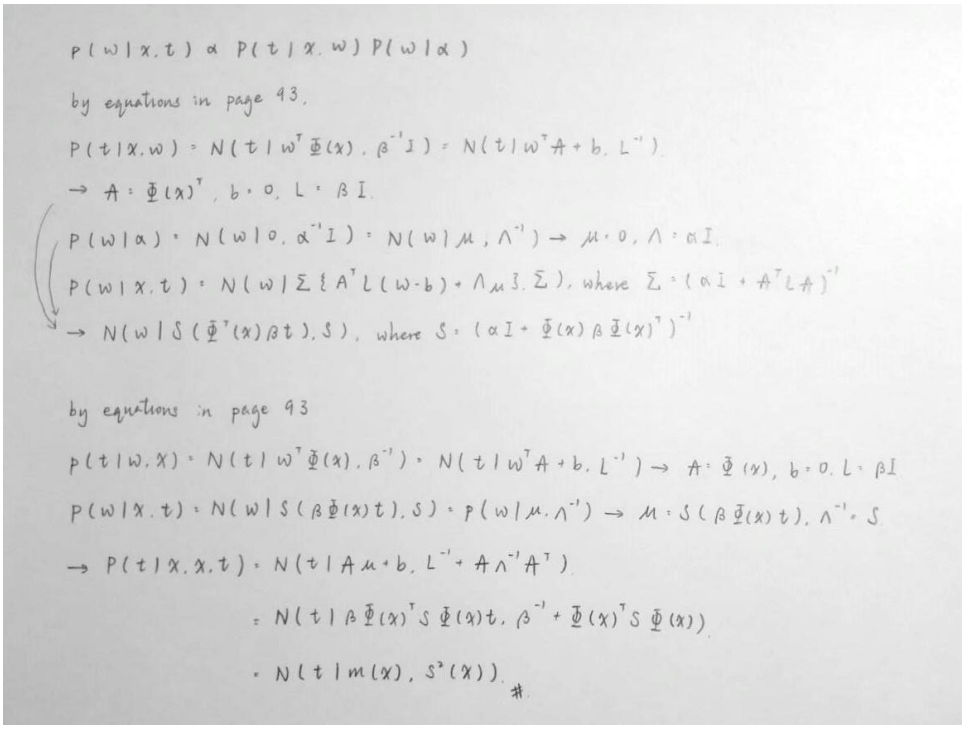
- **Bayesian Linear Regression**

1. **Why we need the basis function φ(x) for linear regression? And what is the benefit for applying basis function over linear regression?**

   Given a problem of using input x to predict a continuous target y, we could use linear regression to solve this problem. But if we know the relationship between x and y is non-linear, and not sure what form of this relationship it is. We could tackle this by using linear basis function models, which ***assure that the target is a linear combination of a set of basis functions****.*

   The benefit for applying basis function over linear regression is it effectively takes our one-dimensional x values and projects them into a higher dimension, so that a linear fit can fit more complicated relationships between x and y.

2. **Prove that the predictive distribution just mentioned is the same with the form $\mathbf{p(t|x}, \boldsymbol{x}, \boldsymbol{t}) = \boldsymbol{\mathcal{N}}(\mathbf{t|m(x)}, \mathbf{s^2(x)})$**



3. **Could we use linear regression function for classification? Why or why not? Explain it!**

   No, the objective of a linear regression model is to minimize the distance between the predicted value and actual value. If there exists imbalanced data, the best fit line may sensitive to these imbalanced data. Thus, it may lead to misclassifications.

- ## Linear Regression

1. **(a) In the feature selection stage, please apply polynomials of order M = 1 and M = 2 over the dimension D = 7 input data. Please evaluate the corresponding RMS error on the training set and valid set. Code Result**

```
M =  1 , training error: 0.0005140167677501864 , validation error: 0.0023106298863022484
M =  2 , training error: 0.00022176323538085685 , validation error: 0.002483574990422217
```

   (b) How will you analysis the weights of polynomial model M = 1 and select the most contributive feature? Code Result, Explain

```
Index(['GRE_score', 'TOFEL_score', 'University_rating', 'SOP', 'LOR ', 'CGPA',
       'Research'],
      dtype='object')
[[-0.20560671]
 [ 0.11444135]
 [ 0.21293897]
 [ 0.1308546 ]
 [ 0.09056928]
 [ 0.11439221]
 [ 0.43985194]
 [ 0.0515844 ]]
```

   From the weights of M = 1, we can observe that "CGPA" is the most contribute feature, and "Research" is the least one.

2. **(a) Which basis function will you use to further improve your regression model, Polynomial, Gaussian, Sigmoidal, or hybrid? Explain**

```
M =  2 , training error: 0.00022176323538085685 , validation error: 0.002483574990422217
```

```
Gaussian - training error: 0.0013891708657683601 , validation error: 0.003336239225609239
```

```
Sigmoid - training error: 0.0011235875268768294 , validation error: 0.0031850896293164127
```

   From the above results, polynomial basis function has lower error both in training and validation stages comparing to the others. Thus, I would use polynomial basis function in the following questions.

(b) Introduce the basis function you just decided in (a) to linear regression model and analyze the result you get. (Hint: You might want to discuss about the phenomenon when model becomes too complex.) **Code Result, Explain**

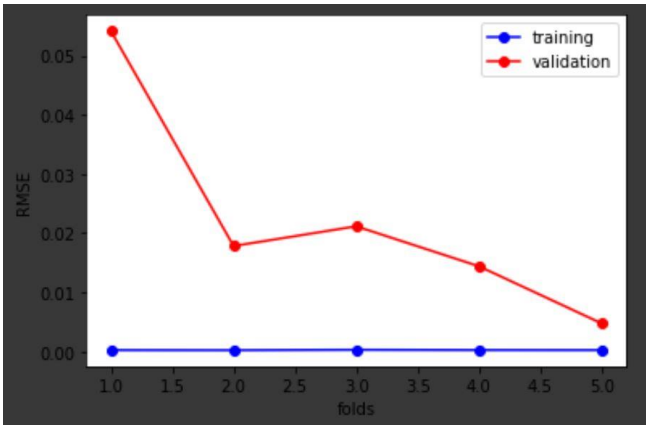
φ(x) = [φ1(x), φ2(x), ..., φN (x), φbias(x)]

weights of M=2:

```
[[-1.85863374] [-0.11041452]  [ 6.39597288]
 [ 6.11104602]  [ 1.51524991]  [ 0.02840861]
 [-1.58163423]  [ 0.6273461 ]  [ 0.53736669]
 [ 0.15413275]  [-7.32002854]  [-0.2997792 ]
 [-0.50502966]  [ 2.80596566]  [-0.36359679]  [ 0.12678768]
 [ 0.28851679]  [-1.09355813]  [ 0.1132395 ]  [-0.79088025]
 [ 0.54333375]  [ 0.04214218]  [-0.34706119]  [-0.10253465]
 [-2.6306552 ]  [ 0.74371821]  [ 0.16746312]  [ 2.43955529]
 [-0.62413437]  [-0.16602873]  [-0.65736641]  [ 5.8971264 ]
 [-1.52341625]  [ 0.7686077 ]  [-1.4898756 ]  [-6.87908304]]
```

The weights of polynomial function (M = 2) has 36 (= 1 bias + 7 + 28) parameters, and each parameter corresponds to the weight of $x_i x_j$. From the above figures, there are some feature combinations have negative relation with labels, and the others has positive relation with labels. Furthermore, "TOFEL_score" and "Research" is the most contribute feature combination, and "GRE_score" and "CGPA" is the least one.

(c) Apply N-fold cross-validation in your training stage to select at least one hyperparameter(order, parameter number, ...) for model and do some discussion (underfitting, overfitting). **Code Result, Explain**



Here I set the fold number N=5 in the cross-validation. From the above figure, we can see the training error in all five folds are close to 0.00. Validation error varies in five folds and all much larger than training error. It's a phenomena of ***overfitting****: "the production of an analysis that corresponds too closely or exactly to a particular set of data, and may therefore fail to fit additional data or predict future observations reliably".*

3. **(a) What is the key difference between maximum likelihood approach and maximum a posterior approach? Explain**

Comparing the equation of MAP with MLE, we can see that the only difference is that MAP includes prior in the formula, which means that the likelihood is weighted by the prior in MAP.

In the other words, MLE is a special case of MAP, when prior follows a uniform distribution.

(b) Use Maximum a posterior approach method to retest the model in 2 you designed. You could choose Gaussian distribution as a prior. Code Result

```
MAP training error: 0.0003570744413858093 , validation error: 0.0023493203528307024
ML training error: 0.11558635628263877 , validation error: 0.11756318068829154
```

(c) Compare the result between maximum likelihood approach and maximum a posterior approach. Is it consistent with your conclusion in (a)? Explain

Yes, the result is consistent with my conclusion in (a).

ML does not use any prior knowledge about the expected distribution of the parameters θ and thus may overfit to the particular data. MAP method adds a prior distribution of the parameters θ, and it has also to conform to prior knowledge about the parameter distribution.

MAP estimation can therefore be seen as a regularization of ML estimation.