Machine Learning (Homework 2)

Due date : 12/01 23:59

1 Sequential Bayesian Learning (50%)

Conjugate prior assures that the posterior distribution has the same functional form as the prior. This means we can use the posterior we computed this time as the prior for the next time. This property plays an important role in sequential Bayesian learning.

Dataset:
The file 1_data.mat contains two sequences $\mathbf{x} = \{x_1, x_2, ..., x_{100} | 0 \le x_i \le 2\}$ and $\mathbf{t} = \{t_1, t_2, ..., t_{100}\}$ which represent the input sequence and the corresponding target sequence, respectively.

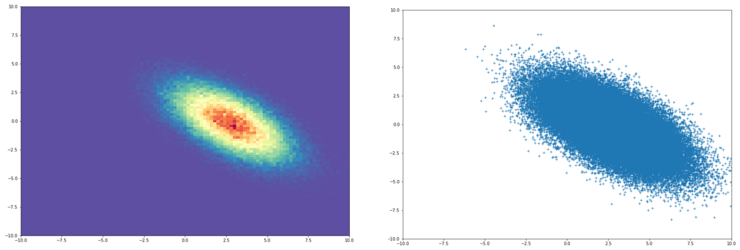
Bayesian Learning:
The posterior distribution $p(\mathbf{w}|\mathbf{t}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N)$ with the given prior $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_0 = \mathbf{0}, \mathbf{S}_0^{-1} = 10^{-6}\mathbf{I})$. The precision of likelihood function $p(\mathbf{t}|\mathbf{w}, \beta)$ or $p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta)$ is chosen to be $\beta = 1$.

Basis Function:
Please apply the sigmoidal basis functions $\boldsymbol{\phi} = [\phi_0, \dots, \phi_{M-1}]^\top$ of the form $\phi_j(x) = \sigma(\frac{x-\mu_j}{s})$ where $\sigma(a)$ is the logistic sigmoid function defined in (3.6). In this exercise, please take the following parameter settings for your basis functions: $M = 3$, $s = 0.1$ and $\mu_j = \frac{2j}{M}$ with $j = 0, \dots, (M-1)$. please take the data size to be $N = 5, 10, 30$ and 80 for the following two questions:

1. Similar to Fig. 3.9, please generate five curve samples from the parameter posterior distribution.

2. Similar to Fig. 3.8, please plot the predictive distribution of target value t and show the mean curve and the region of variance with one standard deviation on either side of the mean curve.

In addition, similar to the middle column of Fig. 3.7, please arbitrarily select two weights by yourself and carefully plot the corresponding prior distributions when $N = 5, 10, 30$ and 80.

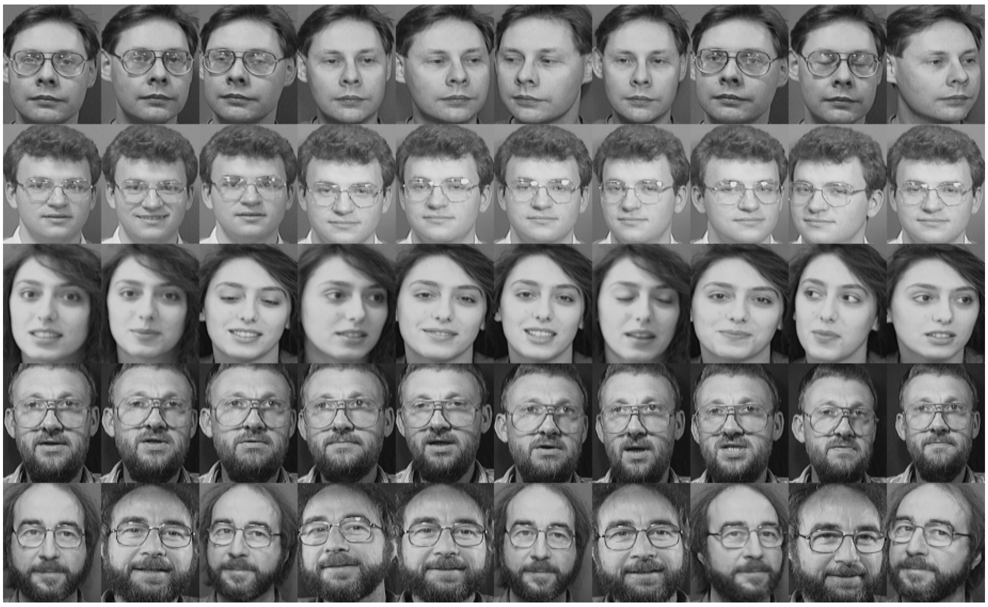


(Both figures are acceptable, but left one will get higher score)

Note : Please train your model by fitting data sequentially, this means when you already compute the result of case $N = 10$, you can continue to compute the result of case $N = 30$.

2 Logistic Regression (50%)

You are given the data set of Human faces (Faces.zip). This data set contains 5 subjects. There are ten different images of a subject. In this exercise, you will implement *gradient descent* and *Newton-Raphson* algorithms to construct a multiclass logistic regression model with softmax transformation (e.g. $p(C_k|\boldsymbol{\phi}) = y_k(\boldsymbol{\phi}) = \exp(a_k)/\sum_j \exp(a_j)$). The error function is formed by using the cross-entropy function as $E(\mathbf{w}) = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \ln y_{nk}$.



Note : You need to normalize the data before training and should randomly select five images as training data and the others as testing data for each subject.

1. Set the initial $\mathbf{w}$ to be zero, and show the learning curve of $E(\mathbf{w})$ and the accuracy of classification versus the number of epochs until convergence of training data. Gradient descent algorithm is applied.

2. Show the classification result of test data.

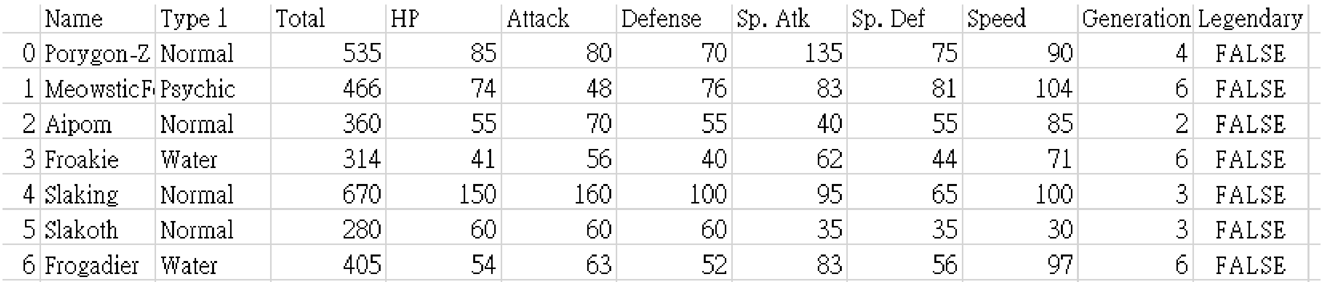
3. Use the principal component analysis (PCA) to reduce the dimension of data and plot five eigenvectors corresponding to top five eigenvalues.

4. Repeat 1 and 2 by applying Netwon-Raphson algorithm. PCA should be used to reduce the dimension of face images to 2, 5 and 10. Make some discussion.

5. Make some discussion on the results of Netwon-Raphson and gradient descent algorithms.

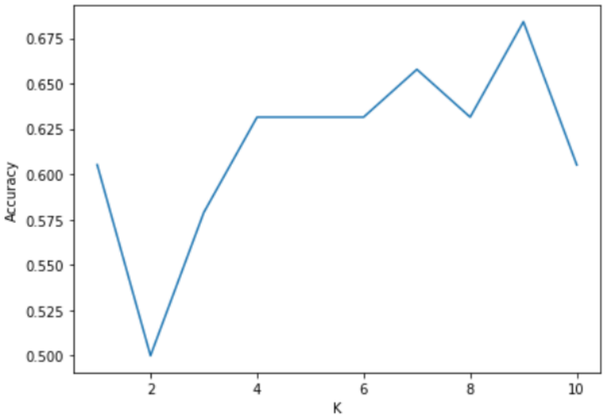
3 Nonparametric Methods (Bonus 20%)

You are given the data set of Pokemons (Pokemon.csv). This data set contains 3 classes (Water, Normal and Psychic). The column "Type 1" in the file is the labels of classes (target values), the other dimensions are the features of data. In this exercise, you will implement K-nearest-neighbor to construct a multiclass classification model. You may refer Section 2.5.



	Name	Type 1	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
0	Porygon-Z	Normal	535	85	80	70	135	75	90	4	FALSE
1	MeowsticF	Psychic	466	74	48	76	83	81	104	6	FALSE
2	Aipom	Normal	360	55	70	55	40	55	85	2	FALSE
3	Froakie	Water	314	41	56	40	62	44	71	6	FALSE
4	Slaking	Normal	670	150	160	100	95	65	100	3	FALSE
5	Slakoth	Normal	280	60	60	60	35	35	30	3	FALSE
6	Frogadier	Water	405	54	63	52	83	56	97	6	FALSE

1. K-nearest-neighbor classifier is implemented in the following procedure:
   - There are 158 data samples in this dataset.You should use first 120 samples as training data, and the remaining 38 samples as test data. (This is unbalance dataset)
   - You need to preprocess all features by subtracting the mean and normalizing by standard deviation. (formula : $\frac{x-\mu}{\sigma}$)
   - In inference stage, you compare each test sample with 120 training samples and measure the Euclidean distance between them.
   - You can use the class with the largest number of occurrences for those K closest training samples to test sample as the prediction of this test sample.
   - Try different K (from 1 to 10).
   - Plot the figure of accuracy where horizontal axis is K and vertical axis is accuracy.



2. Please implement the principal component analysis (PCA) for training samples and reduce the dimension of training and test data to 7, 6, and 5 by using the bases obtained from PCA. Repeat the above procedure.

4 Rule

1. Please name the assignment as HW2_StudentID_Name.zip (e.g. HW2_0123456_XXX.zip).
2. Your zip file should contain source code and report. Do not paste your code in the report.
3. Your report should be converted to pdf file.
4. Only .py and .m are acceptable for your source code
5. Your implementation will be graded according to
   - completeness
   - algorithm correctness
   - model description

  - discussion

6. Using Python and NumPy is encouraged for you in machine learning areas, and matlab is also acceptable.

7. Don’t use high level toolbox/module functions (e.g. sklearn, polyfit).

8. DO NOT PLAGIARISM (We will check program similarity score.)