Machine Learning (Homework 3)

Due date : 12/27

1 Gaussian Process for Regression (60%)

In this exercise, you will implement the Gaussin process (GP) for regression. The file data gp.mat contains $N = 100$ input samples $\mathbf{x} : \{x_1, x_2, \ldots, x_{100}\}, 0 \leq x_i \leq 2$ and the corresponding target values $\mathbf{t} : \{t_1, t_2, \ldots, t_{100}\}$. Please take the first 60 points as the training set and the rest as the test set. A regression function $y(\cdot)$ is used to express the target value by

$$t_n = y(x_n) + \epsilon_n$$

where the noise signal ϵ_n is Gaussian distributed by $\epsilon_n \sim \mathcal{N}(0, \beta^{-1})$ with $\beta^{-1} = 1$.

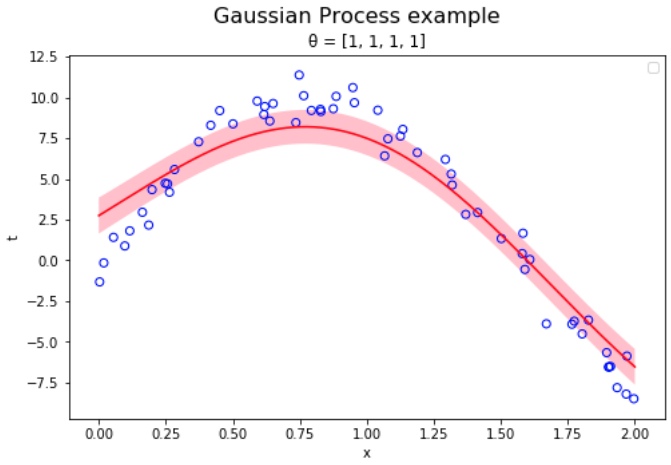
1. Implement the Gaussian process with the exponential-quadratic kernel function given by

$$k(\mathbf{x}_n, \mathbf{x}_m) = \theta_0 \exp\left\{ -\frac{\theta_1}{2}||\mathbf{x}_n - \mathbf{x}_m||^2 \right\} + \theta_2 + \theta_3 \mathbf{x}_n^\top \mathbf{x}_m$$

   where the hyperparameters $\boldsymbol{\theta} = \{\theta_0, \theta_1, \theta_2, \theta_3\}$ are fixed. Use the training set and run four different combinations:

   - linear kernel $\boldsymbol{\theta} = \{0, 0, 0, 1\}$
   - squared exponential kernel $\boldsymbol{\theta} = \{1, 4, 0, 0\}$
   - exponential-quadratic kernel $\boldsymbol{\theta} = \{1, 4, 0, 5\}$
   - exponential-quadratic kernel $\boldsymbol{\theta} = \{1, 32, 5, 5\}$

2. Plot the prediction result like Figure 6.8 of textbook for training set but one standard deviation instead of two and without the green curve. The title of the figure should be the value of hyperparameters used in this model. The red line shows the mean $m(\cdot)$ of the GP predictive distributioin. The pink region corresponds to plus and minus one standard deviation. Training data points are shown in blue. An example is shown as follows.



3. Calculate the corresponding root-mean-square errors

$$E_{\mathrm{RMS}} = \sqrt{\frac{1}{N}(m(x_n) - t_n)^2}$$

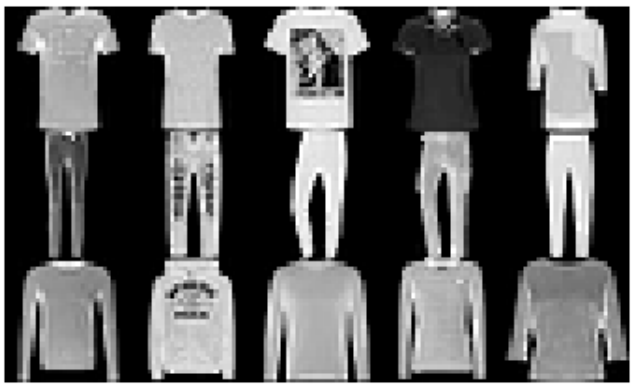
for both training and test sets with respect to the four kernels.

4. Try to tune the hyperparameters by trial and error to find the best combination for the dataset or apply the automatic relevance determination (ARD) in Chapter 6.4.4 of textbook. (If you implement the ARD method, you will receive extra bonus points!)

5. Explain your findings and make some discussion.

2 Support Vector Machine (40%)

Support vector machine (SVM) is known as a popular method for pattern classification. In this exercise, you will implement SVM for classification by using Fashion-MNIST which is a dataset of Zalando's article images as given in x_train.csv and t_train.csv. The input data include three categories: T-shirt, Trouser and Pullover. Each example is a 28x28 grayscale image, associated with a label.



Data Description

- **x_train** is a 18000×784 matrix, where each row is the first two scaled principal values of a training image.
- **t_train** is a 18000×1 vector, which records the classes of the training images. 0, 1, 2 represent T-shirt, Trouser and Pullover, respectively.

In the training procedure of SVM, you need to optimize with respect to Lagrange multipliers $\alpha = \{\alpha_n\}$. Here, you can use the Sequential Minimal Optimization to solve the problem. For details, you can refer to the paper [Platt, John. "Sequential minimal optimization: A fast algorithm for training support vector machines" (1998)]. For matlab, smo.m is provided for you to obtain the multipliers (coefficients). You are required to use this result to implement the rest of SVM algorithms. As for Python, scikit-learn is a free software machine learning library which introduces sklearn.svm to run model training. You are allowed to use this library to calculate the multipliers (coefficients) instead of using the **predict function** directly.

In this exercise, you will implement two kinds of kernel SVM

$$y(\mathbf{x}) = \sum_{n=1}^{N} \alpha_n t_n k(\mathbf{x}, \mathbf{x}_n) = \mathbf{w}^\top \mathbf{x} + b$$

$$\mathbf{w} = \sum_{n=1}^{N} \alpha_n t_n \phi(\mathbf{x}_n)$$

- **linear kernel :**

$$k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j$$

- **polynomial (homogeneous) kernel of degree 2:**

$$k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j) = (\mathbf{x}_i^\top \mathbf{x}_j)^2$$

$$\phi(\mathbf{x}) = [x_1^2, \sqrt{2}x_1x_2, x_2^2]$$

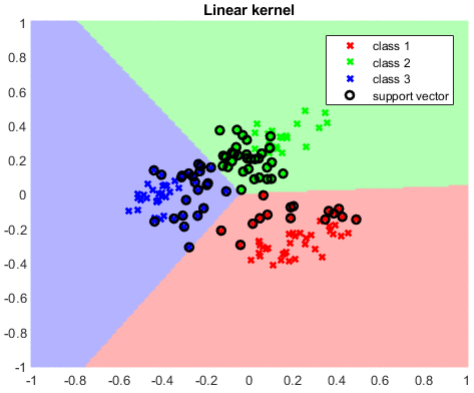
$$\mathbf{x} = [x_1, x_2]$$

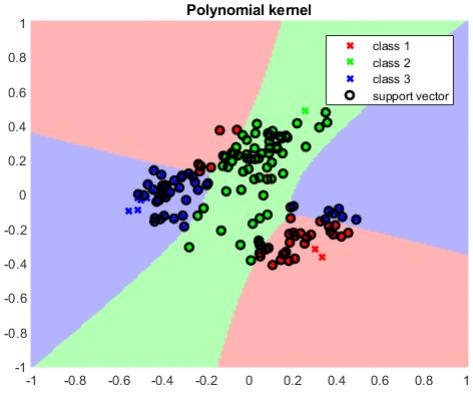
SVM is binary classifier, but the application here has three classes. To deal with this issue, there are two decision approaches, one is "one-versus-the-rest", and another is "one-versus-one"

1. Analyze the difference between two decision approaches. Decide which one you want to use and explain why you choose this approach.
2. Use the dataset to build a SVM with linear kernel to do multi-class classification. Then plot the corresponding decision boundary and support vectors.
3. Repeat (2) with polynomial kernel (degree = 2).
4. Discuss the difference between (2) and (3).

Hints

- In this exercise, we strongly recommend using matlab to avoid tedious preprocessing occurred in python.
- If you use other languages, you are allowed to use toolbox only for multipliers (coefficients).
- You need to implement the whole algorithms except for multipliers (coefficients).





3 Gaussian Mixture Model (30%)

In this exercise, you will implement a Gaussian mixture model (GMM) and apply it in image segmentation. First, use the K-means algorithm to find K central pixels. Second, use Expectation maximization (EM) algorithm (please refer to textbook p.438-p.439) to optimize the parameters of the model. The input data is hw3_3.jpeg. According to the maximum likelihood, you can decide the color μ_k, $k \in [1, \ldots, K]$ of each pixel x_n of ouput image.

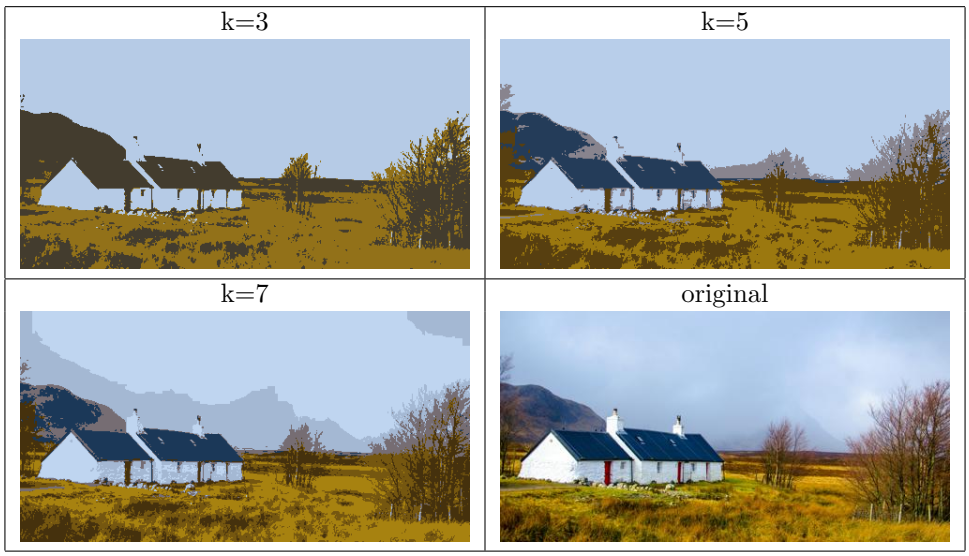
1. Please build a K-means model by minimizing

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} ||x_n - \mu_k||^2$$

   and show the table of estimated $\{\mu_k\}_{k=1}^{K}$.

2. Use $\{\mu_k\}_{k=1}^{K}$ from the K-means model as the means, and calculate the corresponding variances σ_k^2 and mixing coefficient π_k for initialization of GMM $p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \sigma_k^2)$. Optimize the model by maximizing the log likelihood function $\log p(x|\pi, \mu, \sigma^2)$ through EM algorithm. Plot the log likelihood curve of GMM. (Please terminate EM algorithm when the iteration arrives 100)

3. Repeat step (1) and (2) for $K = 3$, 5, 7, and 10. Please show the resulting images in your report. Below are some examples.



4. The input image is with licence free for personal and commercial use. Image from: https://www.pexels.com/photo/clouds-daylight-forest-grass-371589/