Assignment 2 : CS-E4830 Kernel Methods in Machine Learning 2017

The **deadline** for this assignment is **Thursday 05.10.2017 at 4pm**. If you have **questions** about the assignment, you can ask them in the 'General discussion' section on MyCourses or attend the Q&A session on 29.09.17 at 8:30 am in U1/U154, Otakaari 1. We will have a tutorial session regarding the **solutions** of this assignment on 06.10.17 at 8:30 am in U1/U154. The solutions will also be available in MyCourses.

The report for the assignment should contain your proposed solution for the pen and paper exercises. Regarding the computer exercise, you should explain what you have done and include figure(s) (correctly annotated with legend and axe titles) or table(s) that summarize your results. You should also comment about the results you obtained. The report **and** the code (in Matlab, Python or R) must be returned as a single .zip file to MyCourses (naming convention: lastname-firstname-assignment2.zip). In case you have hand-written solutions for the pen and paper exercises you can scan them and add them in your report or submit them to any of the course teaching assistants (Parisa, Celine or Sandor) in their office (A319 or A323).

Pen and paper exercise

Kernel computation

Let $\kappa_1 : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $\kappa_2 : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be two kernel functions. These kernels are respectively associated with the feature maps ϕ_1 and ϕ_2. Let $\mathbf{K}_1$ and $\mathbf{K}_2$ be the kernel matrices of κ_1 and κ_2 over some data X such that $[\mathbf{K}_1]_{i,j} = \kappa_1(\mathbf{x}_i, \mathbf{x}_j)$ and $[\mathbf{K}_2]_{i,j} = \kappa_2(\mathbf{x}_i, \mathbf{x}_j)$ for $\mathbf{x}_i, \mathbf{x}_j \in X$.

Question 1: (1 point)

State and prove whether $\mathbf{K} = \mathbf{K}_1\mathbf{K}_2$ (matrix multiplication) is a valid kernel matrix or not. Please support your answer with a mathematical proof or a counter example.

Question 2: (1 point)

Let $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a function defined as $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \kappa_1(\mathbf{x}_i, \mathbf{x}_j)\kappa_2(\mathbf{x}_i, \mathbf{x}_j)$ for $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$. Prove that κ is a valid kernel function and that its values can be written as $\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$, where $\phi(\mathbf{x}) = \phi_1(\mathbf{x}) \otimes \phi_2(\mathbf{x})$ (tensor product of two vectors).

(Hint: The tensor product of two vectors $\mathbf{v} = [v_1, v_2]'$ and $\mathbf{w} = [w_1, w_2, w_3]'$ is defined as: $\mathbf{v} \otimes \mathbf{w} = [v_1w_1, v_1w_2, v_1w_3, v_2w_1, v_2w_2, v_2w_3]'$, where the components of the tensor product $\mathbf{v} \otimes \mathbf{w}$ are computed as the product of all possible pairs of the components of $\mathbf{v}$ and $\mathbf{w}$.)

Kernel centering

Let $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel function and $\phi : \mathcal{X} \to F$ a feature map associated with this kernel. Let $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_\ell\}$ be the set of training inputs.

Centering the data in the feature space consists in moving the origin of the feature space to the center of mass of the training features $\frac{1}{\ell}\sum_{i=1}^{\ell}\phi(\mathbf{x}_i)$ and generally helps to improve the performance. After centering, the feature map is given by: $\phi_c(\mathbf{x}) = \phi(\mathbf{x}) - \frac{1}{\ell}\sum_{i=1}^{\ell}\phi(\mathbf{x}_i)$. We will see in this question that centering can be performed implicitly by transforming the kernel values.

Question 3: (1 point)

Show that

$$\kappa_c(\mathbf{x}_i, \mathbf{x}_j) = \kappa(\mathbf{x}_i, \mathbf{x}_j) - \frac{1}{\ell}\sum_{p=1}^{\ell}\kappa(\mathbf{x}_p, \mathbf{x}_j) - \frac{1}{\ell}\sum_{q=1}^{\ell}\kappa(\mathbf{x}_i, \mathbf{x}_q) + \frac{1}{\ell^2}\sum_{p,q=1}^{\ell}\kappa(\mathbf{x}_p, \mathbf{x}_q),$$

where $\kappa_c(\mathbf{x}_i, \mathbf{x}_j) = \langle\phi_c(\mathbf{x}_i), \phi_c(\mathbf{x}_j)\rangle$ is the kernel value after centering.

Loss functions

Let $y \in \{-1, 1\}$ and g be a real-valued function mapping $\mathbf{x}$ to y. In this exercise, we consider the following loss functions:

- *Logistic loss*: $\ell_{logistic}(g(\mathbf{x}), y) = \log(1 + \exp(-yg(\mathbf{x})))$,
- *Quadratic loss*: $\ell_2(g(\mathbf{x}), y) = (g(\mathbf{x}) - y)^2$,
- *Hinge loss*: $\ell_{hinge}(g(\mathbf{x}), y) = \max(0, 1 - yg(\mathbf{x}))$.

These loss functions can be rewritten as functions of one variable $m = yg(\mathbf{x})$:

- *Logistic loss*: $\ell_{logistic}(m) = \log(1 + \exp(-m))$,
- *Quadratic loss*: $\ell_2(m) = (m - 1)^2$,
- *Hinge loss*: $\ell_{hinge}(m) = \max(0, 1 - m)$.

Question 4: (1 point)

Show whether the above loss functions are differentiable or not, and if it is differentibale compute the derivative.

Question 5: (1 point)

Show whether the above loss functions are convex or not.

A function $f : X \to \mathbb{R}$ is convex if for any two points x_1 and x_2 of X, and any number $\lambda \in [0, 1]$ we have $f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2)$. (Hint: if the function is differentiable, see Question 4, the nonnegativity of the second derivative at every x can prove the convexity.)

Computer exercises

In this exercise, you will apply the kernel ridge regression approach on the forest fires dataset. The goal is to predict the burned area of the forest using meteorological and other data (more information can be found at https://archive.ics.uci.edu/ml/datasets/Forest+Fires).

The file 'data_all.mat' contains the inputs and labels for the training set (X_train and y_train), as well as the inputs and labels for the test set (X_test, y_test). If you are not using Matlab, you can load these files separately: 'X_train.txt', 'X_test.txt', 'y_train.txt' and 'y_test.txt'. The label vector has been normalized.

Question 6 (1 point):

Write a function that implements the kernel ridge regression algorithm. This function should take in input:

- $\mathbf{K}_{train}$: kernel matrix between training examples,
- $\mathbf{K}_{train\_test}$: kernel matrix between the training and test examples,
- $\mathbf{y}_{train}$: vector containing the training outputs,
- λ: regularization parameter

and should return the predictions for the test examples: $g(\mathbf{x}) = \mathbf{y}'(\lambda \mathbf{I}_\ell + \mathbf{K})^{-1}\mathbf{k}(\mathbf{x})$. $\mathbf{k}(\mathbf{x})$ is a vector containing the kernel values between each training point and the point $\mathbf{x}$ to be predicted: $\mathbf{k}(\mathbf{x}) = [\kappa(\mathbf{x}_1, \mathbf{x}), \ldots, \kappa(\mathbf{x}_\ell, \mathbf{x})]'$.

Question 7 (1 point):

Write a function that computes the kernel matrix $\mathbf{K}$ of a polynomial kernel κ between two matrices $\mathbf{X}$ and $\mathbf{Z}$. First compute the linear kernel $\mathbf{K}_{i,j}^{linear} = \langle \mathbf{x}_i, \mathbf{z}_j \rangle$, then from the linear kernel the polynomial kernel can be computed by $\mathbf{K}_{i,j}^{poly} = \kappa^{poly}(\mathbf{x}_i, \mathbf{z}_j) = (\mathbf{K}_{i,j}^{linear} + c)^S$.

As the values of $\langle \mathbf{x}_i, \mathbf{z}_j \rangle$ can be very large, it is better to normalize the linear kernel before it is inserted into the polynomial one, where the normalized linear kernel can be computed by this formula

$$\mathbf{K}_{i,j}^{linear} = \kappa^{linear}(\mathbf{x}_i, \mathbf{z}_j) = \frac{\langle \mathbf{x}_i, \mathbf{z}_j \rangle}{\sqrt{\langle \mathbf{x}_i, \mathbf{x}_i \rangle \langle \mathbf{z}_j, \mathbf{z}_j \rangle}}.$$

Question 8 (1.5 points):

In this question, you will apply the kernel ridge regression code that you have implemented on the forest fires dataset using a polynomial kernel. The parameter c of the polynomial kernel will be fixed to 0.1.

Compute the mean squared error (MSE) obtained on the training and test set for different choices of the regularization parameter λ ($\lambda = [10^{-3}, 10^{-2}, \ldots, 10^{5}]$) and of the kernel parameter S ($S = [1, 2, 4, 6, 8, 10]$):

$$MSE_{train} = \frac{1}{n_{train}} \sum_{i \in S_{train}} (\hat{g}(\mathbf{x}_i) - y_i)^2,$$

$$MSE_{test} = \frac{1}{n_{test}} \sum_{i \in S_{test}} (\hat{g}(\mathbf{x}_i) - y_i)^2,$$

where n_{train} and n_{test} are the number of training and test examples.

Plot the MSE_{train} and MSE_{test} obtained for different values of λ and S. Comment on the curve(s) you obtain.

(Hint: the test MSE should be around 2.4×10^{-3} for the best parameters.)

Question 9 (1.5 points):

Implement a selection of the kernel and regularization parameters using k-fold cross-validation on the training set with $k = 5$. In k-fold cross-validation, the training set is randomly split into k subsamples of approximately equal size. Here, use the random split of the training data in five folds that is contained in the variable cv_fold (or the file cv_fold.txt). Use one subsample for testing the method and the remaining $k - 1$ subsamples as training data. Measure the mean squared error obtained on the test subsample. Repeat the cross-validation process k times, with each of the k subsamples used exactly once as the test data. Compute the averaged MSE over the k folds. Repeat the whole cross-validation process for different values of the parameters λ and S.

Select the pair of parameters that minimizes the averaged MSE. Indicate the parameter values you obtain.