# Assignment 2 : CS-E4830 Kernel Methods in Machine Learning 2019

The **deadline** for this assignment is **Thursday 28.02.2019 at 4pm**. If you have **questions** about the assignment, you can ask them in the 'General discussion' section on MyCourses. We will have a tutorial session regarding the **solutions** of this assignment on 28.02.19 at 4:15 pm in TU1(1017), TUAS, Maarintie 8. The solutions will also be available in MyCourses.

Please follow the **submission instructions** given in MyCourses: `https://mycourses.aalto.fi/course/view.php?id=20602&section=2`.

## Pen & Paper exercise

### 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 moves 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 1**: (3 points)

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.

**Solution**

$$\begin{aligned}
\kappa_c(\mathbf{x}_i, \mathbf{x}_j) &= \langle \phi_c(\mathbf{x}_i), \phi_c(\mathbf{x}_j) \rangle \\
&= \langle \phi(\mathbf{x}_i) - \frac{1}{\ell}\sum_{p=1}^{\ell} \phi(\mathbf{x}_p), \;\; \phi(\mathbf{x}_j) - \frac{1}{\ell}\sum_{q=1}^{\ell} \phi(\boldsymbol{x}_q) \rangle \\
&= \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle - \frac{1}{\ell}\langle \phi(\mathbf{x}_i), \sum_{q=1}^{\ell} \phi(\mathbf{x}_q) \rangle - \frac{1}{\ell}\langle \sum_{p=1}^{\ell} \phi(\mathbf{x}_p), \phi(\mathbf{x}_j) \rangle \\
&+ \frac{1}{\ell^2}\langle \sum_{p=1}^{\ell} \phi(\mathbf{x}_p), \sum_{q=1}^{\ell} \phi(\mathbf{x}_q) \rangle \\
\kappa_c(\mathbf{x}_i, \mathbf{x}_j) &= \kappa(\mathbf{x}_i, \mathbf{x}_j) - \frac{1}{\ell}\sum_{q=1}^{\ell} \kappa(\mathbf{x}_i, \mathbf{x}_q) - \frac{1}{\ell}\sum_{p=1}^{\ell} \kappa(\mathbf{x}_p, \mathbf{x}_j) + \frac{1}{\ell^2}\sum_{p,q=1}^{\ell} \kappa(\mathbf{x}_p, \mathbf{x}_q).
\end{aligned}$$

**Centering as Linear Operator**

**Training:** Let $\mathbf{K} \in \mathbb{R}^{n\times n}$ be a symmetric (training) kernel matrix. This kernel matrix was constructed based on the training set $S$. We can now express the centering (with respect to the training set) as linear operator:

$$\mathbf{K}_c = \underbrace{\left[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{n}\right]}_{\text{Centering operator}} \mathbf{K}\left[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{n}\right] \tag{1}$$

$$= \left(\mathbf{K} - \frac{\mathbf{1}\mathbf{1}^T}{n}\mathbf{K}\right)\left[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{n}\right] \tag{2}$$

$$= \mathbf{K} - \underbrace{\mathbf{K}\frac{\mathbf{1}\mathbf{1}^T}{n}}_{*} - \underbrace{\frac{\mathbf{1}\mathbf{1}^T}{n}\mathbf{K}}_{**} + \underbrace{\frac{\mathbf{1}\mathbf{1}^T}{n}\mathbf{K}\frac{\mathbf{1}\mathbf{1}^T}{n}}_{***}, \tag{3}$$

with $\mathbf{I} \in \mathbf{n} \times \mathbf{n}$ being an identity matrix, and $\mathbf{1} \in \{n\}$ being a vector of ones, and:

- * Vector of row-averages repeated n-times along the columns
- ** Vector of column-averages repeated n-times along the rows
- *** Overall mean, repeated for each entry in the $n \times n$-matrix

However, this formulation only works for the training kernel matrix.

## Multiclass(multinomial) classification

On Lecture 4 and 5, the Bayes classifier has been introduced, see Slides 9 and 10 of Lecture 4, and Slides 9 of Lecture 5. On those slides a decision rule to predict the classes, $C_1$ and $C_2$ has been presented. That rule selects that class which has the greater conditional probability at a given x, namely

$$\arg\max_k P(y = C_k | X = x), k = 1, 2$$

. This classification can deal with two classes.

**Question 2**: (2 points)

Let $\mathbf{x}_i \in \mathcal{R}^d$ be an input example, and $\mathbf{w}_k \in \mathcal{R}^d, k = 1, \ldots, K$ a set of parameter vectors assigned to each class in the multi-class classification. Let the probability $P(Y_i = k|X = x_i)$ of a class with respect to $\mathbf{x}_i$ be given by $\frac{1}{Z}\exp(\langle \mathbf{w}_k, \mathbf{x}_i \rangle)$, called *Gibbs measure*, where $Z$ is a normalization factor to guarantee that $\frac{1}{Z}\exp(\langle \mathbf{w}_k, \mathbf{x}_i \rangle)$ is a probability.

The task is to suggest a multi-class decision function for this concrete probability model, and derive the value of $Z$ for a fixed number of classes.

Hint: Try to understand the formula on Slide 9 of Lecture 5 about Bayes classifier.

**Solution**

Recall that a given $x$ class $C_j$ is selected in the binary classification if $P(y = j|X = x) \geq 0.5$. It means $j = \arg\max_k[P(y = k|X = x), k = 1, 2]$, thus class with the largest probability is chosen. We can follow the same approach in multi-class classification

$$j = \arg\max_k P(y = k|X = x), \; k = 1, \ldots, K$$

Since $P(y = k|X = x) = \frac{1}{Z}\exp(\langle \mathbf{w}_k, \mathbf{x}_i \rangle)$ therefore we can write

$$j = \arg\max_k \frac{1}{Z}\exp(\langle \mathbf{w}_k, \mathbf{x}_i \rangle), \; k = 1, \ldots, K$$

To compute $Z$ we can exploit that the sum of the conditional probabilities of all classes has to be 1, thus we have

$$Z = \sum_{k=1}^{K} P(Y = k|X = x) = \sum_{k=1}^{K} \exp(\langle \mathbf{w}_k, \mathbf{x}_i \rangle).$$

Finally the multi-class classification has this form

$$j = \arg\max_k \frac{\exp(\langle \mathbf{w}_k, \mathbf{x}_i \rangle)}{\sum_{k=1}^{K} \exp(\langle \mathbf{w}_k, \mathbf{x}_i \rangle)} \;, k = 1, \ldots, K.$$

**Question 3**: (2 Bonus points)

We are given a binary classification problem, where we know the probability densities, $p(x, C_1)$ and $p(x, C_2)$ relating to the two classes. Prove that the probability of the minimum misclassification error satisfies this inequality:

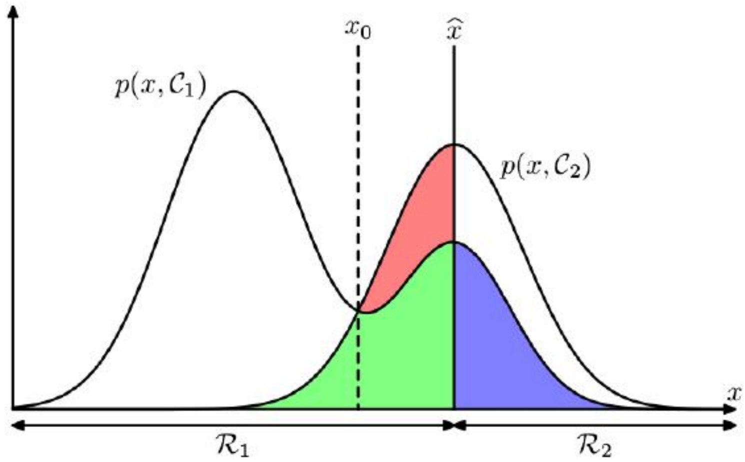
$$P(\text{Minimum misclassification error}) \leq \int_{x \in \mathcal{X}} (p(x, C_1)p(x, C_2))^{1/2}\, dx \tag{4}$$

In the proof you can apply the following inequality, for any $a \geq 0$ and $b \geq 0$ we have

$$\min(a, b) \leq (ab)^{1/2}. \tag{5}$$

To derive what is the minimum misclassification error, recall the figure on Slide 9 of Lecture 5 about Bayes classifier. Think about which part of the function graph covers that error, and how it can be computed.

**Solution**

Recall this figure from the lectures: 


The minimum misclassification error corresponds to the union of the green and the blue area. That area covers the overlap between the density functions of the classes. It is the consequence of the fact that in the classification that class is chosen at a given $x$ whose probability is the largest, thus the minimum error corresponds to the intersection of the areas under the curves of density functions, where both classes have the probability to occur. The area of the intersection is given by the integral

$$P(\text{Minimum misclassification error}) = \int_{x \in \mathcal{X}} \min(p(x, C_1)p(x, C_2))dx$$

Now by exploiting the inequality $\min(a, b) \leq (ab)^{1/2}$, where $a = p(x, C_1)$ and $b = p(x, C_2)$ for any $x$, we can write

$$P(\text{Minimum misclassification error}) = \int_{x \in \mathcal{X}} \min(p(x, C_1)p(x, C_2))dx \leq \int_{x \in \mathcal{X}} (p(x, C_1)p(x, C_2))^{1/2}\, dx.$$

## Computer Exercise

Solve the computer exercise in JupyterHub (`https://jupyter.cs.aalto.fi`). The instructions for that are given in MyCourses: `https://mycourses.aalto.fi/course/view.php?id=20602&section=3`.