INF554 - Machine and Deep Learning
Lab 5 - Kernel Methods

1 Kernel Methods

First we look at kernels, and then their use in machine learning via the 'kernel trick'. Then we look at the well-known kernel method "support vector machine" classifier, and finally, we will implement and study spectral clustering which is an unsupervised method strongly related to (in fact, is a particular case of) kernel methods.

1.1 Positive Definite Kernels

A **positive definite kernel** on a set $\mathcal{X}$ is a function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ which is:

- symmetric, i.e. $\forall\ (x_1, x_2) \in \mathcal{X}^2$:
$$K(x_1, x_2) = K(x_2, x_1);$$
- and verifying, $\forall n \in \mathbb{N}, \forall (x_1, x_2, ..., x_n) \in \mathcal{X}^n, \forall (a_1, a_2, ..., a_n) \in \mathbb{R}^n$, the inequality:
$$\sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j K(x_i, x_j) \geq 0.$$

Alternatively, a function $K : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ is a positive definite kernel if and only if $\forall n \in \mathbb{N}, \forall (x_1, x_2, ..., x_n) \in \mathcal{X}^n$, the matrix $\mathbf{K}$ defined as:
$$\mathbf{K}_{i,j} = K(x_i, x_j)$$
is symmetric and positive semidefinitive. In a nutshell, positive definite kernels aim at providing **similarity measures** among elements in $\mathcal{X}$. One of the most simple positive definite kernel is probably the **linear kernel** $K^{\text{lin}} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ defined as:

$$\forall (x_1, x_2) \in \mathbb{R}^d \times \mathbb{R}^d, \quad K^{\text{lin}}(x_1, x_2) = \langle x_1, x_2 \rangle_{\mathbb{R}^d}.$$

Question 1
Prove that K^{lin} is a positive definite kernel.

Task 1
Implement the linear kernel K^{lin}.

We can also construct more sophisticated kernels by adding, scaling and/or multiplying kernels:

Question 2
Let K_1, K_2 be two positive definite kernels $K_1 : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $K_2 : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Show that $K_1 + K_2$ is a positive definite kernel.

Question 3
Let K be a positive definite kernels $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Let $c > 0$. Show that cK is a positive definite kernel.

Question 4
(Bonus) Let K_1, K_2 be two positive definite kernel $K_1 : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $K_2 : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Let $K_3 : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ with, $\forall (x_1, x_2) \in \mathcal{X} \times \mathcal{X}, K_3(x_1, x_2) = K_1(x_1, x_2)K_2(x_1, x_2)$. Show that K_3 is a positive definite kernel.

1.2 Kernel Methods in Machine Learning

The main goal of **kernel methods** is to provide separability of classes in a high-dimensional feature space, even if the data was not easily separable in the original data space.

Theorem (Aronszajn, 1950): K is a positive definite kernel, if and only if, there exists a mapping f and a Hilbert space $\mathcal{H}$ such as: $K(x_i, x_j) = \langle f(x_i); f(x_j) \rangle$.

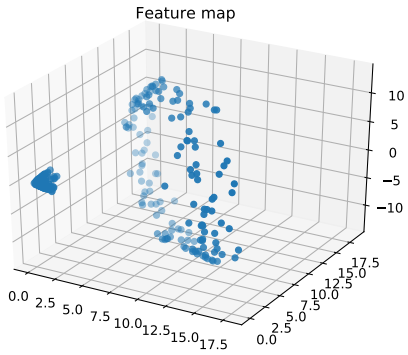


Figure 1: A three dimensional feature map obtained with a polynomial kernel, separating non-linearly separable data

Unlike methods that explicitly compute the map into feature space (as we did with polynomial basis expansion in the first lab), the feature space of kernel methods is *implicit*. As provided by the **kernel trick**, we only need a similarity function for pairwise comparison. This 'trick' is useful for learning problems involving strings, graphs, trees, and so on, where a good feature representation in a fixed-length vector space is costly and challenging to obtain. In this view, a kernel is thus simply a measure of similarity between two instances, as addressed in the previous section.

Question 5
Suppose feature mapping $\phi(\mathbf{x}) = [x_1^2, \sqrt{2}x_1x_2, x_2^2]^\top$. Derive the corresponding polynomial kernel $K(\mathbf{x}, \tilde{\mathbf{x}}) = \langle \mathbf{x}, \tilde{\mathbf{x}} \rangle^2 = (\mathbf{x}^\top \tilde{\mathbf{x}})^2$.

Task 2
Implement the Gaussian kernel function.

2 Support Vector Machines

The support vector machine (SVM) is perhaps the machine learning method which is most associated with kernels. An SVM is a support vector classifier (SVC) with a non-linear kernel. A SVC is a maximum

margin method which allows points to fall into the margin. The idea is to not only to find a decision boundary, but a large margin which separates the two training classes (the decision boundary is the center of this margin).

As with all kernel methods, the key point to find a representation of the model expressed only in terms of inner products (wrt training points $\mathbf{x}_i$). The **dual problem** of the SVC can be written as

$$\max_{\boldsymbol{\alpha}} \left\{ \sum_{i=1}^{n} \alpha_i - \frac{1}{2C} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \right\} \quad \text{subject to} \quad 0 \leq \alpha_i \quad \text{and} \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

for learning, and, for the classifier as

$$\sum_{i=1}^{n} \alpha_i y_i \langle \mathbf{x}_i, \mathbf{x}_j \rangle + b$$

noting that the non-zero α_is define the support vectors (there will probably be relatively few of them).

With the kernel trick, we simply replace the inner products with our kernel of choice. How to chose the kernel? It is typically selected/designed according to the problem domain. An initial question can always be: do we need a non-linear kernel?

> **Task 3**
> Use the `svc` class from the ScikitLearn (`sklearn`) library with the kernels you implemented above. Experiment using `data3.txt` provided in `data/`. There is some plotting code provided for you to visualize the decision boundary and the support vectors. Compare the results with the different kernels. Experiment also with different values of σ.

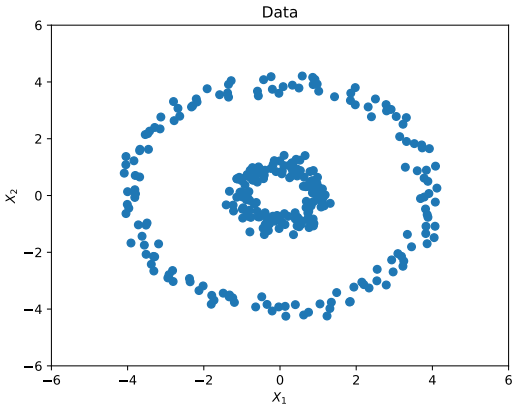
> **Question 6**
> Recall the effect of outliers on MSE (from, e.g., the first lab). What is the effect of outliers on the decision boundary of SVMs.

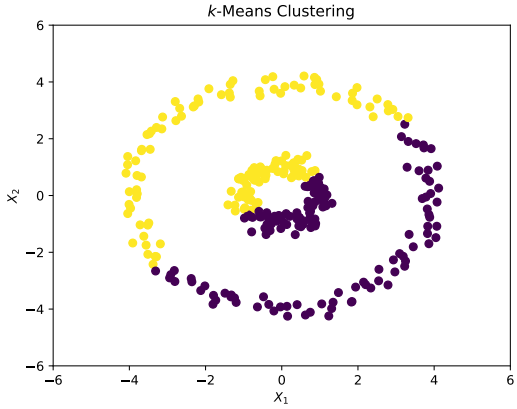
As always we should be careful to avoid overfitting. The C parameter (as it is called in `sklearn`) is a positive value that controls the penalty for misclassified training examples. A large C parameter tells the SVM to try harder to classify all the examples correctly, whereas a smaller C will try harder to get a large margin, allowing for more points to fall into it.

> **Task 4**
> Experiment with the regularization parameter C.

3 Spectral Clustering (Bonus)

Clustering methods like k-means are centrality based methods. However there are particular patterns which are difficult to solve with this approach, such as concentric rings:





In this section we look at a connectivity-based method called **spectral clustering**. This method can be implemented as follows:

Algorithm 1: Spectral Clustering

Data: $(x_1, \ldots, x_N)$

Result: $(\mu_1, \ldots, \mu_K), (S_1, \ldots, S_K)$ Centroids (in the projected space) and cluster assignments.

1 **Obtain a *similarity matrix*** $\mathbf{A}$ under a kernel

2 **Make a *graph*** (adjacency matrix $\mathbf{W}$ and degree matrix $\mathbf{D}$), where instances are vertices, you can use the full graph or only the nearest neighbors.

3 **Eigendecompose** the *Laplacian matrix* $\mathbf{L} = \mathbf{D} - \mathbf{W}$:

$$\mathbf{U}\boldsymbol{\lambda}\mathbf{U}^\top = \mathbf{L}$$

4 **Order columns of** $\mathbf{U}$ (the eigenvectors) according to eigenvalues $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$

5 **Project** (take top-k components):

$$\mathbf{X}' = \mathbf{U}_{:,1:k}$$

6 **Apply** k-means to $\mathbf{X}'$

Question 7

(Bonus) Show that, for all $x \in \mathbb{R}^N$: $x^\top \mathbf{L} x = \frac{1}{2} \sum_{i,j} \mathbf{W}_{i,j}(x_i - x_j)^2$

Question 8

Based on the previous question, show that if all elements of $\mathbf{W}$ are positive then the eigenvalues of $\mathbf{L}$ verify: $0 \leq \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$. Then, show that:

1. If the graph is *connected*, $\mathbf{L}$ has a single null eigenvalue with eigenvector $\mathbf{1}$.
2. For any graph, then number of null eigenvalues of $\mathbf{L}$ is exactly the number of *connected components* of the graph, and that the corresponding eigenvectors are the indicatrix of those components.

Question 9

Based on the previous questions explain why:

1. If there is only one null eigenvalue we can discard the corresponding eigenvector.
2. We use the eigenvectors corrsponding to the smallest eigenvalues.

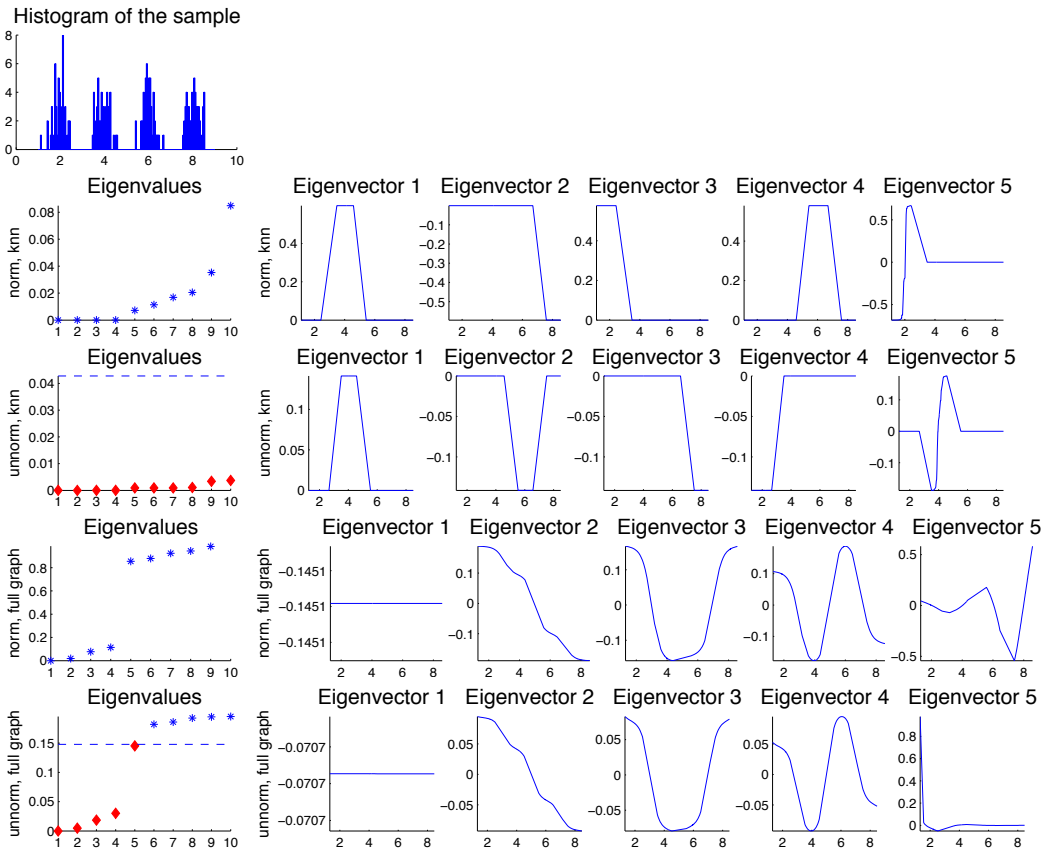


Figure 2: Representation of eigenvalues and eigenvectors for a toy dataset and various graphs. Credit to: https://arxiv.org/abs/0711.0189

Task 5

Implement spectral clustering and apply it to the synthetic data (generated as part of the code given) for $k = 2$ clusters. Plot the result. Plot also the intermediary step of the points in eigenspace.

Note that there is a strong connection of spectral clustering to kernel k-means with a Gaussian kernel. To see this, you would begin by writing k-means as a function of inner products.

Question 10

Show that kernel k-means can be equivalent to spectral clustering. Hint: Write k-means as a function of inner products.