

Chapter 21

Kernel Methods

Given any dataset $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^n$, we call the original d -dimensional space (\mathbb{R}^d) the *input space*. Let ϕ be a (typically non-linear) mapping to another high-dimensional space, which we term the *feature space*. Conceptually, the idea behind non-linear analysis is to work in the feature space, where we can find **linear models**, which in fact correspond to *non-linear models* in the input space, since the linear models in feature space are over non-linear dimensions (or features).

The main problem with this approach is that it requires an explicit transformation of each point \mathbf{x}_i in the input space to the point $\phi(\mathbf{x}_i)$ in the feature space. Since the mapping ϕ can be very high dimensional, even infinite dimensional, working directly in the feature space is not computationally efficient.

If it can be demonstrated that the problem requires only dot products $\phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$ in the feature space, then in fact, we can replace those dot products by Mercer kernels that work directly in input space: $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$. As we saw in Section 20.4, the polynomial and gaussian kernels are examples of such non-linear positive semi-definite kernels.

To summarize, there are two key elements of the kernel trick:

- Show that the analysis task requires only dot products in the feature space: $\phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$
- Replace the dot product by a positive semi-definite kernel that works in the input space: $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$.

We give two examples of kernel methods applied to principal component analysis, and fisher discriminants.

21.1 Kernel Principal Component Analysis (Kernel PCA)

Recall from Section 4.1 that the first principal component captures the direction with the most projected variance. Assuming that all points \mathbf{x}_i have been mapped to the corresponding points $\phi(\mathbf{x}_i)$ in the feature space, we can find the direction of the most variance \mathbf{u}_1 (with $\mathbf{u}_1^T \mathbf{u}_1 = 1$), by solving for the eigenvector corresponding to the largest eigenvalue of Σ_ϕ :

$$\Sigma_\phi \mathbf{u}_1 = \lambda_1 \mathbf{u}_1 \quad (21.1)$$

where the Σ_ϕ is the covariance matrix for the transformed points, given as:

$$\Sigma_\phi = \frac{1}{n} \sum_{i=1}^n \phi(\mathbf{x}_i) \phi(\mathbf{x}_i)^T \quad (21.2)$$

Here we assume for the moment that the points are centered (i.e., the mean has been subtracted from each point), in the feature space, which may not really be the case, since ϕ is some non-linear mapping. We will address how to center the points later.

Plugging in the expansion of Σ_ϕ from (21.2) into (21.1), we get:

$$\left(\frac{1}{n} \sum_{i=1}^n \phi(\mathbf{x}_i) \phi(\mathbf{x}_i)^T \right) \mathbf{u}_1 = \lambda_1 \mathbf{u}_1 \quad (21.3)$$

$$\frac{1}{n} \sum_{i=1}^n \phi(\mathbf{x}_i) (\phi(\mathbf{x}_i)^T \mathbf{u}_1) = \lambda_1 \mathbf{u}_1 \quad (21.4)$$

$$\sum_{i=1}^n \left(\frac{\phi(\mathbf{x}_i)^T \mathbf{u}_1}{n \lambda_1} \right) \phi(\mathbf{x}_i) = \mathbf{u}_1 \quad (21.5)$$

$$\sum_{i=1}^n a_i \phi(\mathbf{x}_i) = \mathbf{u}_1 \quad (21.6)$$

where $a_i = \frac{\phi(\mathbf{x}_i)^T \mathbf{u}_1}{n \lambda_1}$ is a scalar value. From (21.6) we see that the best direction in the feature space, \mathbf{u}_1 , is just a linear combination of the transformed points, where the scalars a_i show the importance of each point towards the direction of most variance.

We can now substitute (21.6) back into (21.3) to get:

$$\left(\frac{1}{n} \sum_{i=1}^n \phi(\mathbf{x}_i) \phi(\mathbf{x}_i)^T \right) \left(\sum_{j=1}^n a_j \phi(\mathbf{x}_j) \right) = \lambda_1 \sum_{i=1}^n a_i \phi(\mathbf{x}_i) \quad (21.7)$$

$$\frac{1}{n} \sum_{i=1}^n \sum_{j=1}^n a_j \phi(\mathbf{x}_i) \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) = \lambda_1 \sum_{i=1}^n a_i \phi(\mathbf{x}_i) \quad (21.8)$$

$$\sum_{i=1}^n \left(\phi(\mathbf{x}_i) \sum_{j=1}^n a_j \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) \right) = n \lambda_1 \sum_{i=1}^n a_i \phi(\mathbf{x}_i) \quad (21.9)$$

$$\sum_{i=1}^n \left(\phi(\mathbf{x}_i) \sum_{j=1}^n a_j K(\mathbf{x}_i, \mathbf{x}_j) \right) = n \lambda_1 \sum_{i=1}^n a_i \phi(\mathbf{x}_i) \quad (21.10)$$

We have so far managed to replace one of the dot products with the kernel. To make sure that all computations in feature space are only in terms of dot products, we can take any point $\phi(\mathbf{x}_k)$, and multiply (21.10) by $\phi(\mathbf{x}_k)^T$ on both sides to obtain:

$$\sum_{i=1}^n \left(\phi(\mathbf{x}_k)^T \phi(\mathbf{x}_i) \sum_{j=1}^n a_j K(\mathbf{x}_i, \mathbf{x}_j) \right) = n \lambda_1 \sum_{i=1}^n a_i \phi(\mathbf{x}_k)^T \phi(\mathbf{x}_i) \quad (21.11)$$

$$\sum_{i=1}^n K(\mathbf{x}_k, \mathbf{x}_i) \sum_{j=1}^n a_j K(\mathbf{x}_i, \mathbf{x}_j) = n \lambda_1 \sum_{i=1}^n a_i K(\mathbf{x}_k, \mathbf{x}_i) \quad (21.12)$$

Let us define the *kernel matrix* as the matrix \mathbf{K} whose entries record all the kernel values for all pairs of points, given as:

$$\mathbf{K} = \begin{pmatrix} K(\mathbf{x}_1, \mathbf{x}_1) & K(\mathbf{x}_1, \mathbf{x}_2) & \cdots & K(\mathbf{x}_1, \mathbf{x}_n) \\ K(\mathbf{x}_2, \mathbf{x}_1) & K(\mathbf{x}_2, \mathbf{x}_2) & \cdots & K(\mathbf{x}_2, \mathbf{x}_n) \\ \vdots & \vdots & \vdots & \vdots \\ K(\mathbf{x}_n, \mathbf{x}_1) & K(\mathbf{x}_n, \mathbf{x}_2) & \cdots & K(\mathbf{x}_n, \mathbf{x}_n) \end{pmatrix} \quad (21.13)$$

Furthermore, let \mathbf{K}_i denote row i of the kernel matrix, written as the column vector:

$$\mathbf{K}_i = (K(\mathbf{x}_i, \mathbf{x}_1) \ K(\mathbf{x}_i, \mathbf{x}_2) \ \cdots \ K(\mathbf{x}_i, \mathbf{x}_n))^T \quad (21.14)$$

Let \mathbf{a} denote the column vector of weights:

$$\mathbf{a} = (a_1 \ a_2 \ \cdots \ a_n)^T \quad (21.15)$$

We can plug \mathbf{K}_i and \mathbf{a} into (21.12), and rewrite it as:

$$\sum_{i=1}^n K(\mathbf{x}_k, \mathbf{x}_i) \mathbf{K}_i^T \mathbf{a} = n\lambda_1 \mathbf{K}_k^T \mathbf{a} \quad (21.16)$$

In fact, since we can choose any of the n points, $\phi(\mathbf{x}_k)$, in the feature space, to obtain (21.12), we have a set of n equations:

$$\begin{aligned} \sum_{i=1}^n K(\mathbf{x}_1, \mathbf{x}_i) \mathbf{K}_i^T \mathbf{a} &= n\lambda_1 \mathbf{K}_1^T \mathbf{a} \\ \sum_{i=1}^n K(\mathbf{x}_2, \mathbf{x}_i) \mathbf{K}_i^T \mathbf{a} &= n\lambda_1 \mathbf{K}_2^T \mathbf{a} \\ &\vdots = \vdots \\ \sum_{i=1}^n K(\mathbf{x}_n, \mathbf{x}_i) \mathbf{K}_i^T \mathbf{a} &= n\lambda_1 \mathbf{K}_n^T \mathbf{a} \end{aligned}$$

We can compactly represent all of these n equations as follows:

$$\mathbf{K}^2 \mathbf{a} = n\lambda_1 \mathbf{K} \mathbf{a} \quad (21.17)$$

Multiplying by \mathbf{K}^{-1} on both sides, we obtain:

$$\boxed{\mathbf{K} \mathbf{a} = n\lambda_1 \mathbf{a}} \quad (21.18)$$

which we can recognize as an eigenvalue-eigenvector problem. In other words, the weight vector \mathbf{a} is the eigenvector corresponding to the largest eigenvalue $n\lambda_1$ of the kernel matrix \mathbf{K} .

Once \mathbf{a} is found, we can plug it back into (21.6) to obtain the first kernel principal component \mathbf{u}_1 . The only constraint we impose is that it has to be normalized to be a unit vector, as follows:

$$\mathbf{u}_1^T \mathbf{u}_1 = 1 \quad (21.19)$$

$$\sum_{i=1}^n \sum_{j=1}^n a_i a_j \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) = 1 \quad (21.20)$$

$$\mathbf{a}^T \mathbf{K} \mathbf{a} = 1 \quad (21.21)$$

Plugging in (21.18) into the equation above, we get:

$$\mathbf{a}^T \mathbf{K} \mathbf{a} = 1 \quad (21.22)$$

$$\mathbf{a}^T (n\lambda_1 \mathbf{a}) = 1 \quad (21.23)$$

$$n\lambda_1 \mathbf{a}^T \mathbf{a} = 1 \quad (21.24)$$

$$\|\mathbf{a}\|^2 = \frac{1}{n\lambda_1} \quad (21.25)$$

In other words, we have to scale the weight vector \mathbf{a} so that its norm is $\|\mathbf{a}\| = \sqrt{\frac{1}{n\lambda_1}}$.

Finally, we can project any point \mathbf{x}_i onto the principal direction, as follows:

$$\mathbf{u}_1^T \phi(\mathbf{x}_i) = \sum_{j=1}^n a_j \phi(\mathbf{x}_j)^T \phi(\mathbf{x}_i) = \sum_{j=1}^n a_j K(\mathbf{x}_j, \mathbf{x}_i) \quad (21.26)$$

Thus we have shown that all computations, either for the solution of the principal component, or for the projection of points, can be carried out using only the kernel function.

Finally, note that, as before, we can obtain the additional principal components by solving for the other eigenvalues, and eigenvectors of (21.18). In other words, if we sort the eigenvalues in decreasing order $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_r \geq 0$, we can obtain the j -th principal component as the corresponding eigenvector \mathbf{a}_j , which has to be normalized so that the norm is $\|\mathbf{a}_j\|^2 = \frac{1}{n\lambda_j}$, provided $\lambda_j > 0$.

21.1.1 Centering in Feature Space

In the analysis above, we assumed that the points were centered in feature space. This is not always the case, so we need to make sure that we can center the points using only the kernel function.

Let $\boldsymbol{\mu}_\phi$ denote the mean of the points in feature space, given as:

$$\boldsymbol{\mu}_\phi = \frac{1}{n} \sum_{i=1}^n \phi(\mathbf{x}_i) \quad (21.27)$$

We can center each point by subtracting the mean from it:

$$\hat{\phi}(\mathbf{x}_i) = \phi(\mathbf{x}_i) - \boldsymbol{\mu}_\phi \quad (21.28)$$

The centered kernel matrix is then given as

$$\hat{\mathbf{K}} = \{\hat{K}(\mathbf{x}_i, \mathbf{x}_j)\}_{i,j=1}^n \quad (21.29)$$

where

$$\hat{K}(\mathbf{x}_i, \mathbf{x}_j) = \hat{\phi}(\mathbf{x}_i)^T \hat{\phi}(\mathbf{x}_j) \quad (21.30)$$

$$= (\phi(\mathbf{x}_i) - \boldsymbol{\mu}_\phi)^T (\phi(\mathbf{x}_j) - \boldsymbol{\mu}_\phi) \quad (21.31)$$

$$= \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) - \phi(\mathbf{x}_i)^T \boldsymbol{\mu}_\phi - \phi(\mathbf{x}_j)^T \boldsymbol{\mu}_\phi + \boldsymbol{\mu}_\phi^T \boldsymbol{\mu}_\phi \quad (21.32)$$

$$= K(\mathbf{x}_i, \mathbf{x}_j) - \frac{1}{n} \sum_{k=1}^n \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_k) - \frac{1}{n} \sum_{k=1}^n \phi(\mathbf{x}_j)^T \phi(\mathbf{x}_k) + \frac{1}{n^2} \sum_{k=1}^n \sum_{l=1}^n \phi(\mathbf{x}_k)^T \phi(\mathbf{x}_l) \quad (21.33)$$

$$= K(\mathbf{x}_i, \mathbf{x}_j) - \frac{1}{n} \sum_{k=1}^n K(\mathbf{x}_i, \mathbf{x}_k) - \frac{1}{n} \sum_{k=1}^n K(\mathbf{x}_j, \mathbf{x}_k) + \frac{1}{n^2} \sum_{k=1}^n \sum_{l=1}^n K(\mathbf{x}_k, \mathbf{x}_l) \quad (21.34)$$

In other words, we can compute the centered kernel matrix using only the kernel function. Over all the pairs of points, we can write all n^2 entries in compact matrix notation as follows:

$$\hat{\mathbf{K}} = \mathbf{K} - \mathbf{1}_n \mathbf{K} - \mathbf{K} \mathbf{1}_n + \mathbf{1}_n \mathbf{K} \mathbf{1}_n \quad (21.35)$$

where $\mathbf{1}_n$ is a $n \times n$ matrix, all of whose entries equal $\frac{1}{n}$.

This way, we can use the centered matrix $\hat{\mathbf{K}}$ instead of \mathbf{K} for computing the eigenvector and eigenvalues for kernel PCA. In other words, we replace (21.18) with:

$$\boxed{\hat{\mathbf{K}} \mathbf{a} = n\lambda_1 \mathbf{a}} \quad (21.36)$$

21.2 Kernel Discriminant Analysis

Kernel PCA ignores the class values for each point and finds the orthogonal directions of most variance. Kernel discriminant analysis, like linear discriminant analysis (LDA), tries to find a direction that maximizes the separation between the classes in feature space.

We assume that we have two classes, and let the dataset be partitioned into two parts: $\mathcal{D}_1 = \{\phi(\mathbf{x}_i) | y_i = +1\}$ and $\mathcal{D}_2 = \{\phi(\mathbf{x}_i) | y_i = -1\}$. Further, let $n_1 = |\mathcal{D}_1|$ and $n_2 = |\mathcal{D}_2|$. The goal is to find the direction vector \mathbf{w} in feature space, with unit magnitude ($\|\mathbf{w}\| = 1$), so as to maximize (see (19.20) in Chapter 19):

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}} \quad (21.37)$$

The between class scatter matrix \mathbf{S}_B is given as:

$$\mathbf{S}_B = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \quad (21.38)$$

where $\boldsymbol{\mu}_i$ is the mean of class \mathcal{D}_i in feature space

$$\boldsymbol{\mu}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \phi(\mathbf{x}_j) \quad (21.39)$$

The within class scatter matrix \mathbf{S}_W in feature space, is given as:

$$\mathbf{S}_W = \mathbf{S}_1 + \mathbf{S}_2 \quad (21.40)$$

where \mathbf{S}_i is the scatter matrix for class i :

$$\mathbf{S}_i = \sum_{\mathbf{x}_j \in \mathcal{D}_i} (\phi(\mathbf{x}_j) - \boldsymbol{\mu}_i)(\phi(\mathbf{x}_j) - \boldsymbol{\mu}_i)^T \quad (21.41)$$

Finally, the direction vector is obtained as the eigenvector corresponding to the largest eigenvalue of the following equation:

$$(\mathbf{S}_W^{-1} \mathbf{S}_B) \mathbf{w} = \lambda_1 \mathbf{w} \quad (21.42)$$

Our goal is to show that all these computations can be done via only dot products in the feature space, which can then be replaced by the kernel function between pairs of points in input space.

First, it can be shown that the direction vector \mathbf{w} can be expressed as a linear combination of the points in feature space. In other words, we can express \mathbf{w} as follows:

$$\mathbf{w} = \sum_{i=1}^n a_i \phi(\mathbf{x}_i) \quad (21.43)$$

The projected mean, $\mathbf{w}^T \boldsymbol{\mu}_i$, can be written as:

$$\mathbf{w}^T \boldsymbol{\mu}_i = \left(\sum_{j=1}^n a_j \phi(\mathbf{x}_j) \right) \left(\frac{1}{n_i} \sum_{\mathbf{x}_k \in \mathcal{D}_i} \phi(\mathbf{x}_k) \right) \quad (21.44)$$

$$= \frac{1}{n_i} \sum_{j=1}^n \sum_{\mathbf{x}_k \in \mathcal{D}_i} a_j \phi(\mathbf{x}_j)^T \phi(\mathbf{x}_k) \quad (21.45)$$

$$= \frac{1}{n_i} \sum_{j=1}^n \sum_{\mathbf{x}_k \in \mathcal{D}_i} a_j K(\mathbf{x}_j, \mathbf{x}_k) \quad (21.46)$$

$$= \mathbf{a}^T \mathbf{m}_i \quad (21.47)$$

where $\mathbf{a} = (a_1, a_2, \dots, a_n)^T$ is the weight vector, and

$$\mathbf{m}_i = \frac{1}{n_i} \begin{pmatrix} \sum_{\mathbf{x}_k \in \mathcal{D}_i} K(\mathbf{x}_1, \mathbf{x}_k) \\ \sum_{\mathbf{x}_k \in \mathcal{D}_i} K(\mathbf{x}_2, \mathbf{x}_k) \\ \vdots \\ \sum_{\mathbf{x}_k \in \mathcal{D}_i} K(\mathbf{x}_n, \mathbf{x}_k) \end{pmatrix} \quad (21.48)$$

In other words, \mathbf{m}_i is a vector that stores the sum of the kernel values from elements in class c_i to all other points.

Thus we can re-write the separation between the projected means as follows:

$$|m_1 - m_2|^2 = |\mathbf{w}^T \boldsymbol{\mu}_1 - \mathbf{w}^T \boldsymbol{\mu}_2|^2 \quad (21.49)$$

$$= |\mathbf{a}^T \mathbf{m}_1 - \mathbf{a}^T \mathbf{m}_2|^2 \quad (21.50)$$

$$= \mathbf{a}^T (\mathbf{m}_1 - \mathbf{m}_2) (\mathbf{m}_1 - \mathbf{m}_2)^T \mathbf{a} \quad (21.51)$$

$$= \mathbf{a}^T \mathbf{M} \mathbf{a} \quad (21.52)$$

where $\mathbf{M} = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T$. Thus we obtain

$$\mathbf{w}^T \mathbf{S}_B \mathbf{w} = |m_1 - m_2|^2 = \mathbf{a}^T \mathbf{M} \mathbf{a} \quad (21.53)$$

In a similar manner, we can compute the within class scatter matrix, by computing the projected variance for each class s_1^2 and s_2^2 , purely in terms of the kernel function, to obtain:

$$\mathbf{w}^T \mathbf{S}_W \mathbf{w} = s_1^2 + s_2^2 = \mathbf{a}^T \mathbf{N} \mathbf{a} = \mathbf{a}^T (\mathbf{N}_1 + \mathbf{N}_2) \mathbf{a} \quad (21.54)$$

where $\mathbf{N}_i = \mathbf{E}_i (\mathbf{I} - \mathbf{1}_{n_i}) \mathbf{E}_i^T$, and \mathbf{E}_i is a $n \times n_i$ kernel matrix restricted to class i , given as $\mathbf{E}_i(j, k) = K(\mathbf{x}_j, \mathbf{x}_k)$ for all $j \in [1, n]$ and all $k \in [1, n_i]$. \mathbf{I} is the $n_i \times n_i$ identity matrix, and $\mathbf{1}_{n_i}$ is a $n_i \times n_i$ matrix all of whose entries are $\frac{1}{n_i}$.

Combining (21.53) and (21.54), we obtain the new maximization condition:

$$\boxed{J(\mathbf{a}) = \frac{\mathbf{a}^T \mathbf{M} \mathbf{a}}{\mathbf{a}^T \mathbf{N} \mathbf{a}}} \quad (21.55)$$

Notice how all the terms in the expression above only contain kernel functions. We can now obtain the weight vector \mathbf{a} as the eigenvector corresponding to the largest eigenvector for the equation:

$$\boxed{(\mathbf{N}^{-1} \mathbf{M}) \boldsymbol{\alpha} = \lambda_1 \boldsymbol{\alpha}} \quad (21.56)$$

Once \mathbf{a} has been obtained, we make sure that we normalize \mathbf{w} to be a unit vector, based on (21.43), as follows:

$$\mathbf{w}^T \mathbf{w} = 1 \quad (21.57)$$

$$\sum_{i=1}^n \sum_{j=1}^n a_i a_j \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) = 1 \quad (21.58)$$

$$\mathbf{a}^T \mathbf{K} \mathbf{a} = 1 \quad (21.59)$$

$$(21.60)$$

Finally, we can project any point \mathbf{x}_i onto the discriminant direction, as follows:

$$\mathbf{w}^T \phi(\mathbf{x}_i) = \sum_{j=1}^n a_j \phi(\mathbf{x}_j)^T \phi(\mathbf{x}_i) = \sum_{j=1}^n a_j K(\mathbf{x}_j, \mathbf{x}_i) \quad (21.61)$$