**STAT 535: Statistical Machine Learning** **Winter 2019**

# Lecture 6: SVM, PCA, and Kernel methods

*Instructor: Yen-Chi Chen*

In this lecture, we will study two problems: the support vector machine (SVM) and the principle component analysis (PCA). Both methods can be *kernelized* using the reproducing kernel Hilbert space (RKHS). We will see that the key insight of kernelization is to replace the inner product by a kernel inner product. We start with SVM.

## 6.1 Support vector machine

Consider the binary classification problem that our observations are

$$(X_1, Y_1), \cdots, (X_n, Y_n)$$

such that $X_i \in \mathbb{R}^d$ and $Y_i$ are binary. To simplify the classification problem, instead of using binary representation of the two classes, we use $Y_i \in \{+1, -1\}$ to indicate the two classes.

SVM is one of the most popular classification tools that machine learning experts were using before the invention of deep neural nets. SVM is motivated by constructing the 'best' linear classifier that separates two classes when the two classes are indeed separable by a linear classifier (in this case, we call the problem linearly separable). Two classes are *linearly separable* if there exists one hyperplane $H(x) = b^T x + a$ such that the sign of $H$ perfectly separate the two classes, i.e,

$$Y_i \cdot H(X_i) = Y_i(b^T X_i + a) > 0$$

for all $i = 1, \cdots, n$.

Given a linear classifier $H(x) = b^T x + a$, the distance from observation $(X_i, Y_i)$ to the decision boundary $H(x) = 0 = b^T x + a$ is

$$\frac{1}{\|b\|} Y_i(b^T X_i + a).$$

Thus, the *margin*–minimal distance to the decision boundary–is

$$\frac{1}{\|b\|} \min_i Y_i(b^T X_i + a).$$

The SVM attempts to find the classifier that maximizes the margin, i.e, the SVM tries to solve

$$\mathsf{argmax}_{b,a} \quad \frac{1}{\|b\|} \min_i Y_i(b^T X_i + a). \tag{6.1}$$

Because the classifier is unchanged when we rescale $(a, b)$ by $(\kappa a, \kappa b)$, we add an constraint

$$Y_i(b^T X_i + a) \geq 1 \quad \text{for all } i = 1, \cdots, n \tag{6.2}$$

and there exists at least one observation such that the equality holds. Thus, the minimization of margin becomes maximizing $\frac{1}{\|b\|}$ subject to the constraint in equation (6.2) so the problem in equation (6.1) is equivalently to solving

$$\mathsf{argmax}_{b,a} \quad \frac{1}{\|b\|} \quad \text{subject to equation (6.2)}$$

or equivalently,

$$\mathsf{argmin}_{b,a} \quad \frac{1}{2}\|b\|^2 \quad \text{subject to equation (6.2).} \tag{6.3}$$

Note that we add the additional $\frac{1}{2}$ to simplify later derivation.

Using the Lagrangian multiplier, we can derive the dual form of the problem in equation (6.3) as minimizing the following Lagrangian:

$$L(a,b,\lambda) = \frac{1}{2}\|b\|^2 - \sum_{i=1}^{n} \lambda_i [Y_i(b^T X_i + a) - 1], \tag{6.4}$$

where $\lambda_1, \cdots, \lambda_n$ are the Lagrangian multiplier. To simplify the problem, we consider taking derivatives of $L$ with respect to $a$ and $b$ first, which leads to

$$\begin{aligned} \frac{\partial L}{\partial b} &= 0 \Rightarrow b = \sum_{i=1}^{n} \lambda_i Y_i X_i \\ \frac{\partial L}{\partial a} &= 0 \Rightarrow 0 = \sum_{i=1}^{n} \lambda_i Y_i \end{aligned} . \tag{6.5}$$

Plugging the above two equations into equation (6.4), we obtain

$$\begin{aligned} L_1(\lambda) &= \frac{1}{2}\|\sum_{i=1}^{n} \lambda_i Y_i X_i\|^2 - \sum_{i=1}^{n} \lambda_j [Y_j((\sum_{i=1}^{n} \lambda_i Y_i X_i)^T X_j + a) - 1] \\ &= \frac{1}{2}\|\sum_{i=1}^{n} \lambda_i Y_i X_i\|^2 - \|\sum_{i=1}^{n} \lambda_i Y_i X_i\|^2 + \sum_{i=1}^{n} \lambda_i \\ &= \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j Y_i Y_j X_i^T X_j \\ &= \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j Y_i Y_j \langle X_i, X_j \rangle, \end{aligned} \tag{6.6}$$

where we use $\langle \cdot, \cdot \rangle$ to denote inner product. Thus, the SVM problem becomes the above quadratic minimization problem– we will find $\lambda_1, \cdots, \lambda_n$ such that $L_1(\lambda)$ is minimized.

It turns out that many $\lambda_i$'s will have a value of 0. From Lagrangian multiplier theory, we know that when $\lambda_i = 0$, the constraint $Y_i(b^T X_i + a) - 1 = 0$ does not hold, i.e, the pair $(X_i, Y_i)$ is not the observation that is closest to the decision boundary (so they are not on the margin). Those observation with $\lambda_i > 0$ are the ones that determines the margin so we will call them the *support vectors*.

### 6.1.1 Not linearly separable case

The above analysis is under the assumption that the data is linearly separable. When the data is not linearly separable, the constraint in equation (6.2)

$$Y_i(b^T X_i + a) \geq 1$$

is too strict. We have to relax it.

One approach to relax such constraint is to introduce a set of *slack variables* $\xi_1, \cdots, \xi_n \geq 0$ and replace the above constraint by

$$Y_i(b^T X_i + a) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \text{for all } i = 1, \cdots, n. \tag{6.7}$$

If we attempt to optimize equation (6.3) with respect to the above constraint, the we can always choose $\xi_i$ to be as large as possible and it is bad for solving the problem. To avoid choose a large value of $\xi_i$'s, we modify the optimization problem in equation (6.4)

$$\mathsf{argmin}_{b,a,\xi} \quad C\sum_{i=1}^n \xi_i + \frac{1}{2}\|b\|^2 \quad \text{subject to equation (6.7)}, \tag{6.8}$$

where $C > 0$ behaves like a penalization parameter (later we will show this). After solving this problem, the value of $\xi_i$ falls into three interesting regimes

$$\xi_i \begin{cases} = 0, & \text{if } i\text{-th observation is correctly classified and away from the boundary,} \\ \in (0,1], & \text{if } i\text{-th observation is correctly classified but close to the boundary,} \\ > 1, & \text{if } i\text{-th observation is mis-classified.} \end{cases} \tag{6.9}$$

Solving equation (6.8) is also a challenging task so we consider the dual problem of it. Using Lagrangian multipliers, we obtain the Lagrangian form of it as

$$L_3(a,b,\lambda,\xi,\mu) = C\sum_{i=1}^n \xi_i + \frac{1}{2}\|b\|^2 - \sum_{i=1}^n \lambda_i[Y_i(b^TX_i + a) - 1 - \xi_i] - \sum_{i=1}^n \mu_i\xi_i, \tag{6.10}$$

where $\lambda$ and $\mu$ are Lagrangian multipliers. Again, to simplify the problem we take derivatives, which leads to

$$\begin{aligned} \frac{\partial L}{\partial b} &= 0 \Rightarrow b = \sum_{i=1}^n \lambda_i Y_i X_i \\ \frac{\partial L}{\partial a} &= 0 \Rightarrow 0 = \sum_{i=1}^n \lambda_i Y_i \quad . \\ \frac{\partial L}{\partial \xi_i} &= 0 \Rightarrow C = \lambda_i + \mu_i \end{aligned} \tag{6.11}$$

Among all of them, the last constraint is very informative– because $\mu_i \geq 0$, we obtain a constraint $\lambda_i \leq C$.

Plugging equation (6.11) into equation (6.10), we obtain an interesting result:

$$L_4(\lambda,\mu) = \sum_{i=1}^n \lambda_i - \frac{1}{2}\sum_{i=1}^n\sum_{j=1}^n \lambda_i\lambda_j Y_i Y_j \langle X_i, X_j\rangle. \tag{6.12}$$

Namely, the objective function is the same as the linearly separable case. But the minimization of equation (6.12) has a slightly different constraints:

$$0 \leq \lambda_i \leq C, \quad \text{for all } i = 1, \cdots n. \tag{6.13}$$

The upper bound on $\lambda_i$'s is due to the penalty on the slack variables in the objective function.

Another way to view the SVM is to study the slack variable $\xi_i$. We can rearrange equation (6.7) to obtain

$$\xi_i \geq 1 - Y_i(b^TX_i + a), \quad \xi_i \geq 0,$$

which can be re-expressed as

$$\xi_i \geq [1 - Y_i(b^TX_i + a)]_+,$$

where $[x]_+ = \max\{x, 0\}$. Since in equation (6.8), we are trying to minimize $\xi_i$, it is easy to see that at the optima, we will always choose

$$\xi_i = [1 - Y_i(b^TX_i + a)]_+.$$

Thus, equation (6.8) can be rewritten as

$$\begin{aligned}&\mathsf{argmin}_{b,a,\xi} \quad C\sum_{i=1}^{n}[1-Y_i(b^TX_i+a)]_+ + \frac{1}{2}\|b\|^2\\ \Leftrightarrow\ &\mathsf{argmin}_{b,a,\xi} \quad \sum_{i=1}^{n}[1-Y_i(b^TX_i+a)]_+ + \frac{1}{2C}\|b\|^2.\end{aligned} \tag{6.14}$$

The quantity $[1-Y_i(b^TX_i+a)]_+$ is called the *hinge loss* $L(a,b)=(1-ab)_+$. So the SVM is to minimize a regularized hinge loss with squared penalty.

**Remark (KKT conditions).** In the above analysis, we are moving between a constrained form and its dual Lagrangian form. One sufficient condition that make the optima from the two problems agree is the *Karush-Kuhn-Tucker (KKT) conditions* hold. It is a very important topic in optimization and I would highly recommend you to look for related materials[1].

## 6.2 Principle component analysis

The principle component analysis (PCA) is a common multivariate analysis approach to discover the hidden structure within the data. It is also used to discover the hidden subspace within the data.

Suppose that we observed $X_1,\cdots,X_n\in\mathbb{R}^d$ and without loss of generality, we assume that the data is centered, i.e, the sample average is 0. Let

$$\widehat{\Sigma}_n=\frac{1}{n}\sum_{i=1}^{n}X_iX_i^T\in\mathbb{R}^d$$

be the sample covariance matrix. Since the covariance matrix is a positive semi-definite matrix, it has well-defined eigenvalues and eigenvectors. Let $\lambda_1\geq\lambda_2\geq\cdots\lambda_d$ be the ordered eigenvalues and $v_1,\cdots,v_d$ be the corresponding eigenvectors. The eigenvectors of the covariance matrix are called the *principle vectors* of the data. In particular, $v_1$ will be called the first principle vector and $v_2$ will be called the second principle vector.

Why are these principle vectors important? Now suppose that we make a linear combination of the variables such that $Z_i(\mu)=X_i^T\mu$, where $\|\mu\|=1$. This creates a set of new observations $Z_1,\cdots,Z_n$. Which choice of $\mu$ will make $Z_i$ to have the highest sample variance? i.e., we want to find

$$\mathsf{argmax}_{\mu:\|\mu\|=1}\frac{1}{n}\sum_{i=1}^{n}Z_i^2(\mu).$$

You can prove that the solution to the above maximization is $v_1$, the first principle vector and $\lambda_1$ will be the maximal variance. The second principle vector can be defined in a similar way but with the constraint that the vector $\mu$ has to be orthogonal to $v_1$.

In fact, the spectral decomposition of $\Sigma_n$ shows a very interesting property of principle vectors:

$$\widehat{\Sigma}_n=\sum_{j=1}^{d}\lambda_jv_jv_j^T.$$

Thus, the covariance matrix can be viewed as the summation of covariance from each eigenvector multiply by the eigenvalue. So an eigenvalue represents the contribution of covariance from the corresponding eigenvector.

[1] A good reference: https://www.cs.cmu.edu/~ggordon/10725-F12/slides/16-kkt.pdf

PCA is often used to perform *dimension reduction*. Suppose that the original data has $d$ variables and we would like to reduce the dimension to $s < d$ by a linear transformation of the original variables. Namely, we want to find the matrix $Q \in \mathbb{R}^{d\times s}$ such that observations in the reduced dimension $S_i(Q) = Q^T X_i$ are similar to the original data in the following sense. We want to find $Q$ such that the trace of the covariance matrix of $S_1, \cdots, S_n$ is minimized, i.e.,

$$\mathsf{argmin}_Q \quad \mathsf{Tr}(\widehat{\Sigma}_S(Q)),$$

where $\widehat{\Sigma}_S(Q) = \frac{1}{n}\sum_{i=1}^n S_i(Q)S_i(Q)^T$. Namely, we want to find the best $s$ linear subspace such that the data has the least residual variance.

## 6.3 Kernel methods

**WARNING:** the kernel method (also known as the kernel trick) is using a different idea of *kernel* from the kernel function used in density estimation and regression.

The kernel method is a feature mapping approach that we convert the original features/covariates into a new sets of features and perform statistical inference. The idea is very simple, given a feature $x \in \mathcal{X} \subset \mathbb{R}^d$, we construct a (non-linear) mapping $\phi(x) \in \mathbb{R}^m$ and then use $\phi(x)$ as our new features to analyze data.

The power of kernel method is that we do not need to explicitly compute the new feature $\phi(x)$. We can compute the inner product $\langle\phi(x), \phi(y)\rangle = \sum_{j=1}^m \phi_j(x)\phi_j(y) = K(x, y)$ without evaluating the new feature. So this even allows the dimension of the new feature $m$ to be infinite. Namely, suppose that we want to compute the inner product between two transformed observations $\phi(X_i), \phi(X_j)$, we just need to compute

$$K(X_i, X_j) = \langle\phi(X_i), \phi(X_j)\rangle.$$

There is no need to compute $\phi(X_i)$ and $\phi(X_j)$. One scenario that $m$ is infinite is the case where the new feature $\phi(x) = \eta_x$ is a function (in $L_2$ space). In this case, the inner product can be defined as

$$\langle\phi(x), \phi(y)\rangle = \int \eta_x(z)\eta_y(z)dz = K(x, y).$$

As you can see, as long as we use a good feature mapping $\phi$, the computation of the inner product is just evaluating the kernel.

It turns out that the kernel methods reverse this direction–we specify the kernel function first and then the kernel function implies a set of new features $\phi(x)$. But what kernel function $K$ will implies such a good inner product property? The **Mercer's theorem** shows that roughly speaking, when the kernel function $K$ is positive semi-definite, i.e.,

$$\int\int K(x, y)f(x)f(y)dxdy \geq 0$$

for all $f \in L_2(\mathcal{X})$, such $\phi$ that corresponds to $K$ exists. Therefore, we just need to specify a good kernel function and then the feature mapping will be determined automatically. Here are some commonly-used kernel function:

- *Polynomials.* $K(x, y) = (\langle x, y\rangle + a)^r$, where $a, r$ are tuning parameters.
- *Sigmoid.* $K(x, y) = \tanh(a\langle x, y\rangle + b)$, where $a, b$ are tuning parameters.
- *Gaussian.* $K(x, y) = \exp\left(-\frac{1}{2\sigma^2}\|x - y\|^2\right)$, where $\sigma^2$ is a tuning parameter.

### 6.3.1 Kernel SVM

Kernelizing the SVM is very straight forward. Recall that we need to optimize equation (6.12) to find the SVM solution:

$$L_4(\lambda,\mu)=\sum_{i=1}^n\lambda_i-\frac{1}{2}\sum_{i=1}^n\sum_{j=1}^n\lambda_i\lambda_jY_iY_j\langle X_i,X_j\rangle.$$

We simply replace $\langle X_i,X_j\rangle$ by the kernel $K(X_i,X_j)$, leading to

$$L_4(\lambda,\mu;K)=\sum_{i=1}^n\lambda_i-\frac{1}{2}\sum_{i=1}^n\sum_{j=1}^n\lambda_i\lambda_jY_iY_jK(X_i,X_j).$$

Then we can optimize the above with respect to $\lambda$ and the constraint that $0\leq\lambda_j\leq C$.

### 6.3.2 Kernel PCA

PCA is another scenario that is commonly kernelized. It is often called the *kernel PCA*. The kernelization of the PCA is not as simple as the SVM since it does not directly involve an inner product. Recall that the covariance matrix

$$\widehat{\Sigma}_n=\frac{1}{n}\sum_{i=1}^nX_iX_i^T.$$

Suppose that we perform a feature mapping that maps $X_i$ into $\phi(X_i)\in\mathbb{R}^m$, then the covariance matrix becomes

$$\widehat{\Sigma}_\phi=\frac{1}{n}\sum_{i=1}^n\phi(X_i)\phi(X_i)^T\in\mathbb{R}^{m\times m}.$$

Let $(\lambda_\ell,v_\ell)$ be the $\ell$-th eigenvalue/vector pair of $\widehat{\Sigma}_\phi$, i.e.,

$$\widehat{\Sigma}_\phi v_\ell=\lambda_\ell v_\ell=\frac{1}{n}\sum_{i=1}^n\phi(X_i)\phi(X_i)^Tv_\ell. \tag{6.15}$$

Recall that in the PCA, the key is to find the eigenvector/value pairs $(\lambda_\ell,v_\ell)$ for each $\ell=1,\cdots,m$. In what follows, we will show that we do not need to compute $\phi(X_i)$ at all to obtain the eigenstructures of $\widehat{\Sigma}_\phi$.

Because of the property of the kernel function (later we will introduce this–a property of the RKHS), we can express the eigenvector

$$v_\ell=\sum_{i=1}^na_{i\ell}\phi(X_i)\in\mathbb{R}^m.$$

With this, we can rewrite equation (6.15) as

$$\begin{aligned}
\lambda_\ell \sum_{i=1}^n a_{i\ell}\phi(X_i) &= \lambda_\ell v_\ell \\
&= \widehat{\Sigma}_\phi v_\ell \\
&= \frac{1}{n}\sum_{i=1}^n \phi(X_i)\phi(X_i)^T \left(\sum_{j=1}^n a_{j\ell}\phi(X_j)\right) \\
&= \frac{1}{n}\sum_{i,j=1}^n \phi(X_i)a_{j\ell}\phi(X_i)^T\phi(X_j) \\
&= \frac{1}{n}\sum_{i,j=1}^n \phi(X_i)a_{j\ell}K(X_i,X_j) \\
&= \frac{1}{n}\sum_{i,j=1}^n a_{j\ell}\phi(X_j)K(X_i,X_j).
\end{aligned}$$

Inner product the above with $\phi(X_k)$, we obtain

$$\lambda_\ell \sum_{i=1}^n a_{i\ell}\phi(X_k)^T\phi(X_i) = \frac{1}{n}\sum_{i,j=1}^n a_{j\ell}\phi(X_k)^T\phi(X_j)K(X_i,X_j)$$

or equivalently,

$$\lambda_\ell \sum_{i=1}^n a_{i\ell}K(X_k,X_i) = \frac{1}{n}\sum_{i,j=1}^n a_{j\ell}K(X_k,X_j)K(X_i,X_j). \tag{6.16}$$

Let $a_\ell = (a_{1\ell},\cdots,a_{n,\ell}) \in \mathbb{R}^n$ and $\mathbf{K} = [K(X_i,X_j)] \in \mathbb{R}^{n\times n}$. The vector $a_\ell$ can be viewed as the coefficient vector that is of interest. The we can rewrite equation (6.16) as

$$\lambda_\ell \mathbf{K} a_\ell = \frac{1}{n}\mathbf{K}^2 a_\ell$$

or equivalently,

$$\lambda_\ell a_\ell = \frac{1}{n}\mathbf{K} a_\ell.$$

Thus, the coefficient vector $a_\ell$ is the eigenvector of $\mathbf{K}$ and $\lambda_\ell$ will be the corresponding eigenvalue divided by $n$.

So the eigenvectors (and eigenvalues) of $\widehat{\Sigma}_\phi$ can be obtained by using the eigenvectors (and eigenvalues) of $\mathbf{K}$. The matrix $\mathbf{K}$ can be computed without explicitly finding the feature mapping $\phi$.

### 6.3.3 RKHS: reproducing kernel Hilbert space

The kernel methods rely on the fact that when the kernel function $K(x,y)$ has nice properties, it implies a useful feature mapping that avoid explicitly computing the feature. Now we take a deeper look at the kernel function $K(x,y)$ and the inner product structure. We begin with an interesting property called the *reproducing property*.

To simplify the problem, we consider the case where $\phi(x)$ is an infinite dimensional object so we write it as $\phi(x) = \eta_x(\cdot)$, where the input value $x$ is now an index (or a parameter) that is fixed and the feature is a function with the index $x$.

To see how the reproducing property works, consider the Dirac delta function $\delta_y(x) = \delta_0(y-x) = \delta_0(x,y)$ which put a probability mass 1 at location $y$. Then for any function $f(x)$, the integral

$$\int \delta_0(x,y) f(x) dx = \langle \delta_0(\cdot, y), f(\cdot)\rangle = f(y),$$

where the inner product here is defined using the usual inner product between two functions in $L_2$ space. So we insert a function $f(x)$ and then get back the same function $f(y)$ (with a different argument). The reproducing property refers to functions $K(x,y)$ that behave like $\delta_0(x,y)$.

The reproducing property can be generalized to other Hilbert space. In particular, we are interested in the reproducing kernel Hilbert space (RKHS), which is defined as follows.

**Definition 6.1** *For a compact subset $\mathcal{X} \subset \mathbb{R}^d$ and a Hilbert space $\mathcal{H}$ of functions $f : \mathcal{X} \mapsto \mathbb{R}$, we say that $\mathcal{H}$ is an RKHS if there exists $K : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ such that*

- *$K$'s span is dense in $\mathcal{H}$, i.e., $\mathcal{H} = \overline{\text{span}\{K(\cdot, x) : x \in \mathcal{X}\}}$.*
- *$K$ is a reproducing kernel, i.e, $\langle K(x,\cdot), f(\cdot)\rangle_{\mathcal{H}} = f(x)$ for any $f \in \mathcal{H}$, the inner product is defined below.*

Let $g$ and $h \in \mathcal{H}$ be such that

$$g(\cdot) = \sum_{i=1}^{M} \alpha_i K(\cdot, x_i), \quad h(\cdot) = \sum_{j=1}^{N} \beta_j K(\cdot, x_j).$$

We define the inner product to be

$$\langle g, h\rangle_{\mathcal{H}} = \sum_{i,j} \alpha_i \beta_j K(x_i, x_j).$$

It is easy to see that for an RKHS,

$$\langle K(x,\cdot), K(y,\cdot)\rangle_{\mathcal{H}} = K(x,y).$$

Thus, when we choose the feature map $\phi(x) = \eta_x(\cdot) = K(x,\cdot)$, then

$$\langle \phi(x), \phi(y)\rangle_{\mathcal{H}} = \langle K(x,\cdot), K(y,\cdot)\rangle_{\mathcal{H}} = K(x,y).$$

So the kernel methods construct the mapping to map $x$ into $K(x,\cdot)$ and convert $\mathcal{X}$ into $\mathcal{H}$.

The reproducing property can be viewed as a property of *evaluation functional.* This makes the RKHS different from the usual Hilbert space. The evaluation function is the functional $T_x(f) = f(x)$. In RKHS, $T_x(f) = \langle f, K(x,\cdot)\rangle_{\mathcal{H}}$. Roughly speaking, in a general Hilbert space, the evaluation functional is not continuous. To see this, consider the Dirac delta function in the $L_2$ space. It is easy to see that $\langle f, \delta_x\rangle = f(x)$. Now we consider a sequence of functions $f_n(x) = \sqrt{n} I(\|x\| \leq 1/n^2)$ it is easy to see that this function converges in $L_2$ distance to $f_0(x) = 0$. More explicitly,

$$\|f_n - f_0\|_2 = \int n I(\|x\| \leq 1/n^2) dx = \frac{2}{n} \to 0.$$

However, after applying the evaluation function, you can see that

$$\langle f_n, \delta_x\rangle = f_n(x) = \sqrt{n} \not\to \langle f_0, \delta_x\rangle = f_0(x)$$

at $x = 0$. Thus, the evaluation functional is not continuous. On the other hand, you can show that in RKHS, the evaluation functional is always continuous. In fact, *a Hilbert space is an RKHS if and only if the evaluation function is continuous.*

Recall that we mentioned the Mercer's theorem previously. Here is how the theorem goes. Let $T_K$ be a linear operator such that for $f \in L_2(\mathcal{X})$, $T_K(f)(x) = \int K(x,y)f(y)dy$.

**Theorem 6.2 (Mercer's theorem)** *Assume that $K$ is a continuous symmetric positive semi-definite kernel over $\mathcal{X} \times \mathcal{X}$, where $\mathcal{X}$ is compact. Then there exists an orthonormal basis $\{e_i(\cdot) : i = 1, \cdots, \}$ of $L_2(\mathcal{X})$ consisting of eigenfunctions of $T_K$ such that*

$$K(x,y) = \sum_{i=1}^{n} \lambda_i e_i(x) e_i(y),$$

*where $\lambda_i \geq 0$ are the corresponding eigenvalues.*

It also implies another representation under the regular $L_2$ space:

$$\phi(x) = (\sqrt{\lambda_1} e_1(x), \sqrt{\lambda_2} e_2(x), \cdots).$$

The quantities $\lambda_i$ and $e_i(x)$ are from Theorem 6.2.

For any functions $f, g \in \mathcal{H}$, we can expand them by the kernel function $K(\cdot,\cdot)$ or the basis $\{e_i(\cdot)\}$:

$$f(\cdot) = \sum_{i=1}^{M} \alpha_i K(\cdot, x_i) = \sum_{k=1}^{\infty} a_k e_k(\cdot), \quad g(\cdot) = \sum_{j=1}^{N} \beta_j K(\cdot, x_j) = \sum_{k=1}^{\infty} b_k e_k(\cdot).$$

Then we have

$$\langle f, g \rangle_{\mathcal{H}} = \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j), \quad \langle f, g \rangle = \sum_{k=1} a_k b_k.$$

**These two inner products can be different!** In fact, when using the RKHS inner product, we can still express the inner product in terms of orthornormal basis but with the following forms:

$$\langle f, g \rangle_{\mathcal{H}} = \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j) = \sum_{k=1} \frac{a_k b_k}{\lambda_k}.$$

**Example (two inner products are different).** Here is one example that shows how the two inner products are different. Consider the $1D$ Gaussian kernel $K(x,y) = \exp(-\frac{1}{2}|x-y|^2)$ and two functions $f(x) = \exp(-\frac{1}{2}|x-1|^2) = K(x,1)$ and $g(x) = \exp(-\frac{1}{2}|x+1|^2) = K(x,-1)$. Then the inner product under RKHS is

$$\langle f, g \rangle_{\mathcal{H}} = K(1,-1) = \exp(-2)$$

but the inner product under $L_2$ space is

$$\langle f, g \rangle = \int \exp(-\frac{1}{2}|x-1|^2 - -\frac{1}{2}|x+1|^2)dx = \int \exp(-x^2 - 1)dx = \sqrt{\pi}\exp(-1).$$

Note that there are 4 equivalent conditions for continuous, symmetric $K$ defined on compact set $\mathcal{X}$:

(K1) Every Gram matrix (i.e., $\mathbf{K} = [K(x_i, x_j)] \in \mathbb{R}^{n \times n}$ for any $x_1, \cdots, x_n$) is positive semi-definite.

(K2) The operator $T_K$ is positive semi-definite.

(K3) The kernel function $K$ can be expressed as $K(x,y) = \sum_{i=1}^{\infty} \lambda_i e_i(x) e_i(y)$.

(K4) $K$ is a reproducing kernel of an RKHS over $\mathcal{X}$.

The fact that (K3) implies (K4) shows that Theorem 6.2 implies that we can use $\phi(x) = K(x,\cdot)$ so the feature map can always be interpreted as $K(x,\cdot)$.

The RKHS generalizes the kernel methods to a much wider class of problems. The following representer theorem[2] shows that it can be used in regression and classification problems. Suppose that we observe random variables

$$(X_1, Y_1), \cdots, (X_n, Y_n).$$

**Theorem 6.3 (Representer Theorem (Kimeldorf and Wahba 1971))** *Let $\mathcal{H}$ be an RKHS with a kernel $K$ over a compact set $\mathcal{X}$. Let $L : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}$ be a loss function and $g$ be a strictly increasing function. For any solution $\widehat{f}_n$ to the following minimization problem*

$$\widehat{f}_n = \operatorname*{argmin}_{f \in \mathcal{H}} \sum_{i=1}^{n} L(Y_i, f(X_i)) + g(\|f\|_{\mathcal{H}}),$$

*we can express it as*

$$\widehat{f}_n(\cdot) = \sum_{i=1}^{n} \widehat{\alpha}_i K(X_i, \cdot),$$

*for some coefficients $\widehat{\alpha}_1, \cdots, \widehat{\alpha}_n$.*

The significance of the above theorem is that when we are minimizing the regularized loss, we only need to use the kernel function evaluated at observations to represent the final estimator. There is no need to use an expansion with infinite number of kernel basis. The above theorem can be found in the following two papers (the first one is the original version but the second one has an elegant form and proof):

> Kimeldorf, G., & Wahba, G. (1971). Some results on Tchebycheffian spline functions. *Journal of mathematical analysis and applications*, 33(1), 82-95.
> Schölkopf, B., Herbrich, R., & Smola, A. J. (2001, July). A generalized representer theorem. *In International conference on computational learning theory* (pp. 416-426). Springer, Berlin, Heidelberg.

**RKHS regression.** Using the representer theorem, one can derive the RKHS regression easily. The RKHS regression is to find the regression function that minimizes the following penalized loss function:

$$R_{\mathcal{H}}(m) = \sum_{i=1}^{n} (Y_i - m(X_i))^2 + \lambda \|m\|_{\mathcal{H}}^2.$$

Let $\mathbf{K} = [K(X_i, X_j)] \in \mathbb{R}^{n \times n}$ be the kernel matrix (Gram matrix). You can show that

$$\widehat{\alpha} = (\mathbf{K} + \lambda \mathbb{I})^{-1} \mathbb{Y}$$

and the predicted response is

$$\widehat{\mathbb{Y}} = \mathbf{K}\widehat{\alpha} = \mathbf{K}(\mathbf{K} + \lambda \mathbb{I})^{-1} \mathbb{Y},$$

which implies that the RKHS regression is another linear smoother.

[2] Modified from https://ocw.mit.edu/courses/sloan-school-of-management/15-097-prediction-machine-learning-and-statistics-spring-201
lecture-notes/MIT15_097S12_lec13.pdf