IEICE Transactions on Information and Systems,
vol.E89-D, no.7, pp.2243–2249, 2006.

Constructing Kernel Functions for Binary Regression

Masashi Sugiyama (sugi@cs.titech.ac.jp)
Department of Computer Science, Tokyo Institute of Technology
2-12-1, O-okayama, Meguro-ku, Tokyo, 152-8552, Japan

Hidemitsu Ogawa (hidemitsu-ogawa@kuramae.ne.jp)
Toray Engineering Co., Ltd.
1-1-45 Oe, Otsu, Shiga, 520-2141, Japan.

Abstract

Kernel-based learning algorithms have been successfully applied in various problem domains, given appropriate kernel functions. In this paper, we discuss the problem of designing kernel functions for binary regression and show that using a bell-shaped cosine function as a kernel function is optimal in some sense. The rationale of this result is based on the Karhunen-Loève expansion, i.e., the optimal approximation to a set of functions is given by the principal component of the correlation operator of the functions.

Keywords

supervised learning, regression, kernel methods, kernel functions, Karhunen-Loève expansion, principal component analysis, binary regression, Gaussian kernel.

1 Introduction

In recent years, a number of kernel-based learning algorithms such as the regularization networks [11, 5, 2], the support vector machines [18, 10, 14, 15], and the Gaussian processes [20, 19] have received growing attention. These kernel methods are shown to generalize very well in various problem domains, given appropriate *kernel functions*. Thus, properly choosing or designing the kernel function is crucial in kernel methods. In this paper, we discuss the problem of designing kernel functions. A lot of attention have been paid recently to designing kernel functions especially for non-vectorial structured data [18, 13, 8, 1, 21, 9, 16, 17, 14, 4]. In this paper, however, we consider the problem of designing kernel functions for standard vectorial data.

A kernel function is usually specified by the family of functions (Gaussian, polynomial, etc.) and kernel parameters (width, order, etc.). In practice, learning with kernels is often carried out by using a fixed family of kernel functions (say the Gaussian kernel) and the kernel parameters (Gaussian widths) are optimized by some model selection method such as cross-validation. Although it is in principle possible to also choose the family of kernel functions by cross-validation, this practice does not seem so common because of infinitely many degrees of freedom in the optimization of the family of kernel functions.

In this paper, we focus on the binary regression problem where the learning target function is binary, and show that using a bell-shaped cosine function as a kernel function is optimal in some sense. The rest of this paper is organized as follows. In Section 2 and Section 3, our basic idea of designing kernel functions and its details are described. Section 4 reports the experimental results for standard benchmark data sets, and Section 5 concludes the paper.

2 Basic Idea of Designing Kernel Functions

In this section, we illustrate our basic idea of designing kernel functions.

We consider the regression problem of approximating an unknown learning target function from training examples. Let us denote the learning target function by $f(\boldsymbol{x})$, which is defined on $\mathbb{R}^d$. We employ the kernel regression model (or the *kernel machine*) for learning:

$$\hat{f}(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i K(\boldsymbol{x}, \boldsymbol{x}_i), \tag{1}$$

where $\{\alpha_i\}_{i=1}^n$ are parameters to be estimated from training examples, $K(\boldsymbol{x}, \boldsymbol{x}')$ is a kernel function, and $\boldsymbol{x}_i$ ($\in \mathbb{R}^d$) is a training input point. We do not impose the positive semi-definiteness on the kernel function[1].

In the following, we focus on translation-invariant kernels [14], i.e., $K(\boldsymbol{x}, \boldsymbol{x}')$ depends only on $\boldsymbol{x} - \boldsymbol{x}'$. A notable feature of kernel regression models with translation-invariant kernels is that the shape of kernel functions is common to any $\boldsymbol{x}'$. That is, $\boldsymbol{x}'$ can

[1]The positive semi-definiteness of the kernel function is not required, e.g., in the ridge estimation [7].

be interpreted as the center of kernel functions. This fact implies that each kernel is responsible for local approximation in the vicinity of each training input point $\boldsymbol{x}_i$. For this reason, we consider the problem of approximating the learning target function $f(\boldsymbol{x})$ *locally* by a single kernel function and derive the optimal shape of kernel functions.

Let $\psi(\boldsymbol{x})$ be a local function centered at $\boldsymbol{x}'$ and Ψ be the set of all local functions. Let $\mathcal{H}$ be a functional Hilbert space which contains Ψ. The inner product and norm in $\mathcal{H}$ are denoted by $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$, respectively. We treat ψ as a random function[2] and denote the expectation over ψ by $\mathbb{E}$. Then the kernel design problem is formulated as the problem of searching for the optimal approximation to the set Ψ in the function space $\mathcal{H}$. Since we are interested in finding the optimal family of kernel functions, scaling of the kernel functions is not important. Therefore, we will search for the optimal direction ϕ_{opt} in the function space $\mathcal{H}$. Here, we define our optimality criterion by

$$\phi_{opt} = \operatorname*{argmin}_{\phi \in \mathcal{H}} \mathbb{E} \| \psi - \psi_\phi \|^2, \tag{2}$$

where ψ_ϕ is the orthogonal projection of ψ onto ϕ, i.e.,

$$\psi_\phi = \frac{\langle \psi, \phi \rangle}{\|\phi\|^2} \phi. \tag{3}$$

Let R be the correlation operator of local functions, i.e., it is defined by using $\varphi \in \mathcal{H}$ as

$$R\varphi = \mathbb{E}[\langle \varphi, \psi \rangle \psi]. \tag{4}$$

Then the well-known *Karhunen-Loève expansion* [3] asserts that the optimal direction ϕ_{opt} is given by the eigenfunction $\phi_{max}(x)$ associated with the largest eigenvalue λ_{max} of the correlation operator R. Based on this fact, we propose using the kernel function defined by

$$K(\boldsymbol{x}, \boldsymbol{x}') = \phi_{opt}\left(\frac{\boldsymbol{x} - \boldsymbol{x}'}{c} \right), \tag{5}$$

where $\boldsymbol{x}'$ is the center of the kernel function and c is a positive scaler that controls the kernel width (i.e., the larger c is, the wider the kernel width is). Since the above kernel consists of the principal component of the correlation operator, we call it *the principal component (PC) kernel*.

3 Constructing Kernel Functions for Binary Regression

In this section, we construct a kernel function for binary regression using the above idea.

Let us consider a one-dimensional binary regression problem, i.e., the output of the learning target function $f(x)$ is either 0 or 1 (see Figure 1). Then it can be observed that

[2] This does not mean that ψ is a probability density function, but the function ψ is drawn randomly as an element of the function space $\mathcal{H}$.

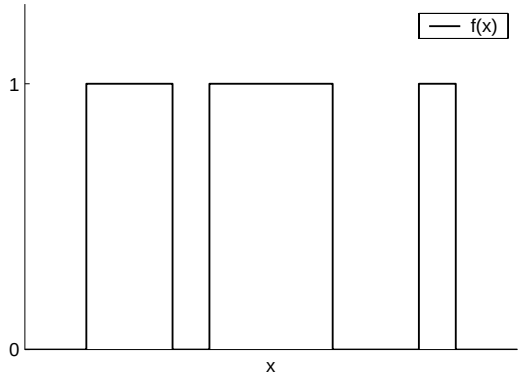
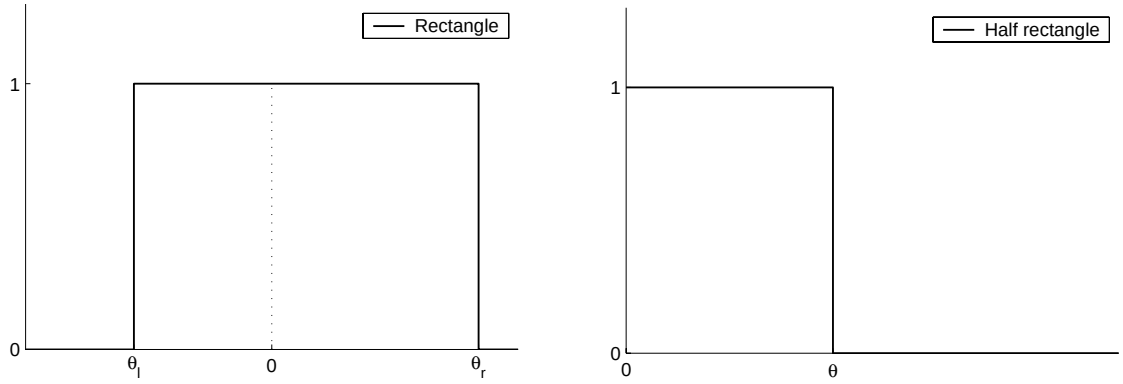


Figure 1: An example of one-dimensional binary learning target function $f(x)$.



(a) (b)

Figure 2: (a) a rectangle function and (b) a half-rectangle function.

such a binary function consists of rectangle functions with different widths. Therefore, in the binary regression cases, the set of local functions is given by a set of rectangle functions with different widths (see Figure 2-(a)). We regard the widths of rectangle functions (θ_l and θ_r in the figure) as random variables. Since we do not have any prior knowledge on the probability distribution of the widths, the distribution should be defined in an "unbiased" manner. Here we suppose that the width is bounded, and we use the uniform distribution for the widths because it is non-informative.

The above formulation implies that the problem is symmetric, so we only consider the right-half of the rectangle functions (see Figure 2-(b)). Without loss of generality, let us normalize the width into $[0, 1]$. Then the half-rectangle function is expressed by

$$\psi_\theta(x) = \begin{cases} 1 & \text{if } 0 \leq x \leq \theta, \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

where θ $(0 \leq \theta \leq 1)$ denotes the width of the half-rectangle function. Since we assumed the uniform distribution for θ, the probability density function $p(\theta)$ is given by

$$p(\theta) = 1 \text{ for } 0 \leq \theta \leq 1. \tag{7}$$

Let us use $L_2[0,1]$ as a functional Hilbert space $\mathcal{H}$. That is, $\mathcal{H}$ is spanned by the functions $\varphi(x)$ defined on $[0,1]$ such that

$$\int_0^1 |\varphi(x)|^2 dx < \infty. \tag{8}$$

The inner product in $\mathcal{H}$ is defined by

$$\langle \varphi, \varphi' \rangle = \int_0^1 \varphi(x)\overline{\varphi'(x)}dx, \tag{9}$$

where $\overline{\cdot}$ denotes the complex conjugate of a complex number. The norm is defined by $\|\varphi\| = \sqrt{\langle \varphi, \varphi \rangle}$. Then the correlation operator R of the half-rectangle functions is expressed using any $\varphi \in \mathcal{H}$ as

$$R\varphi = \int_0^1 \langle \varphi, \psi_\theta \rangle \psi_\theta d\theta. \tag{10}$$

Now let us solve the eigenproblem for R:

$$R\phi = \lambda\phi. \tag{11}$$

This eigenproblem can be expressed as follows.

Lemma 1 *The eigenproblem for R is expressed as*

$$\int_0^1 r(x,y)\phi(y)dy = \lambda\phi(x), \tag{12}$$

where

$$r(x,y) = \begin{cases} 1-y & \text{if } x \le y, \\ 1-x & \text{if } x > y. \end{cases} \tag{13}$$

A proof of Lemma 1 is given in A. Based on this lemma, we have the following theorem which gives the analytic solutions of the eigenproblem.

Theorem 2 *All positive eigenvalues $\{\lambda_p\}_{p=0}^{\infty}$ and associated normalized eigenfunctions $\{\phi_p(x)\}_{p=0}^{\infty}$ of R are given by*

$$\lambda_p = \frac{4}{(2p+1)^2\pi^2}, \tag{14}$$

$$\phi_p(x) = \sqrt{2}\cos\frac{(2p+1)\pi}{2}x. \tag{15}$$

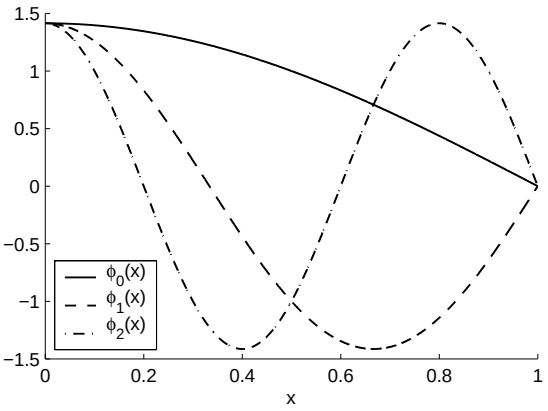


Figure 3: Profiles of the eigenfunctions $\phi_0(x)$, $\phi_1(x)$, and $\phi_2(x)$.

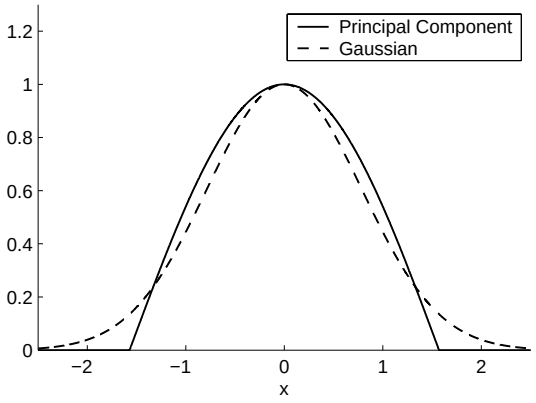


Figure 4: The profiles of the principal component kernel for binary regression and Gaussian kernel. Both kernels are centered at the origin.

A proof of Theorem 2 is given in B. Profiles of the three leading eigenfunctions $\phi_0(x)$, $\phi_1(x)$, and $\phi_2(x)$ are illustrated in Figure 3.

From Eq.(5) and Theorem 2, we have the following PC kernel for the binary regression problem.

$$K(x,x') = \begin{cases} \cos\left(\dfrac{x-x'}{c}\right) & \text{if } \dfrac{|x-x'|}{c} \leq \dfrac{\pi}{2}, \\ 0 & \text{otherwise,} \end{cases} \tag{16}$$

where the coefficient $\sqrt{2}$ is omitted, $\frac{2}{\pi}c$ is redefined by c, and the symmetry of the cosine function is used. A profile of the above kernel function is illustrated in Figure 4. Note that this kernel is not positive semi-definite, which can be confirmed by the fact that $x_1 = 0$, $x_2 = 0.1$, $x_3 = 0.7$, and $x_4 = 1.6$ yield a negative kernel matrix.

For comparison, a profile of the popular Gaussian kernel is also illustrated in the same figure, showing that the derived PC kernel is rather similar to the Gaussian kernel. This fact has the following implication: Binary classification problems are sometimes solved as binary regression problems using the squared-loss [2, 15]. In this scenario, an interesting

experimental fact is known that smooth kernels such as the Gaussian kernel work very well in practice, although the binary learning target function is not at all smooth. Our result partially explains this interesting phenomenon—Gaussian-like bell-shaped kernel functions are shown to approximate binary functions very well.

When the input variable $\boldsymbol{x}$ is multi-dimensional, there are several possibilities to extend the above result, e.g., the element-wise product of the one-dimensional kernel function or radially symmetric kernel function. In the following section, we focus on the radially symmetric kernel because it is computationally less expensive:

$$K(\boldsymbol{x},\boldsymbol{x}') = \begin{cases} \cos\left(\dfrac{\|\boldsymbol{x}-\boldsymbol{x}'\|}{c}\right) & \text{if } \dfrac{\|\boldsymbol{x}-\boldsymbol{x}'\|}{c} \leq \dfrac{\pi}{2}, \\ 0 & \text{otherwise,} \end{cases} \tag{17}$$

4 Simulations

In this section, we investigate the experimental performance of the proposed kernel function.

As benchmarks, we use the *IDA data sets* which are standard binary classification data sets originally used in the paper [12]. The data sets are available from '`http://ida.first.fraunhofer.de/projects/bench /benchmarks.htm`'. All the data sets we use here are binary classification problems and the labels in the original data sets are -1 and $+1$. Here we convert -1 to 0 to fit the current setting. In the theoretical discussions provided in the previous sections, we focused on binary regression problems (squared error). Therefore, the application of the proposed kernel to binary classification (misclassification error) is a heuristic.

We use the kernel regression model defined by Eq.(1) for learning and determine the parameters $\{\alpha_i\}_{i=1}^n$ by the ridge regression (RR) [7]. More specifically, the parameters $\{\alpha_i\}_{i=1}^n$ are determined such that the regularized training error is minimized.

$$\min_{\{\alpha_i\}_{i=1}^n} \left(\sum_{i=1}^n \left(\hat{f}(x_i) - y_i \right)^2 + \lambda \sum_{j=1}^n \alpha_j^2 \right), \tag{18}$$

where λ is a positive scalar called the *ridge parameter*. A minimizer of Eq.(18) is given by

$$(\hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_n)^\top = (\boldsymbol{K}^2 + \lambda \boldsymbol{I})^{-1} \boldsymbol{K} (y_1, y_2, \ldots, y_n)^\top, \tag{19}$$

where $\boldsymbol{I}$ denotes the identity matrix and $\boldsymbol{K}$ is the so-called kernel matrix, whose (i,j)-th element is given by

$$\boldsymbol{K}_{i,j} = K(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{20}$$

As a kernel function $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$, we use the proposed principal component kernel (PCK). Model parameters such as the widths of the kernels and the ridge parameter λ are determined following the original paper [12]: The model parameters are optimized using the first 5 realizations of each data set. For each realization, the model parameters are chosen

Table 1: Simulation results for IDA data sets. d is the input dimension and n is the number of training examples. Mean and standard deviations of the misclassification rates obtained by the ridge regression with the principal component kernel (RR+PCK) or with the Gaussian kernel (RR+GK) are described. All values are described in percent. A significantly better method by the t-test at the significance level 1% is described with boldface.

Data Set	d	n	RR+PCK	RR+GK
Banana	2	400	10.9±0.44	**10.4±0.41**
B. Cancer	9	200	**25.5±4.34**	27.2±4.84
Diabetes	8	468	23.2±1.82	23.0±1.68
German	20	700	24.1±2.08	24.2±2.21
Heart	13	170	15.9±3.41	15.5±3.25
Image	18	1300	6.52±0.71	**3.68±0.52**
Ringnorm	20	400	**2.69±0.32**	5.35±0.68
F. Solar	9	666	33.3±1.58	33.5±1.50
Splice	60	1000	11.4±0.58	11.1±0.59
Thyroid	5	140	6.61±2.84	**5.31±2.46**
Titanic	3	150	22.5±1.02	22.4±1.12
Twonorm	20	400	**2.45±0.14**	2.52±0.17
Waveform	21	400	9.73±0.44	9.81±0.48

from a wide range of values by 5-fold cross-validation. Then the median of 5 chosen values are used throughout all realization of the data set.

We compare the misclassification rates obtained by the principal component kernel (RR+PCK) with that obtained by the Gaussian kernel (RR+GK), which are described in Table 1. All values are described in percent. For each data set, a significantly better method by the *t-test* [6] at the significance level 1% is described with boldface. The table shows that PCK and GK work comparably, implying that the standard GK may be used as an approximation to PCK.

Table 2 shows the sparseness of the kernel matrix, i.e., the percentage of zeros in the kernel matrix. Thanks to the locality of the PC kernel, it provides a sparse kernel matrix for several data sets, which contributes to reducing computation time. Since RR+PCK and RR+GK are comparable in accuracy, using PCK is advantageous because of the sparseness.

Finally, we compare the misclassification rates obtained by RR+PCK with the support vector machine with the Gaussian kernel (SVM+GK)[3], which are described in Table 3. The table shows that RR+PCK works as well as SVM+GK, which is known to be one of the best existing classifiers. Although application to binary classification is rather heuristic, the results show that the proposed method (RR+PCK) may be useful in practice.

[3]The results of SVM+GK are borrowed from the paper [12].

Table 2: Mean and standard deviations of the sparsity of the kernel matrix. All values are described in percent.

Data Set	PCK	GK
Banana	57.3±0.94	0±0
B. Cancer	0±0	0±0
Diabetes	0.06±0.03	0±0
German	0±0	0±0
Heart	0±0	0±0
Image	2.38±0.45	0±0
Ringnorm	1.32±0.27	0±0
F. Solar	0±0	0±0
Splice	0±0	0±0
Thyroid	21.7±2.55	0±0
Titanic	8.96±3.08	0±0
Twonorm	0±0	0±0
Waveform	0±0	0±0

Table 3: Mean and standard deviations of the misclassification rates obtained by the ridge regression with the principal component kernel (RR+PCK) or the support vector machine with the Gaussian kernel (SVM+GK) are described. All values are described in percent. A significantly better method by the t-test at the significance level 1% is described with boldface.

Data Set	RR+PCK	SVM+GK
Banana	**10.9±0.44**	11.5±0.66
B. Cancer	25.5±4.34	26.0±4.74
Diabetes	23.2±1.82	23.5±1.73
German	24.1±2.08	23.6±2.07
Heart	15.9±3.41	16.0±3.26
Image	6.52±0.71	**2.96±0.60**
Ringnorm	2.69±0.32	**1.66±0.12**
F. Solar	33.3±1.58	32.4±1.82
Splice	11.4±0.58	10.9±0.66
Thyroid	6.61±2.84	**4.80±2.19**
Titanic	22.5±1.02	22.4±1.02
Twonorm	**2.45±0.14**	2.96±0.23
Waveform	9.73±0.44	9.88±0.43

5 Discussions and Conclusions

Optimizing a family of kernel functions or the "shape" of kernel functions is a hard task because it includes infinitely many degrees of freedom. In this paper, we showed that the optimal kernel shape is given by the principal component of the correlation operator of local functions, which resulted in a bell-shaped cosine kernel (see Figure 4). As the simulation results showed, the ridge regression with the proposed kernel works as well as the support vector machine, which is known to be an excellent classifier.

The profile of the obtained cosine kernel is rather similar to that of the standard Gaussian kernel. This fact explained why a smooth kernel such as the Gaussian kernel often works well in non-smooth binary regression problems. Indeed, our experiments showed that the proposed kernel and the Gaussian kernel work comparably. Therefore, the Gaussian kernel may be used as an approximation to the bell-shaped cosine kernel, although using the proposed kernel is more advantageous because it provides a sparse kernel matrix.

We did not take the positive semi-definiteness of the kernel function, which is not a problem if kernel machines with ridge estimation are used. However, it would be interesting to investigate whether the same or similar framework can be used for deriving the optimal positive semi-definite kernel.

We focused on the binary regression problem. However, we expect that the proposed kernel design methodology can be extended to more general regression senarios, since the basic idea described in Section 2 does not even exploit the fact that the learning target function is binary. In order to design a kernel function using the proposed methodology in more general scenarios, we need to appropriately specify the correlation operator defined by Eq.(4). If it can be specified using some prior knowledge on the problem domain, the proposed method allows us to beneficially use such prior knowledge. In the absence of such prior knowledge, on the other hand, we have to define the correlation operator such that the solution is not subjectively biased. In the binary regression case we discussed in this paper, our choice was to use the uniform distribution for the widths of the rectangle functions, which seems to be non-informative. A key point of this implementation was that the uniform distribution for the widths does not yield the uniform distribution in the function space. Therefore, we could find a "meaningful" principal component in the function space. In other words, if the uniform distribution in the function space is assumed, arbitrary functions in the function space become principal components so a meanningful outcome can not be obtained. Therefore, in order to extend the proposed kernel design method to be applicable to more general scenarios, it is important to find an appropriate way to design the correlation operator, which remains open currently.

Acknowledgements

The authors would like to thank Maki Fujino for her valuable comments. We are also grateful to anonymous reviewers. M. S. acknowledges MEXT Grant-in-Aid for Scientific Research (17700142) for partial financial support.

A Proof of Lemma 1

Let

$$r(x,y) = \int_0^1 \psi_\theta(x)\overline{\psi_\theta(y)}d\theta. \tag{21}$$

Then it follows from Eq.(10) that

$$\begin{aligned}(R\phi)(x) &= \int_0^1 \left(\int_0^1 \phi(y)\overline{\psi_\theta(y)}dy\right)\psi_\theta(x)d\theta \\ &= \int_0^1 r(x,y)\phi(y)dy. \end{aligned} \tag{22}$$

Substituting Eq.(6) into Eq.(21), we have

$$r(x,y) = \int_{\max(x,y)}^1 d\theta = 1 - \max(x,y), \tag{23}$$

which yields Eq.(13). ■

B Proof of Theorem 2

We shall solve the eigenproblem given by Eq.(12). Let us search eigenfunctions from $C^{(2)}[0,1]$, which is a set of twice-differentiable functions. Substituting Eq.(13) into Eq.(12) yields

$$\lambda\phi(x) = (1-x)\int_0^x \phi(y)dy + \int_x^1 (1-y)\phi(y)dy. \tag{24}$$

Differentiating both-hands sides of Eq.(24) with respect to x, we have

$$\begin{aligned}\lambda\phi'(x) &= -\int_0^x \phi(y)dy + (1-x)\phi(x) - (1-x)\phi(x) \\ &= -\int_0^x \phi(y)dy. \end{aligned} \tag{25}$$

Further differentiating both-hands sides of Eq.(25) with respect to x, we have

$$\lambda\phi''(x) = -\phi(x). \tag{26}$$

It is known that, for $\lambda > 0$, Eq.(26) has a general solution of the form

$$\phi(x) = a_c \cos\frac{x}{\sqrt{\lambda}} + a_s \sin\frac{x}{\sqrt{\lambda}}, \tag{27}$$

where a_c and a_s are complex numbers.

Now we shall determine the coefficients a_c and a_s. Eqs.(24) and (25) yield the following boundary conditions:

$$\phi(1) = 0, \tag{28}$$
$$\phi'(0) = 0. \tag{29}$$

Since differentiating both-hands sides of Eq.(27) with respect to x yields

$$\phi'(x) = \frac{1}{\sqrt{\lambda}}\left(-a_c \sin\frac{x}{\sqrt{\lambda}} + a_s \cos\frac{x}{\sqrt{\lambda}}\right), \tag{30}$$

Eqs.(30) and (29) imply

$$a_s = 0. \tag{31}$$

Therefore, we have

$$\phi(x) = a_c \cos\frac{x}{\sqrt{\lambda}}. \tag{32}$$

Eqs.(32) and (28) imply

$$a_c \cos\frac{1}{\sqrt{\lambda}} = 0, \tag{33}$$

so we have

$$\frac{1}{\sqrt{\lambda}} = \frac{(2p+1)\pi}{2} \text{ for } p = 0, 1, 2, \ldots. \tag{34}$$

For $p = 0, 1, 2, \ldots$, let

$$\lambda_p = \left(\frac{2}{(2p+1)\pi}\right)^2, \tag{35}$$

and let $\phi_p(x)$ be the associated eigenfunction.

$$\phi_p(x) = a_c \cos\frac{x}{\sqrt{\lambda_p}}. \tag{36}$$

In order for the eigenfunctions to be normalized, the coefficient a_c in Eq.(36) should be determined so that the norm of ϕ_p is equal to 1. Eqs.(36) and (35) yield

$$\begin{aligned}
\|\phi_p\|^2 &= |a_c|^2 \int_0^1 \cos^2\frac{x}{\sqrt{\lambda_p}} dx \\
&= \frac{|a_c|^2}{2}\int_0^1 \left(1 + \cos\frac{2x}{\sqrt{\lambda_p}}\right) dx \\
&= \frac{|a_c|^2}{2}\left(1 + \frac{\sqrt{\lambda_p}}{2}\sin\frac{2x}{\sqrt{\lambda_p}}\right) \\
&= \frac{|a_c|^2}{2}\left(1 + \frac{1}{(2p+1)\pi}\sin(2p+1)\pi\right) \\
&= \frac{|a_c|^2}{2} \\
&= 1,
\end{aligned} \tag{37}$$

which holds if $a_c = \sqrt{2}$. Consequently, the eigenfunction $\phi_p(x)$ is given by

$$\phi_p(x) = \sqrt{2}\cos\frac{(2p+1)\pi}{2}x. \tag{38}$$

Finally, we shall show that the eigenvalues $\{\lambda_p\}_{p=0}^{\infty}$ given by Eq.(35) is the complete set of positive eigenvalues of the correlation operator R. Since R is a positive semi-definite operator, it is enough to show

$$\sum_{p=0}^{\infty}\lambda_p = \mathrm{tr}\,(R)\,, \tag{39}$$

where $\mathrm{tr}\,(R)$ denotes the trace of R. It is known that

$$\sum_{p=0}^{\infty}\frac{1}{(2p+1)^2} = \frac{\pi^2}{8}. \tag{40}$$

Therefore the left-hand side of Eq.(39) yields

$$\begin{aligned}\sum_{p=0}^{\infty}\lambda_p &= \sum_{p=0}^{\infty}\left(\frac{2}{(2p+1)\pi}\right)^2 \\ &= \frac{4}{\pi^2}\sum_{p=0}^{\infty}\frac{1}{(2p+1)^2} \\ &= \frac{1}{2}.\end{aligned} \tag{41}$$

On the other hand, Eq.(13) yields

$$\mathrm{tr}\,(R) = \int_0^1 r(x,x)dx = \int_0^1 (1-x)dx = \frac{1}{2}, \tag{42}$$

which proves Eq.(39). ■

References

[1] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods.* Cambridge University Press, Cambridge, 2000.

[2] T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13(1):1–50, 2000.

[3] K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, Inc., Boston, second edition, 1990.

[4] T. Gärtner. A survey of kernels for structured data. *ACM SIGKDD Explorations Newsletter*, 5(1):49–58, 2003.

[5] F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 7(2):219–269, 1995.

[6] R. E. Henkel. *Tests of Significance*. SAGE Publication, Beverly Hills, 1979.

[7] A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(3):55–67, 1970.

[8] T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *Advances in Neural Information Processing Systems 11*, Cambridge, MA., 1999. MIT Press.

[9] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins. Text classification using string kernels. *Journal of Machine Learning Research*, 2:419–444, 2002.

[10] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12:181–201, March 2001.

[11] T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990.

[12] G. Rätsch, T. Onoda, and K.-R. Müller. Soft margins for adaboost. *Machine Learning*, 42(3):287–320, 2001.

[13] B. Schölkopf, P. Simard, A. J. Smola, and V. Vapnik. Prior knowledge in support vector kernels. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. The MIT Press, 1998.

[14] B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

[15] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle. *Least Squares Support Vector Machines*. World Scientific Pub. Co., Singapore, 2002.

[16] K. Tsuda, M. Kawanabe, G. Rätsch, S. Sonnenburg, and K.-R. Müller. A new discriminative kernel from probabilistic models. *Neural Computation*, 14(10):2397–2414, 2002.

[17] K. Tsuda, T. Kin, and K. Asai. Marginalized kernels for biological sequences. *Bioinformatics*, 18(1):268–275, 2002.

[18] V. N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

[19] C. K. I. Williams. Prediction with Gaussian processes: From linear regression to linear prediction and beyond. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 599–621. The MIT Press, Cambridge, 1998.

[20] C. K. I. Williams and C. E. Rasmussen. Gaussian processes for regression. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8, pages 514–520. The MIT Press, 1996.

[21] A. Zien, G. Rätsch, S. Mika, B. Schölkopf, T. Lengauer, and K.-R. Müller. Engineering support vector machine kernels that recognize translation initiation sites. *Bioinformatics*, 16(9):799–807, 2000.