Designing Kernel Functions
Using the Karhunen-Loève Expansion

1 Fraunhofer FIRST, Germany
2 Tokyo Institute of Technology, Japan

Masashi Sugiyama\textsuperscript{1,2} and Hidemitsu Ogawa\textsuperscript{2}
Kernel methods:
Approximate unknown function $f(x)$ by

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

- $\alpha_i$ : Parameters
- $K(x, x')$ : Kernel function
- $x_i$ : Training points

Kernel methods are known to generalize very well, given appropriate kernel function.

Therefore, how to choose (or design) kernel function is critical in kernel methods.
Recently, a lot of attention have been paid to designing kernel functions for non-vectorial structured data. e.g., strings, sequence, trees, graphs.

In this talk, however, we discuss the problem of designing kernel functions for standard vectorial data.
A kernel function is specified by

- A family of functions (Gaussian, polynomial, etc.)
- Kernel parameters (width, order, etc.)

We usually focus on a particular family (say Gaussian), and optimize kernel parameters by, e.g., cross-validation.

In principle, it is possible to optimize the family of kernels by CV.

However, this does not seem so common because of too many degrees of freedom.
Goal of Our Research

- We propose a method for finding optimal family of kernel functions using some prior knowledge on problem domain.

- We focus on
  - Regression (squared-loss)
  - Translation-invariant kernel

\[ K(x, x') = K(x - x') \]

- We do not assume kernel is positive semi-definite, since “kernel trick” is not needed in some regression methods (e.g. ridge).
A general method for designing translation-invariant kernels.
Example of kernel design for binary regression.
Implication of the results.
Specialty of Learning with Translation-Invariant Kernels

- Ordinary linear models:
  \[ \hat{f}(x) = \sum_{i=1}^{p} \alpha_i \varphi_i(x) \]
  \( \alpha_i \): Parameters
  \( \varphi_i(x) \): Basis function

- Kernel models:
  \[ \hat{f}(x) = \sum_{i=1}^{n} \alpha_i K(x - x_i) \]
  \( K(x - x') \): Translation-invariant kernel

- \( x_i \) is center of kernels.

- All basis functions have same shape!
Intuitively, each kernel function is responsible for local approximation in the vicinity of each training input point. Therefore, we consider the problem of approximating a function locally by a single kernel function.
Set of Local Functions and Function Space

- \( \psi(x) \): A local function centered at \( x' \)
- \( \Psi \): Set of all local functions
- \( H \): A functional Hilbert space which contains \( \Psi \) (i.e., space of local functions)
- Suppose \( \psi(x) \) is a probabilistic function.
We are looking for the optimal approximation to the set $\Psi$ of local functions $\psi(x)$.

Since we are interested in optimizing the family of functions, scaling is not important.

We search the optimal direction $\phi_{opt}$ in $H$.

$$\phi_{opt} = \arg \min_{\phi \in H} E \left\| \psi - \psi_{\phi} \right\|^2$$

$E$ : Expectation over $\psi$

$\psi_{\phi}$ : Projection of $\psi$ onto $\phi$
Karhunen-Loève Expansion

\[ \phi_{opt} = \arg \min_{\phi \in \mathcal{H}} E \left\| \psi - \psi \phi \right\|^2 \]

\( R \): Correlation operator of local functions

\[ R \phi = E \left[ \langle \phi, \psi \rangle \psi \right] \]

If \( \psi \) is vector,

\[ R = E \left[ \psi \psi^T \right] \]

Optimal direction \( \phi_{opt} \) is given by the eigenfunction \( \phi_{\text{max}} \) associated with the largest eigenvalue \( \lambda_{\text{max}} \) of \( R \).

\[ R \phi_{\text{max}} = \lambda_{\text{max}} \phi_{\text{max}} \]

Similar to PCA, but \( E[\psi] \neq 0 \).
Using $\phi_{opt}$, we define the kernel function by

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

$x'$: Center  
$c$ : Width

Since the above kernel consists of the principal component of the correlation operator, we call it the principal component (PC) kernel.
Example of Kernel Design: Binary Regression Problem

- Learning target function is binary.

- The set of local functions is a set of rectangular functions with different width.
We assume that the width of rectangular functions is bounded (and normalized).

Since we do not have prior knowledge on the width, we should define its distribution in an “unbiased” manner.

We use uniform distribution for the width since it is non-informative.

\[ \theta_l, \theta_r \sim U(0,1) \]
We use $L_2$ -space as a function space $H$.

Considering the symmetry, the eigenvalue problem $R\phi = \lambda \phi$ is expressed as

$$\int_0^1 r(x, y)\phi(y)dy = \lambda \phi(x)$$

$$r(x, y) = 1 - \max(x, y)$$

The principal component is given by

$$\phi_{\max}(x) = \sqrt{2} \cos \left( \frac{\pi}{2} x \right)$$
PC Kernel for Binary Regression

\[
K(x, x') = \begin{cases} 
\cos \left( \frac{x - x'}{c} \right) & \text{if } \left| \frac{x - x'}{c} \right| \leq \frac{\pi}{2} \\
0 & \text{otherwise}
\end{cases}
\]

- \( x' \): Center
- \( c \): Width

\( x' = 0, c = 1 \)
Implication of The Result

- Binary classification is often solved as binary regression with squared-loss (e.g., regularization networks, least-squares SVMs).

- Although binary function is not smooth at all, smooth Gaussian kernel often works very well in practice.

- Why?
Implication of The Result (cont.)

- By proper scaling, it can be confirmed that the shape of the obtained PC kernel is similar to Gaussian kernel.

- Both kernels work similarly in experiments.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>PC kernel</th>
<th>Gauss kernel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Banana</td>
<td>10.8 ± 0.6</td>
<td>11.4 ± 0.9</td>
</tr>
<tr>
<td>B.Cancer</td>
<td>27.1 ± 4.6</td>
<td>27.1 ± 4.9</td>
</tr>
<tr>
<td>Diabetes</td>
<td>23.2 ± 1.8</td>
<td>23.3 ± 1.7</td>
</tr>
<tr>
<td>F.Solar</td>
<td>33.6 ± 1.6</td>
<td>33.5 ± 1.6</td>
</tr>
<tr>
<td>Heart</td>
<td>16.1 ± 3.3</td>
<td>16.2 ± 3.4</td>
</tr>
<tr>
<td>Ringnorm</td>
<td>2.9 ± 0.3</td>
<td>6.7 ± 0.9</td>
</tr>
<tr>
<td>Thyroid</td>
<td>6.4 ± 3.0</td>
<td>6.1 ± 2.9</td>
</tr>
<tr>
<td>Titanic</td>
<td>22.7 ± 1.4</td>
<td>22.7 ± 1.0</td>
</tr>
<tr>
<td>Twonorm</td>
<td>2.6 ± 0.2</td>
<td>3.0 ± 0.2</td>
</tr>
<tr>
<td>Waveform</td>
<td>10.1 ± 0.7</td>
<td>10.0 ± 0.5</td>
</tr>
</tbody>
</table>
This implies that Gaussian-like bell-shaped function approximates binary functions very well.

This partially explains why smooth Gaussian kernel is suitable for non-smooth classification tasks.
Conclusions

- Optimizing the family of kernel functions is a difficult task because it has infinitely many degrees of freedom.
- We proposed a method for designing kernel functions in regression scenarios.
- The optimal kernel shape is given by the principal component of correlation operator of local functions.
- We can beneficially use prior knowledge on problem domain (e.g., binary)