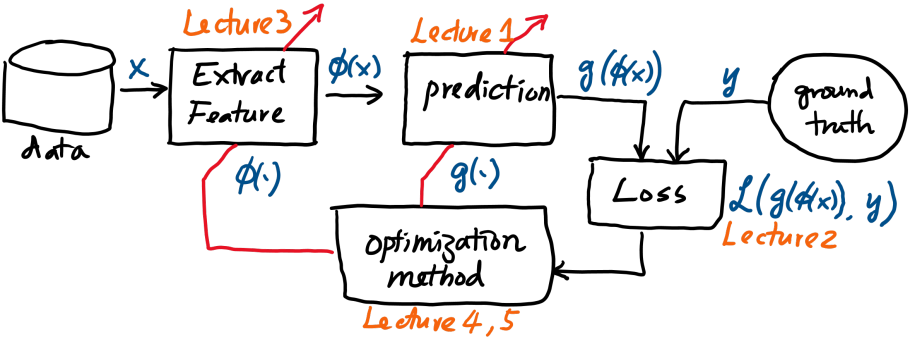
ECE595 / STAT598: Machine Learning I
Lecture 03: Regression with Kernels

Spring 2020

Stanley Chan

School of Electrical and Computer Engineering
Purdue University

Outline



data
x
Extract Feature
Lecture 3
$\phi(x)$
$\phi(\cdot)$
prediction
Lecture 1
$g(\phi(x))$
$g(\cdot)$
y
ground truth
Loss
$\mathcal{L}(g(\phi(x)), y)$
Lecture 2
optimization method
Lecture 4, 5

Outline

Mathematical Background

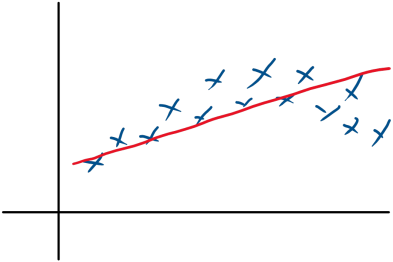
- Lecture 1: Linear regression: A basic data analytic tool
- Lecture 2: Regularization: Constraining the solution
- Lecture 3: Kernel Method: Enabling nonlinearity

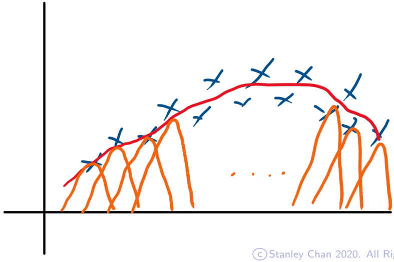
Lecture 3: Kernel Method

- Kernel Method
  - Dual Form
  - Kernel Trick
  - Algorithm
- Examples
  - Radial Basis Function (RBF)
  - Regression using RBF
  - Kernel Methods in Classification

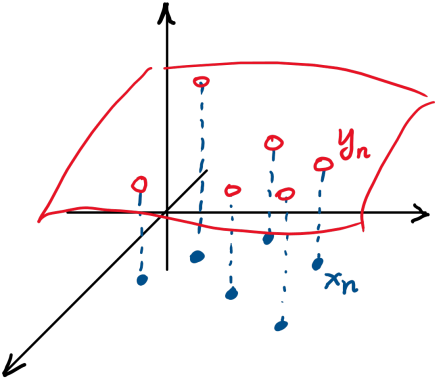
Why Another Method?

- Linear regression: Pick a **global** model, best fit globally.
- Kernel method: Pick a **local** model, best fit locally.
- In kernel method, instead of picking a line / a quadratic equation, we pick a **kernel**.
- A kernel is a measure of **distance** between **training samples**.
- Kernel method buys us the ability to handle nonlinearity.
- Ordinary regression is based on the **columns** (features) of $\boldsymbol{A}$.
- Kernel method is based on the **rows** (samples) of $\boldsymbol{A}$.





Pictorial Illustration



<u>goal</u>: learn the surface

<u>prediction</u>: when new sample comes, Interpolate on the surface

Overview of the Method

Model Parameter:

- We **want** the model parameter $\widehat{\boldsymbol{\theta}}$ to look like: (How? Question 1)

$$\widehat{\boldsymbol{\theta}} = \sum_{n=1}^{N} \alpha_n \boldsymbol{x}^n.$$

- This model expresses $\widehat{\boldsymbol{\theta}}$ as a combination of the **samples**.
- The trainable parameters are α_n, where $n = 1, \ldots, N$.
- If we can make α_n **local**, i.e., non-zero for only a few of them, then we can achieve our goal: localized, sample-dependent.

Predicted Value

- The predicted value of a new sample $\boldsymbol{x}$ is

$$\widehat{y} = \widehat{\boldsymbol{\theta}}^T \boldsymbol{x} = \sum_{n=1}^{N} \alpha_n \langle \boldsymbol{x}, \boldsymbol{x}^n \rangle.$$

- We **want** this model to encapsulate nonlinearity. (How? Question 2)

Dual Form of Linear Regression

Goal: Addresses Question 1: Express $\widehat{\boldsymbol{\theta}}$ as

$$\widehat{\boldsymbol{\theta}} = \sum_{n=1}^{N} \alpha_n \boldsymbol{x}^n.$$

We start by listing out a technical lemma:

Lemma

For any $\boldsymbol{A} \in \mathbb{R}^{N \times d}$, $\boldsymbol{y} \in \mathbb{R}^d$, and $\lambda > 0$,

$$(\boldsymbol{A}^T\boldsymbol{A} + \lambda \boldsymbol{I})^{-1}\boldsymbol{A}^T\boldsymbol{y} = \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T + \lambda\boldsymbol{I})^{-1}\boldsymbol{y}. \tag{1}$$

Proof: See Appendix.

Remark:

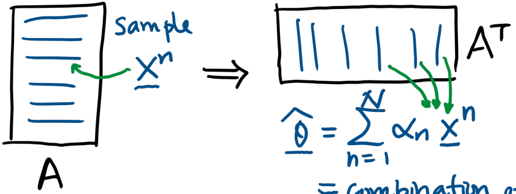
- The dimensions of $\boldsymbol{I}$ on the left is $d \times d$, on the right is $N \times N$.
- If $\lambda = 0$, then the above is true only when $\boldsymbol{A}$ is invertible.

Dual Form of Linear Regression

- Using the Lemma, we can show that

$$
\begin{aligned}
\widehat{\boldsymbol{\theta}} &= (\boldsymbol{A}^T\boldsymbol{A} + \lambda\boldsymbol{I})^{-1}\boldsymbol{A}^T\boldsymbol{y} && \text{(Primal Form)}\\
&= \boldsymbol{A}^T\underbrace{(\boldsymbol{A}\boldsymbol{A}^T + \lambda\boldsymbol{I})^{-1}\boldsymbol{y}}_{\stackrel{\text{def}}{=}\boldsymbol{\alpha}} && \text{(Dual Form)}
\end{aligned}
$$

$$
= \begin{bmatrix} — & (\boldsymbol{x}^1)^T & — \\ — & (\boldsymbol{x}^2)^T & — \\ & \vdots & \\ — & (\boldsymbol{x}^N)^T & — \end{bmatrix}^T \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{bmatrix} = \sum_{n=1}^{N} \alpha_n \boldsymbol{x}^n, \quad \alpha_n \stackrel{\text{def}}{=} [(\boldsymbol{A}\boldsymbol{A}^T + \lambda\boldsymbol{I})^{-1}\boldsymbol{y}]
$$



$$\underline{\hat{\theta}} = \sum_{n=1}^{N} \alpha_n \underline{x}^n$$

= combination of samples

The Kernel Trick

Goal: Addresses Question 2: Introduce nonlinearity to

$$\widehat{y} = \widehat{\boldsymbol{\theta}}^T \boldsymbol{x} = \sum_{n=1}^{N} \alpha_n \langle \boldsymbol{x}, \boldsymbol{x}^n \rangle.$$

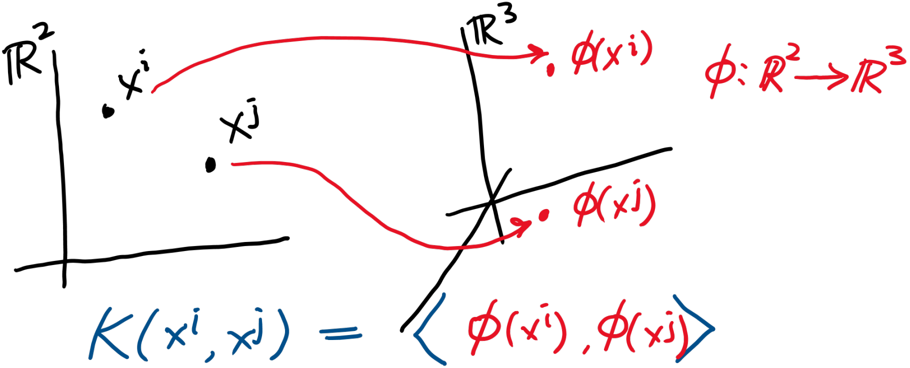
The Idea:

- Replace the inner product $\langle \boldsymbol{x}, \boldsymbol{x}^n \rangle$ by $k(\boldsymbol{x}, \boldsymbol{x}^n)$:

$$\widehat{y} = \widehat{\boldsymbol{\theta}}^T \boldsymbol{x} = \sum_{n=1}^{N} \alpha_n k(\boldsymbol{x}, \boldsymbol{x}^n).$$

- $k(\cdot, \cdot)$ is called a **kernel**.
- A kernel is a measure of the **distance** between two samples $\boldsymbol{x}^i$ and $\boldsymbol{x}^j$.
- $\langle \boldsymbol{x}^i, \boldsymbol{x}^j \rangle$ measure distance in the ambient space, $k(\boldsymbol{x}^i, \boldsymbol{x}^j)$ measure distance in a **transformed** space.
- In particular, a valid kernel takes the form $k(\boldsymbol{x}^i, \boldsymbol{x}^j) = \langle \phi(\boldsymbol{x}^i), \phi(\boldsymbol{x}^j) \rangle$ for some nonlinear transforms ϕ.

Kernels Illustrated



- A kernel typically lifts the ambient dimension to a **higher** one.
- For example, mapping from $\mathbb{R}^2$ to $\mathbb{R}^3$

$$\boldsymbol{x}^n = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad \text{and} \quad \phi(\boldsymbol{x}_n) = \begin{bmatrix} x_1^2 \\ x_1 x_2 \\ x_2^2 \end{bmatrix}$$

Relationship between Kernel and Transform

Consider the following kernel $k(\boldsymbol{u}, \boldsymbol{v}) = (\boldsymbol{u}^T\boldsymbol{v})^2$. What is the transform?

- Suppose $\boldsymbol{u}$ and $\boldsymbol{v}$ are in $\mathbb{R}^2$. Then $(\boldsymbol{u}^T\boldsymbol{v})^2$ is

$$
\begin{aligned}
(\boldsymbol{u}^T\boldsymbol{v})^2 &= \left(\sum_{i=1}^{2} u_i v_i\right)\left(\sum_{j=1}^{2} u_j v_j\right) \\
&= \sum_{i=1}^{2}\sum_{j=1}^{2}(u_i u_j)(v_i v_j) = \begin{bmatrix} u_1^2 & u_1 u_2 & u_2 u_1 & u_2^2 \end{bmatrix}\begin{bmatrix} v_1^2 \\ v_1 v_2 \\ v_2 v_1 \\ v_2^2 \end{bmatrix}.
\end{aligned}
$$

- So if we define ϕ as

$$
\boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad \mapsto \quad \phi(\boldsymbol{u}) = \begin{bmatrix} u_1^2 \\ u_1 u_2 \\ u_2 u_1 \\ u_2^2 \end{bmatrix}
$$

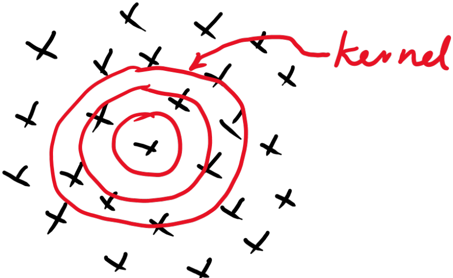
then $(\boldsymbol{u}^T\boldsymbol{v})^2 = \langle \phi(\boldsymbol{u}), \phi(\boldsymbol{v}) \rangle$.

Radial Basis Function

A useful kernel is the **radial basis kernel** (RBF):

$$k(\boldsymbol{u}, \boldsymbol{v}) = \exp\left\{-\frac{\|\boldsymbol{u}-\boldsymbol{v}\|^2}{2\sigma^2}\right\}.$$

- The corresponding nonlinear transform of RBF is **infinite dimensional**. See Appendix.
- $\|\boldsymbol{u}-\boldsymbol{v}\|^2$ measures the distance between two data points $\boldsymbol{u}$ and $\boldsymbol{v}$.
- σ is the std dev, defining "far" and "close".
- RBF enforces **local** structure; Only a few samples are used.



Kernel Method

Given the choice of the kernel function, we can write down the algorithm as follows.

1. Pick a kernel function $k(\cdot, \cdot)$.
2. Construct a kernel matrix $\boldsymbol{K} \in \mathbb{R}^{N \times N}$, where $[\boldsymbol{K}]_{ij} = k(\boldsymbol{x}^i, \boldsymbol{x}^j)$, for $i = 1, \ldots, N$ and $j = 1, \ldots, N$.
3. Compute the coefficients $\boldsymbol{\alpha} \in \mathbb{R}^N$, with

$$\alpha_n = [(\boldsymbol{K} + \lambda \boldsymbol{I})^{-1} \boldsymbol{y}]_n.$$

4. Estimate the predicted value for a new sample $\boldsymbol{x}$:

$$g_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{n=1}^{N} \alpha_n k(\boldsymbol{x}, \boldsymbol{x}^n).$$

Therefore, the choice of the regression function is shifted to the choice of the kernel.

Outline

Mathematical Background

- Lecture 1: Linear regression: A basic data analytic tool
- Lecture 2: Regularization: Constraining the solution
- Lecture 3: Kernel Method: Enabling nonlinearity

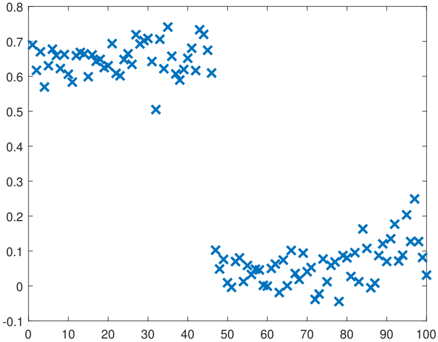
Lecture 3: Kernel Method

- Kernel Method
  - Dual Form
  - Kernel Trick
  - Algorithm
- Examples
  - Radial Basis Function (RBF)
  - Regression using RBF
  - Kernel Methods in Classification

Example

Goal: Use the kernel method to fit the data points shown as follows.

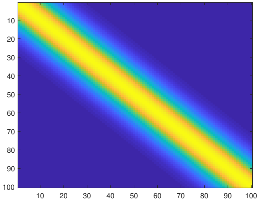
- What is the input feature vector $\boldsymbol{x}^n$? $\boldsymbol{x}^n = t_n$: The time stamps.
- What is the output y_n? y^n is the height.
- Which kernel to choose? Let us consider the RBF.



Example (using RBF)

- Define the fitted function as $g_{\boldsymbol{\theta}}(t)$. [Here, $\boldsymbol{\theta}$ refers to $\boldsymbol{\alpha}$.]
- The RBF is defined as $k(t_i, t_j) = \exp\{-(t_i - t_j)^2/2\sigma^2\}$, for some σ.
- The matrix $\boldsymbol{K}$ looks something below

$$[\boldsymbol{K}]_{ij} = \exp\{-(t_i - t_j)^2/2\sigma^2\}.$$



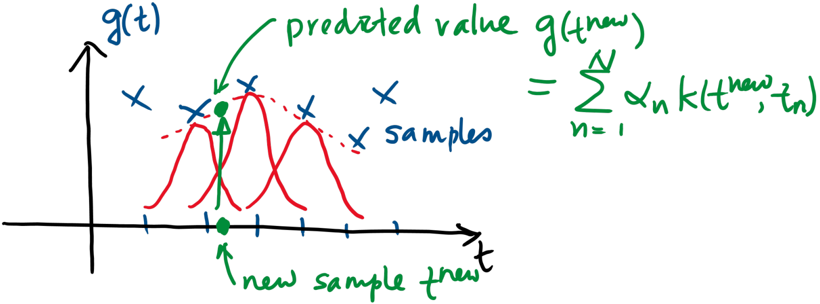
- $\boldsymbol{K}$ is a banded diagonal matrix. (Why?)
- The coefficient vector is $\alpha_n = [(\boldsymbol{K} + \lambda \boldsymbol{I})^{-1}\boldsymbol{y}]_n$.

Example (using RBF)

- Using the RBF, the predicted value is given by

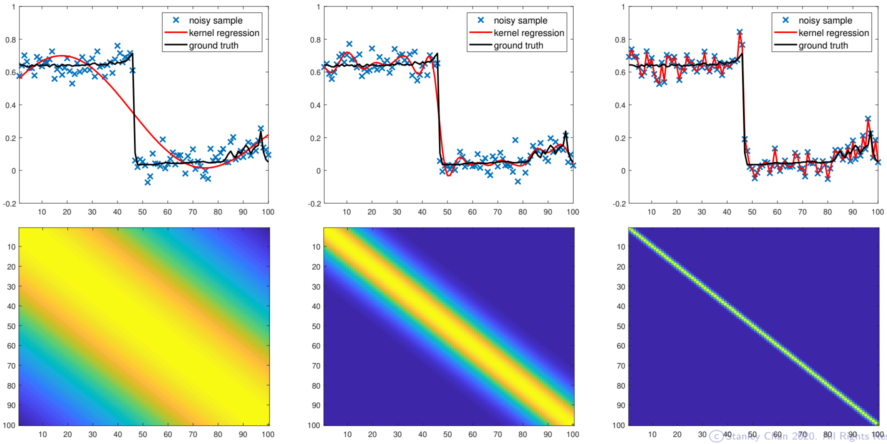
$$g_{\theta}(t^{\text{new}}) = \sum_{n=1}^{N} \alpha_n k(t^{\text{new}}, t_n) = \sum_{n=1}^{N} \alpha_n e^{-\frac{(t^{\text{new}} - t_n)^2}{2\sigma^2}}.$$

- Pictorially, the predicted function g_{θ} can be viewed as the linear combination of the Gaussian kernels.



Effect of σ

- Large σ: Flat kernel. Over-smoothing.
- Small σ: Narrow kernel. Under-smoothing.
- Below shows an example of the fitting and the kernel matrix $\boldsymbol{K}$.



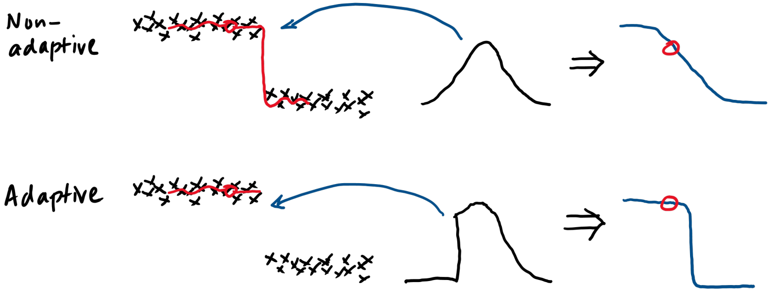
Too large | About right | Too small

Any Improvement?

- We can improve the above kernel by considering $\boldsymbol{x}^n = [y_n,\ t_n]^T$.
- Define the kernel as

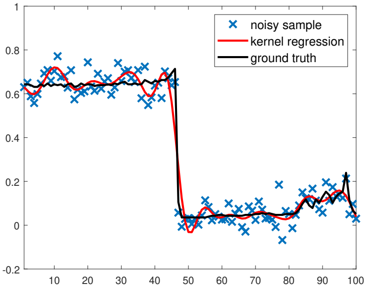
$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left\{-\left(\frac{(y_i - y_j)^2}{2\sigma_r^2} + \frac{(t_i - t_j)^2}{2\sigma_s^2}\right)\right\}.$$

- This new kernel is adaptive (**edge-aware**).

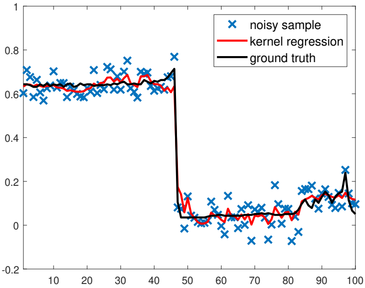


Any Improvement?

Here is a comparison.



without improvement

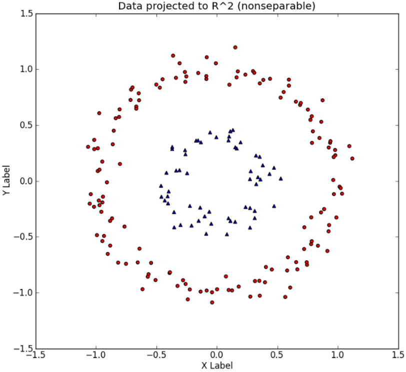


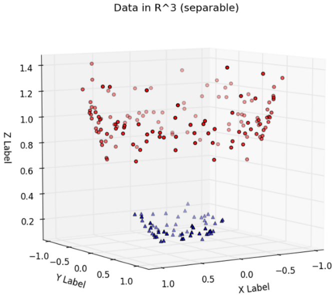
with improvement

- This idea is known as **bilateral filter** in the computer vision literature.
- Can be further extended to 2D image where $\boldsymbol{x}^n = [y_n, \boldsymbol{s}_n]$, for some spatial coordinate $\boldsymbol{s}_n$.
- Many applications. See Reading List.

Kernel Methods in Classification

- The concept of lifting the data to higher dimension is useful for classification. [1]



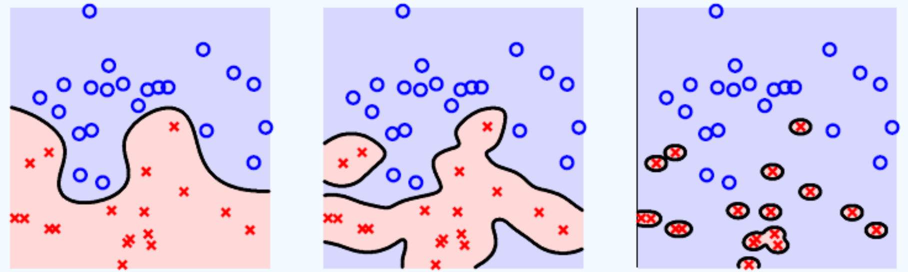


[1] Image source:
https://towardsdatascience.com/understanding-the-kernel-trick-e0bc6112ef78

Kernels in Support Vector Machines

Example. RBF for SVM (We will discuss SVM later in the semester.)

- Radial Basis Function is often used in support vector machine.
- Poor choice of parameter can lead to low training loss, but with the risk of over-fit.
- Under-fitted data can sometimes give better generalization.



$\exp(-1\|\mathbf{x}-\mathbf{x}'\|^2)$ $\exp(-10\|\mathbf{x}-\mathbf{x}'\|^2)$ $\exp(-100\|\mathbf{x}-\mathbf{x}'\|^2)$

Reading List

Kernel Method:

- Learning from Data (Chapter 3.4) https://work.caltech.edu/telecourse
- CMU 10701 Lecture 4 https://www.cs.cmu.edu/~tom/10701_sp11/slides/Kernels_SVM_04_7_2011-ann.pdf
- Berkeley CS 194 Lecture 7 https://people.eecs.berkeley.edu/~russell/classes/cs194/
- Oxford C19 Lecture 3 http://www.robots.ox.ac.uk/~az/lectures/ml/lect3.pdf

Kernel Regression in Computer Vision:

- Bilateral Filter https://people.csail.mit.edu/sparis/bf_course/course_notes.pdf
- Takeda and Milanfar, "Kernel regression for image processing and reconstruction", IEEE Trans. Image Process. (2007) https://ieeexplore.ieee.org/document/4060955

Appendix

Proof of Lemma

Lemma

For any matrix $\boldsymbol{A} \in \mathbb{R}^{N\times d}$, $\boldsymbol{y} \in \mathbb{R}^d$, *and* $\lambda > 0$,

$$(\boldsymbol{A}^T\boldsymbol{A} + \lambda\boldsymbol{I})^{-1}\boldsymbol{A}^T\boldsymbol{y} = \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T + \lambda\boldsymbol{I})^{-1}\boldsymbol{y}. \tag{2}$$

- The left hand side is solution to normal equation, which means $\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{\theta} + \lambda\boldsymbol{\theta} = \boldsymbol{A}^T\boldsymbol{y}$.
- Rearrange terms gives $\boldsymbol{\theta} = \boldsymbol{A}^T\left[\frac{1}{\lambda}(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta})\right]$.
- Define $\boldsymbol{\alpha} = \frac{1}{\lambda}(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta})$, then $\boldsymbol{\theta} = \boldsymbol{A}^T\boldsymbol{\alpha}$.
- Substitute $\boldsymbol{\theta} = \boldsymbol{A}^T\boldsymbol{\alpha}$ into $\boldsymbol{\alpha} = \frac{1}{\lambda}(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\theta})$, we have

$$\boldsymbol{\alpha} = \frac{1}{\lambda}(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{A}^T\boldsymbol{\alpha}).$$

- Rearrange terms gives $(\boldsymbol{A}\boldsymbol{A}^T + \lambda\boldsymbol{I})\boldsymbol{\alpha} = \boldsymbol{y}$, which yields $\boldsymbol{\alpha} = (\boldsymbol{A}\boldsymbol{A}^T + \lambda\boldsymbol{I})^{-1}\boldsymbol{y}$.
- Substitute into $\boldsymbol{\theta} = \boldsymbol{A}^T\boldsymbol{\alpha}$ gives $\boldsymbol{\theta} = \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T + \lambda\boldsymbol{I})^{-1}\boldsymbol{y}$.

Non-Linear Transform for RBF

- Let us consider scalar $u \in \mathbb{R}$.

$$
\begin{aligned}
k(u,v) &= \exp\{-(u-v)^2\} \\
&= \exp\{-u^2\}\exp\{2uv\}\exp\{-v^2\} \\
&= \exp\{-u^2\}\left(\sum_{k=0}^{\infty}\frac{2^k u^k v^k}{k!}\right)\exp\{-v^2\} \\
&= \exp\{-u^2\}\left(1, \sqrt{\frac{2^1}{1!}}u, \sqrt{\frac{2^2}{2!}}u^2, \sqrt{\frac{2^3}{3!}}u^3, \ldots,\right)^T \\
&\quad \times \left(1, \sqrt{\frac{2^1}{1!}}v, \sqrt{\frac{2^2}{2!}}v^2, \sqrt{\frac{2^3}{3!}}v^3, \ldots,\right)\exp\{-v^2\}
\end{aligned}
$$

- So Φ is

$$
\phi(x) = \exp\{-x^2\}\left(1, \sqrt{\frac{2^1}{1!}}x, \sqrt{\frac{2^2}{2!}}x^2, \sqrt{\frac{2^3}{3!}}x^3, \ldots,\right)
$$

Kernels are Positive Semi-Definite

Given $\{\boldsymbol{x}_j\}_{j=1}^N$, construct a $N \times N$ matrix $\boldsymbol{K}$ such that

$$[\boldsymbol{K}]_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Phi(\boldsymbol{x}_i)^T \Phi(\boldsymbol{x}_j).$$

Claim: $\boldsymbol{K}$ is positive semi-definite.

Let $\boldsymbol{z}$ be an arbitrary vector. Then,

$$\begin{aligned}
\boldsymbol{z}^T \boldsymbol{K} \boldsymbol{z} &= \sum_{i=1}^{n} \sum_{j=1}^{N} z_i K_{ij} z_j = \sum_{i=1}^{N} \sum_{j=1}^{N} z_i \Phi(\boldsymbol{x}_i)^T \Phi(\boldsymbol{x}_j) z_j \\
&= \sum_{i=1}^{N} \sum_{j=1}^{N} z_i \left(\sum_{k=1}^{N} [\Phi(\boldsymbol{x}_i)]_k [\Phi(\boldsymbol{x}_j)]_k \right) z_j \overset{(a)}{=} \sum_{k=1}^{N} \left(\sum_{i=1}^{N} [\Phi(\boldsymbol{x}_i)]_k z_i \right)^2 \geq 0
\end{aligned}$$

where $[\Phi(\boldsymbol{x}_i)]_k$ denotes the k-th element of the vector $\Phi(\boldsymbol{x}_i)$.

Existence of Nonlinear Transform

- We just showed that: If $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Phi(\boldsymbol{x}_i)^T \Phi(\boldsymbol{x}_j)$ for any $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, then $\boldsymbol{K}$ is symmetric positive semi-definite.
- The converse also holds: If $\boldsymbol{K}$ is symmetric positive semi-definite for any $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, then there exist Φ such that $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Phi(\boldsymbol{x}_i)^T \Phi(\boldsymbol{x}_j)$.
- This converse is difficult to prove.
- It is called the **Mercer Condition**.
- Kernels satisfying Mercer's condition have Φ.
- You can use the condition to rule out invalid kernels.
- But proving a valid kernel is still hard.