Kernel Methods

Lei Tang

Arizona State University

Jul. 26th, 2007

Introduction

- Linear parametric models for regression and classification.
- Memory-based methods: Parzen probability density estimation, k-nearest neighbor.
- Storing the entire training set in order to make predictions for future data.
- Fast to "train", but slow at prediction.
- Is it possible to connect these two different formulations?

Introduction

- Linear parametric models for regression and classification.
- Memory-based methods: Parzen probability density estimation, k-nearest neighbor.
- Storing the entire training set in order to make predictions for future data.
- Fast to "train", but slow at prediction.
- Is it possible to connect these two different formulations?

Introduction

- Linear parametric models for regression and classification.
- Memory-based methods: Parzen probability density estimation, k-nearest neighbor.
- Storing the entire training set in order to make predictions for future data.
- Fast to "train", but slow at prediction.
- Is it possible to connect these two different formulations?

Dual Representations

- Many Linear models for regression and classification can be reformulated in terms of a dual representation in which kernel function arises naturally.

$$J(w) = \frac{1}{2}\sum_{n=1}^{N}\left\{w^T\phi(x_n) - t_n\right\}^2 + \frac{\lambda}{2}w^Tw \tag{1}$$

The derivative with respect to w is

$$\nabla J(w) = \sum_{i=1}^{N} \left[w^T \phi(x_n) - t_n \right] \phi(x_n) + \lambda w = 0$$

$$\implies \quad w = -\frac{1}{\lambda} \sum_{n=1}^{N} \left\{ w^T \phi(x_n) - t_n \right\} = \sum_{n=1}^{N} a_n \phi(x_n) = \Phi^T a$$

$$a_n = -\frac{1}{\lambda} \left\{ w^T \phi(x_n) - t_n \right\}$$

Plug in the new formulation of $w = \Phi^T a$ into $J(w)$,

$$J(w) = \frac{1}{2}(\Phi w - \mathbf{t})^T(\Phi w - \mathbf{t}) + \frac{\lambda}{2} w^T w$$

$$= \frac{1}{2} a^T \Phi\Phi^T \Phi\Phi^T a - a^T \underbrace{\Phi\Phi^T}_{K} \mathbf{t} + \frac{1}{2}\mathbf{t}^T\mathbf{t} + \frac{\lambda}{2}\Phi\Phi^T a$$

$$J(a) = \frac{1}{2} a^T K K a - a^T K \mathbf{t} + \frac{1}{2}\mathbf{t}^T\mathbf{t} + \frac{\lambda}{2} a^T K a$$

$$\Longrightarrow \quad a = (K + \lambda I_N)^{-1}\mathbf{t}$$

$$y(x) = w^T \phi(x) = a^T \Phi \phi(x) = k(x)^T (K + \lambda I_N)^{-1}\mathbf{t} = a^T k(x)$$

Plug in the new formulation of $w = \Phi^T a$ into $J(w)$,

$$
\begin{aligned}
J(w) &= \frac{1}{2}(\Phi w - \mathbf{t})^T(\Phi w - \mathbf{t}) + \frac{\lambda}{2} w^T w \\
&= \frac{1}{2} a^T \Phi\Phi^T \Phi\Phi^T a - a^T \underbrace{\Phi\Phi^T}_{K} \mathbf{t} + \frac{1}{2}\mathbf{t}^T\mathbf{t} + \frac{\lambda}{2}\Phi\Phi^T a \\
J(a) &= \frac{1}{2} a^T K K a - a^T K \mathbf{t} + \frac{1}{2}\mathbf{t}^T\mathbf{t} + \frac{\lambda}{2} a^T K a
\end{aligned}
$$

$$
\begin{aligned}
\Longrightarrow \quad a &= (K + \lambda I_N)^{-1}\mathbf{t} \\
y(x) &= w^T\phi(x) = a^T\Phi\phi(x) = k(x)^T(K + \lambda I_N)^{-1}\mathbf{t} = a^T k(x)
\end{aligned}
$$

Plug in the new formulation of $w = \Phi^T a$ into $J(w)$,

$$
\begin{aligned}
J(w) &= \frac{1}{2}(\Phi w - \mathbf{t})^T(\Phi w - \mathbf{t}) + \frac{\lambda}{2} w^T w \\
&= \frac{1}{2} a^T \Phi\Phi^T \Phi\Phi^T a - a^T \underbrace{\Phi\Phi^T}_{K} \mathbf{t} + \frac{1}{2}\mathbf{t}^T\mathbf{t} + \frac{\lambda}{2}\Phi\Phi^T a \\
J(a) &= \frac{1}{2} a^T K K a - a^T K \mathbf{t} + \frac{1}{2}\mathbf{t}^T\mathbf{t} + \frac{\lambda}{2} a^T K a
\end{aligned}
$$

$$
\Longrightarrow \quad
\begin{aligned}
a &= (K + \lambda I_N)^{-1}\mathbf{t} \\
y(x) &= w^T \phi(x) = a^T \Phi \phi(x) = k(x)^T (K + \lambda I_N)^{-1}\mathbf{t} = a^T k(x)
\end{aligned}
$$

Advantages of dual methods

- The dual formulation allows the solution to be expressed entirely in terms of the kernel function $k(x, x')$.
- In dual formulation, need to invert a $N \times N$ matrix as

$$a = (K + \lambda I_N)^{-1}\mathbf{t}$$

- In the original parameter, need to invert a $M \times M$ matrix,

$$w = (\lambda I + \Phi^T \Phi)^{-1} \Phi^T \mathbf{t}$$

- If number of instances is smaller than dimensionality, dual formulation is preferred.
- Dual formulation directly works on kernels, avoids the explicit introduction of feature vector $\phi(x)$.

Advantages of dual methods

- The dual formulation allows the solution to be expressed entirely in terms of the kernel function $k(x, x')$.
- In dual formulation, need to invert a $N \times N$ matrix as

$$a = (K + \lambda I_N)^{-1}\mathbf{t}$$

- In the original parameter, need to invert a $M \times M$ matrix,

$$w = (\lambda I + \Phi^T \Phi)^{-1}\Phi^T \mathbf{t}$$

- If number of instances is smaller than dimensionality, dual formulation is preferred.
- Dual formulation directly works on kernels, avoids the explicit introduction of feature vector $\phi(x)$.

Advantages of dual methods

- The dual formulation allows the solution to be expressed entirely in terms of the kernel function $k(x, x')$.
- In dual formulation, need to invert a $N \times N$ matrix as

$$a = (K + \lambda I_N)^{-1}\mathbf{t}$$

- In the original parameter, need to invert a $M \times M$ matrix,

$$w = (\lambda I + \Phi^T\Phi)^{-1}\Phi^T\mathbf{t}$$

- If number of instances is smaller than dimensionality, dual formulation is preferred.
- Dual formulation directly works on kernels, avoids the explicit introduction of feature vector $\phi(x)$.

Advantages of dual methods

- The dual formulation allows the solution to be expressed entirely in terms of the kernel function $k(x, x')$.
- In dual formulation, need to invert a $N \times N$ matrix as

$$a = (K + \lambda I_N)^{-1}\mathbf{t}$$

- In the original parameter, need to invert a $M \times M$ matrix,

$$w = (\lambda I + \Phi^T\Phi)^{-1}\Phi^T\mathbf{t}$$

- If number of instances is smaller than dimensionality, dual formulation is preferred.
- Dual formulation directly works on kernels, avoids the explicit introduction of feature vector $\phi(x)$.

The Representer Theorem

More general case:
Denote by $\Omega : [0, \infty) \rightarrow \mathcal{R}$ a strictly monotonic increasing function, by $\mathcal{X}$ a set, and by c an arbitrary loss function. Then each minimizer $f \in \mathcal{H}$ of the regularized risk

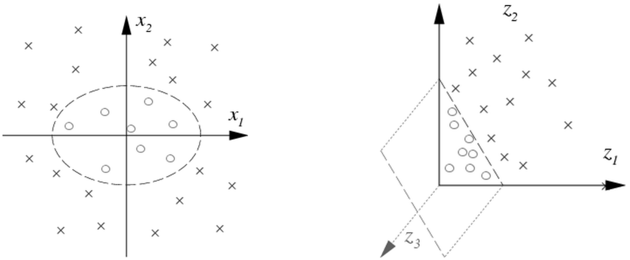
$$c((x_1, t_1, f(x_1)), \cdots, (x_N, t_N, f(x_N))) + \Omega(||f||_{\mathcal{H}})$$

admits a representation of the form

$$f(x) = \sum_{n=1}^{N} a_n k(x_n, x)$$

To be proved later ...

A toy example



Define $\phi([x]_1, [x]_2) = ([x]_1^2, [x]_2^2, \sqrt{2}[x]_1[x]_2)$ or
$\phi([x]_1, [x]_2) = ([x]_1^2, [x]_2^2, [x]_1[x]_2, [x]_2[x]_1)$ Then

$$\begin{aligned} \langle \phi(x), \phi(x') \rangle &= [x]_1^2[x']_1^2 + [x]_2^2[x']_2^2 + 2[x]_1[x]_2[x']_1[x']_2 \\ &= ([x]_1[x']_1 + [x]_2[x']_2)^2 \\ &= \langle x, x' \rangle^2 \end{aligned}$$

The dot product in the 3-dim space can be computed without computing ϕ.

More general case

Suppose the input vector dimension is M, and we define the feature mapping as to all the d-th order products (monomials) of $[x]_j$ of x

$$[x]_{j_1} \cdot [x]_{j_2} \cdots [x]_{j_d}$$

After mapping, the dimension becomes M^d. To compute the inner product, require at least $O(M^d)$ operations.

$$\begin{aligned}
\langle \phi_d(x), \phi_d(x') \rangle &= \sum_{j_1=1}^{M} \sum_{j_2=1}^{M} \cdots \sum_{j_d=1}^{M} [x]_{j_1} \cdots [x]_{j_d} \cdot [x']_{j_1} \cdots [x']_{j_d} \\
&= \sum_{j_1=1}^{M} [x]_{j_1} \cdot [x']_{j_1} \cdots \sum_{j_d=1}^{M} [x]_{j_d} [x']_{j_d} \\
&= \left(\sum_{j=1}^{M} [x]_j \cdot [x']_j \right)^d = \langle x, x' \rangle^d
\end{aligned}$$

Requires only $O(M)$ computation to get the inner product.

More general case

Suppose the input vector dimension is M, and we define the feature mapping as to all the d-th order products (monomials) of $[x]_j$ of x

$$[x]_{j_1} \cdot [x]_{j_2} \cdots [x]_{j_d}$$

After mapping, the dimension becomes M^d. To compute the inner product, require at least $O(M^d)$ operations.

$$\begin{aligned}
\langle \phi_d(x), \phi_d(x') \rangle &= \sum_{j_1=1}^{M} \sum_{j_2=1}^{M} \cdots \sum_{j_d=1}^{M} [x]_{j_1} \cdots [x]_{j_d} \cdot [x']_{j_1} \cdots [x']_{j_d} \\
&= \sum_{j_1=1}^{M} [x]_{j_1} \cdot [x']_{j_1} \cdots \sum_{j_d=1}^{M} [x]_{j_d} [x']_{j_d} \\
&= \left(\sum_{j=1}^{M} [x]_j \cdot [x']_j \right)^d = \langle x, x' \rangle^d
\end{aligned}$$

Requires only $O(M)$ computation to get the inner product.

Myths of Kernel

Kernel is a similarity measure

Kernel corresponds to dot products in feature space $\mathcal{H}$ via a mapping ϕ.

$$k(x, x') = \langle \phi(x), \phi(x') \rangle$$

Questions

1. What kind of kernel functions admits the above form?
2. Give a kernel, how to construct an associated feature space?

Myths of Kernel

Kernel is a similarity measure

Kernel corresponds to dot products in feature space $\mathcal{H}$ via a mapping ϕ.

$$k(x, x') = \langle \phi(x), \phi(x') \rangle$$

Questions

1. What kind of kernel functions admits the above form?
2. Give a kernel, how to construct an associated feature space?

Positive Definite Kernels

Gram Matrix

Given a function $k : \mathcal{X}^2 \rightarrow \mathcal{R}$, and input $x_1, \cdots x_N \in \mathcal{X}$, then the matrix

$$K_{ij} := k(x_i, x_j)$$

is called the Gram matrix.

Positive Definite Kernel

A function k on $\mathcal{X} \times \mathcal{X}$ which for any number of $x_1, x_2, \cdots, x_N \in \mathcal{X}$ gives rise to a positive semi-definite Gram matrix, is called a positive definite matrix.

A positive definite kernel can always be written as inner products of some feature mapping!

Positive Definite Kernels

Gram Matrix

Given a function $k : \mathcal{X}^2 \rightarrow \mathcal{R}$, and input $x_1, \cdots x_N \in \mathcal{X}$, then the matrix

$$K_{ij} := k(x_i, x_j)$$

is called the Gram matrix.

Positive Definite Kernel

A function k on $\mathcal{X} \times \mathcal{X}$ which for any number of $x_1, x_2, \cdots, x_N \in \mathcal{X}$ gives rise to a positive semi-definite Gram matrix, is called a positive definite matrix.

A positive definite kernel can always be written as inner products of some feature mapping!

Positive Definite Kernels

Gram Matrix

Given a function $k : \mathcal{X}^2 \rightarrow \mathcal{R}$, and input $x_1, \cdots x_N \in \mathcal{X}$, then the matrix

$$K_{ij} := k(x_i, x_j)$$

is called the Gram matrix.

Positive Definite Kernel

A function k on $\mathcal{X} \times \mathcal{X}$ which for any number of $x_1, x_2, \cdots, x_N \in \mathcal{X}$ gives rise to a positive semi-definite Gram matrix, is called a positive definite matrix.

A positive definite kernel can always be written as inner products of some feature mapping!

A Wake-Up Quiz

Cauchy-Schwartz Inequality for Kernels

If k is a positive definite kernel, then

$$|k(x_1, x_2)|^2 \leq k(x_1, x_1) \cdot k(x_1, x_2)$$

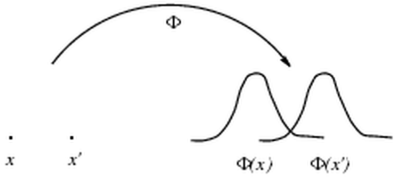
Reproducing Kernel Map

A positive definite kernel can always be written as inner products of some feature mapping!

The strategy to prove:

- Define a feature mapping ϕ into some vector space.
- Define a dot product (strictly a positive definite bilinear form)
- Show that $k(x, x') = \langle \phi(x), \phi(x') \rangle$

Feature Map



- Define a feature map ϕ from $\mathcal{X}$ to the space of functions:

$$\phi(x) = k(\cdot, x)$$

where $k(\cdot, x)$ denotes the function that assigns the value $k(x', x)$ to $x' \in \mathcal{X}$.

Vector Space

- Let the space be all the vectors that can be represented as the following form:

$$f(\cdot) = \sum_{i=1}^{m} \alpha_i k(\cdot, x_i)$$

Here $m \in \mathcal{N}$, $\alpha_i \in \mathcal{R}$ and $x_1, x_2, \cdots, x_m \in \mathcal{X}$ are arbitrary.

- We define the dot product as below:

$$g(\cdot) = \sum_{j=1}^{m'} \beta_j k(\cdot, x_j) \tag{2}$$

where $m' \in \mathcal{N}$, $\beta_j \in \mathcal{R}$, and $x'_1, x'_2, \cdots, x'_{m'} \in \mathcal{X}$. So

$$\langle f, g \rangle := \sum_{i=1}^{m} \sum_{j=1}^{m'} \alpha_i \beta_j k(x_i, x'_j)$$

- Need to show the above is a valid inner product.

Vector Space

- Let the space be all the vectors that can be represented as the following form:

$$f(\cdot) = \sum_{i=1}^{m} \alpha_i k(\cdot, x_i)$$

Here $m \in \mathcal{N}$, $\alpha_i \in \mathcal{R}$ and $x_1, x_2, \cdots, x_m \in \mathcal{X}$ are arbitrary.

- We define the dot product as below:

$$g(\cdot) = \sum_{j=1}^{m'} \beta_j k(\cdot, x_j) \quad (2)$$

where $m' \in \mathcal{N}$, $\beta_j \in \mathcal{R}$, and $x'_1, x'_2, \cdots, x'_{m'} \in \mathcal{X}$. So

$$\langle f, g \rangle := \sum_{i=1}^{m} \sum_{j=1}^{m'} \alpha_i \beta_j k(x_i, x'_j)$$

- Need to show the above is a valid inner product.

Review of Inner Product

Bilinear Form

A bilinear form on a vector space $\mathcal{H}$ is a function $Q : \mathcal{H} \times \mathcal{H} \rightarrow \mathcal{R}$ such that

$$Q((\lambda x + \lambda' x'), x'') = \lambda Q(x, x'') + \lambda' Q(x', x'')$$
$$Q(x'', (\lambda x + \lambda' x')) = \lambda Q(x'', x) + \lambda' Q(x', x'')$$

where $x, x', x'' \in \mathcal{X}$ and $\lambda, \lambda' \in \mathcal{R}$.
If $Q(x, x') = Q(x', x)$, then Q is a symmetric bilinear form.

Dot Product

A dot product on a vector space $\mathcal{H}$ is a symmetric bilinear form that is strictly positive definite; in other words, for all $x \in \mathcal{X}$, $\langle x, x \rangle \geq 0$, with equality only for $x = 0$.

Review of Inner Product

Bilinear Form

A bilinear form on a vector space $\mathcal{H}$ is a function $Q : \mathcal{H} \times \mathcal{H} \rightarrow \mathcal{R}$ such that

$$Q((\lambda x + \lambda' x'), x'') = \lambda Q(x, x'') + \lambda' Q(x', x'')$$
$$Q(x'', (\lambda x + \lambda' x')) = \lambda Q(x'', x) + \lambda' Q(x', x'')$$

where $x, x', x'' \in \mathcal{X}$ and $\lambda, \lambda' \in \mathcal{R}$.
If $Q(x, x') = Q(x', x)$, then Q is a symmetric bilinear form.

Dot Product

A dot product on a vector space $\mathcal{H}$ is a symmetric bilinear form that is strictly positive definite; in other words, for all $x \in \mathcal{X}$, $\langle x, x \rangle \geq 0$, with equality only for $x = 0$.

$$\langle f, g \rangle := \sum_{i=1}^{m} \sum_{j=1}^{m'} \alpha_i \beta_j k(x_i, x_j')$$

- It's bilinear as

$$\langle f, g \rangle = \sum_{j=1}^{m'} \beta_j f(x_j') \qquad \langle f, g \rangle = \sum_{i=1}^{m} \alpha_i g(x_i)$$

- It's symmetric as $\langle f, g \rangle = \langle g, f \rangle$.
- It's positive definite as

$$\langle f, f \rangle = \sum_{i,j=1}^{m} \alpha_i \alpha_j k(x_i, x_j) \geq 0 \qquad \text{(Definition of positive kernel)}$$

- Remains to show $\langle f, f \rangle = 0 \iff f = 0$.

$$\langle f, g\rangle := \sum_{i=1}^{m} \sum_{j=1}^{m'} \alpha_i \beta_j k(x_i, x'_j)$$

- It's bilinear as

$$\langle f, g\rangle = \sum_{j=1}^{m'} \beta_j f(x'_j) \qquad \langle f, g\rangle = \sum_{i=1}^{m} \alpha_i g(x_i)$$

- It's symmetric as $\langle f, g\rangle = \langle g, f\rangle$.
- It's positive definite as

$$\langle f, f\rangle = \sum_{i,j=1}^{m} \alpha_i \alpha_j k(x_i, x_j) \geq 0 \qquad \text{(Definition of positive kernel)}$$

- Remains to show $\langle f, f\rangle = 0 \iff f = 0$.

$$\langle f, g \rangle := \sum_{i=1}^{m} \sum_{j=1}^{m'} \alpha_i \beta_j k(x_i, x'_j)$$

- It's bilinear as

$$\langle f, g \rangle = \sum_{j=1}^{m'} \beta_j f(x'_j) \qquad \langle f, g \rangle = \sum_{i=1}^{m} \alpha_i g(x_i)$$

- It's symmetric as $\langle f, g \rangle = \langle g, f \rangle$.
- It's positive definite as

$$\langle f, f \rangle = \sum_{i,j=1}^{m} \alpha_i \alpha_j k(x_i, x_j) \geq 0 \qquad \text{(Definition of positive kernel)}$$

- Remains to show $\langle f, f \rangle = 0 \iff f = 0$.

$$\langle f, g \rangle := \sum_{i=1}^{m} \sum_{j=1}^{m'} \alpha_i \beta_j k(x_i, x'_j)$$

- It's bilinear as

$$\langle f, g \rangle = \sum_{j=1}^{m'} \beta_j f(x'_j) \qquad \langle f, g \rangle = \sum_{i=1}^{m} \alpha_i g(x_i)$$

- It's symmetric as $\langle f, g \rangle = \langle g, f \rangle$.
- It's positive definite as

$$\langle f, f \rangle = \sum_{i,j=1}^{m} \alpha_i \alpha_j k(x_i, x_j) \geq 0 \qquad \text{(Definition of positive kernel)}$$

- Remains to show $\langle f, f \rangle = 0 \iff f = 0$.

Reproducing Kernel

- $\langle k(\cdot, x), f\rangle = f(x)$
- $\langle k(\cdot, x), k(\cdot, x')\rangle = k(x, x')$ reproducing kernel property

So positive definite kernels k are also called reproducing kernels.

- Note that $\langle \cdot\cdot \rangle$ is a positive kernel in the space of functions as

$$\sum_{i,j=1} \gamma_i, \gamma_j \langle f_i, f_j\rangle = \langle \sum_{i=1}^{\gamma_i} f_i, \sum_{j=1}^{\gamma_j} f_j \rangle \geq 0$$

- Based on the result of our quiz, we have

$$|f(x)|^2 = |\langle k(\cdot, x), f\rangle|^2 \leq k(x, x) \cdot \langle f, f\rangle$$

So $\langle f, f\rangle = 0 \Longrightarrow f(x) = 0$.

Reproducing Kernel

- $\langle k(\cdot, x), f \rangle = f(x)$
- $\langle k(\cdot, x), k(\cdot, x') \rangle = k(x, x')$ reproducing kernel property

So positive definite kernels k are also called reproducing kernels.

- Note that $\langle \cdot \cdot \rangle$ is a positive kernel in the space of functions as

$$\sum_{i,j=1} \gamma_i, \gamma_j \langle f_i, f_j \rangle = \langle \sum_{i=1}^{\gamma_i} f_i, \sum_{j=1}^{\gamma_j} f_j \rangle \geq 0$$

- Based on the result of our quiz, we have

$$|f(x)|^2 = |\langle k(\cdot, x), f \rangle|^2 \leq k(x, x) \cdot \langle f, f \rangle$$

So $\langle f, f \rangle = 0 \Longrightarrow f(x) = 0$.

- $\langle k(\cdot, x), f \rangle = f(x)$
- $\langle k(\cdot, x), k(\cdot, x') \rangle = k(x, x')$ reproducing kernel property

So positive definite kernels k are also called reproducing kernels.

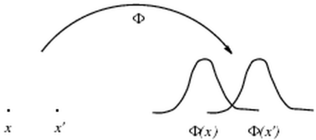
- Note that $\langle \cdot \cdot \rangle$ is a positive kernel in the space of functions as

$$\sum_{i,j=1} \gamma_i, \gamma_j \langle f_i, f_j \rangle = \langle \sum_{i=1}^{\gamma_i} f_i, \sum_{j=1}^{\gamma_j} f_j \rangle \geq 0$$

- Based on the result of our quiz, we have

$$|f(x)|^2 = |\langle k(\cdot, x), f \rangle|^2 \leq k(x, x) \cdot \langle f, f \rangle$$

So $\langle f, f \rangle = 0 \implies f(x) = 0.$



- Define a feature map ϕ from $\mathcal{X}$ to the space of functions:

$$\phi(x) = k(\cdot, x)$$

  where $k(\cdot, x)$ denotes the function that assigns the value $k(x', x)$ to $x' \in \mathcal{X}$.
- Any positive definite kernel can be thought of as a dot product in another space.
- Here, our proof is one possible instantiation of the feature space associated with a kernel, but not unique.

Reproducing Kernel Hilbert Spaces (RKHS)

- In previous example, the space of functions is a dot product space, or equivalently pre-Hilbert space.
- Hilbert space is generalizes the notion of Euclidean space in a way that extends methods of vector algebra from the two-dimensional plane and three-dimensional space to infinite-dimensional spaces.
  - A Hilbert space is an inner product space an abstract vector space in which distances and angles can be measured.
  - Hilbert space is "complete", meaning that if a sequence of vectors approaches a limit, then that limit is guaranteed to be in the space as well.

RKHS

Let $\mathcal{X}$ be a nonempty set (often called index set) and $\mathcal{H}$ a Hilbert space of functions $f : \mathcal{X} \to \mathcal{R}$, Then $\mathcal{H}$ is called a reproducing kernel Hilbert space endowed with the dot product $\langle \cdot, \cdot \rangle$ (and the norm $||f|| := \sqrt{\langle f, f \rangle}$) if there exists a function $k : \mathcal{X} \times \mathcal{X} \to \mathcal{R}$ with the following properties:

1. k has reproducing property: $\langle f, k(x, \cdot) \rangle = f(x)$ for all $f \in \mathcal{H}$; In particular, $k(x, \cdot), k(x', \cdot)\rangle = k(x, x')$
2. k spans $\mathcal{H}$.

RKHS uniquely determines k

Assume two different kernels k and k', we have

$$\langle k(x, \cdot), k'(x', \cdot) \rangle = k(x, x') = k'(x', x)$$

Contradiction!

Reproducing Kernel Hilbert Spaces (RKHS)

RKHS

Let $\mathcal{X}$ be a nonempty set (often called index set) and $\mathcal{H}$ a Hilbert space of functions $f : \mathcal{X} \rightarrow \mathcal{R}$, Then $\mathcal{H}$ is called a reproducing kernel Hilbert space endowed with the dot product $\langle \cdot, \cdot \rangle$ (and the norm $||f|| := \sqrt{\langle f, f \rangle}$) if there exists a function $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{R}$ with the following properties:

1. k has reproducing property: $\langle f, k(x, \cdot) \rangle = f(x)$ for all $f \in \mathcal{H}$; In particular, $k(x, \cdot), k(x', \cdot)\rangle = k(x, x')$
2. k spans $\mathcal{H}$.

RKHS uniquely determines k

Assume two different kernels k and k', we have

$$\langle k(x, \cdot), k'(x', \cdot)\rangle = k(x, x') = k'(x', x)$$

Contradiction!

Mercer's Theorem

If k is a continuous kernel of a positive definite integral operator on $L_2(\mathcal{X})$ (where $\mathcal{X}$ is some compact space),

$$\int_{\mathcal{X}} k(x,x')f(x)f(x')\,dx\,dx' \geq 0,$$

it can be expanded as

$$k(x,x') = \sum_{i=1}^{\infty} \lambda_i \psi_i(x)\psi_i(x')$$

using eigenfunctions ψ_i and eigenvalues $\lambda_i \geq 0$ [34].

In that case

$$\Phi(x) := \begin{pmatrix} \sqrt{\lambda_1}\psi_1(x) \\ \sqrt{\lambda_2}\psi_2(x) \\ \vdots \end{pmatrix}$$

satisfies $\langle \Phi(x), \Phi(x') \rangle = k(x,x')$.

- Define another feature mapping from x to a function (an integral operator) Hilbert space
- Then, the kernel is decomposed as the summation of the eigenfunctions.
- It turns out Mercer's kernel map is also positive definite.

Too complicated to understand. So we skip the details...

Kernel Trick

Given an algorithm which is formulated in terms of a positive kernel (or inner products), one can construct an alternative algorithm by replacing k by another positive definite kernel $\hat{k}$.

Examples of Kernels

- Linear kernel: $k(x, x') = x^T x'$
- Polynomial: $k(x, x') = < x, x' >^d$
- Inhomogeneous Polynomial: $k(x, x') = (< x, x' > + c)^d$
- Gaussian: $k(x, x') = exp\left(-\frac{||x-x'||^2}{2\sigma^2}\right)$
- ......

Kernel Trick

Given an algorithm which is formulated in terms of a positive kernel (or inner products), one can construct an alternative algorithm by replacing k by another positive definite kernel $\hat{k}$.

Examples of Kernels

- Linear kernel: $k(x, x') = x^T x'$
- Polynomial: $k(x, x') = < x, x' >^d$
- Inhomogeneous Polynomial: $k(x, x') = (< x, x' > + c)^d$
- Gaussian: $k(x, x') = exp\left(-\frac{||x-x'||^2}{2\sigma^2}\right)$
- ......

Constructing Kernels

A valid kernel should positive definite or can be written as the inner product in some feature space.

Given valid kernels $k_1(\mathbf{x}, \mathbf{x}')$ and $k_2(\mathbf{x}, \mathbf{x}')$, the following new kernels will also be valid:

$$k(\mathbf{x}, \mathbf{x}') = c k_1(\mathbf{x}, \mathbf{x}') \tag{6.13}$$

$$k(\mathbf{x}, \mathbf{x}') = f(\mathbf{x}) k_1(\mathbf{x}, \mathbf{x}') f(\mathbf{x}') \tag{6.14}$$

$$k(\mathbf{x}, \mathbf{x}') = q\left(k_1(\mathbf{x}, \mathbf{x}')\right) \tag{6.15}$$

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(k_1(\mathbf{x}, \mathbf{x}')\right) \tag{6.16}$$

$$k(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}') \tag{6.17}$$

$$k(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}') k_2(\mathbf{x}, \mathbf{x}') \tag{6.18}$$

$$k(\mathbf{x}, \mathbf{x}') = k_3\left(\boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{x}')\right) \tag{6.19}$$

$$k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^{\mathrm{T}} \mathbf{A} \mathbf{x}' \tag{6.20}$$

$$k(\mathbf{x}, \mathbf{x}') = k_a(\mathbf{x}_a, \mathbf{x}'_a) + k_b(\mathbf{x}_b, \mathbf{x}'_b) \tag{6.21}$$

$$k(\mathbf{x}, \mathbf{x}') = k_a(\mathbf{x}_a, \mathbf{x}'_a) k_b(\mathbf{x}_b, \mathbf{x}'_b) \tag{6.22}$$

where $c > 0$ is a constant, $f(\cdot)$ is any function, $q(\cdot)$ is a polynomial with nonnegative coefficients, $\boldsymbol{\phi}(\mathbf{x})$ is a function from $\mathbf{x}$ to $\mathbb{R}^M$, $k_3(\cdot, \cdot)$ is a valid kernel in $\mathbb{R}^M$, $\mathbf{A}$ is a symmetric positive semidefinite matrix, $\mathbf{x}_a$ and $\mathbf{x}_b$ are variables (not necessarily disjoint) with $\mathbf{x} = (\mathbf{x}_a, \mathbf{x}_b)$, and k_a and k_b are valid kernel functions over their respective spaces.

Gaussian Kernel

The Gaussian Kernel

$$k(x, x') = exp(-\frac{||x - x'||^2}{2\sigma^2})$$

is a valid kernel.

$$\begin{aligned} k(x, x') &= exp(-\frac{x^T x}{2\sigma^2}) exp(\frac{x^T x'}{\sigma^2}) exp(-\frac{x'^T x'}{2\sigma^2}) \\ &= f(x) exp(x^T x' / \sigma^2) f(x') \end{aligned}$$

Quiz

Show the feature vector that corresponds to the Gaussian kernel has infinite dimensionality.

Gaussian Kernel

The Gaussian Kernel

$$k(x, x') = exp(-\frac{||x - x'||^2}{2\sigma^2})$$

is a valid kernel.

$$\begin{aligned} k(x, x') &= exp(-\frac{x^T x}{2\sigma^2}) exp(\frac{x^T x'}{\sigma^2}) exp(-\frac{x'^T x'}{2\sigma^2}) \\ &= f(x) exp(x^T x'/\sigma^2) f(x') \end{aligned}$$

Quiz

Show the feature vector that corresponds to the Gaussian kernel has infinite dimensionality.

Gaussian Kernel

The Gaussian Kernel

$$k(x, x') = exp(-\frac{||x - x'||^2}{2\sigma^2})$$

is a valid kernel.

$$\begin{aligned} k(x, x') &= exp(-\frac{x^T x}{2\sigma^2}) exp(\frac{x^T x'}{\sigma^2}) exp(-\frac{x'^T x'}{2\sigma^2}) \\ &= f(x) exp(x^T x' / \sigma^2) f(x') \end{aligned}$$

Quiz

Show the feature vector that corresponds to the Gaussian kernel has infinite dimensionality.

Kernel and Distance

- As kernel is considered the similarity, we can calculate distance based on kernels.

$$\begin{aligned} ||x - x'||^2 &= <x, x> + <x', x'> -2 <x, x'> \\ &= k(x, x) + k(x', x') - 2k(x, x') \end{aligned}$$

- Gaussian Kernel can be extended to other distance measure instead of Euclidean distance.

$$k(x, x') = exp\left\{ -\frac{1}{2\sigma^2} \left(k(x, x) + k(x', x') - 2k(x, x') \right) \right\}$$

Kernels over sets

- Kernels extend to input that are symbolic, rather than simply vectors of real numbers.
- Kernels can be defined over objects as graphs, sets, strings, and text documents.
- A toy example, a fixed set and define a nonvectorial space consisting of all possible subsets of this set. If A_1 and A_2 are two such subsets, then one simple choice of kernel would be

$$k(A_1, A_2) = 2^{|A_1 \cap A_2|}$$

Quiz: Show this is a valid kernel.

Kernels over sets

- Kernels extend to input that are symbolic, rather than simply vectors of real numbers.
- Kernels can be defined over objects as graphs, sets, strings, and text documents.
- A toy example, a fixed set and define a nonvectorial space consisting of all possible subsets of this set. If A_1 and A_2 are two such subsets, then one simple choice of kernel would be

$$k(A_1, A_2) = 2^{|A_1 \cap A_2|}$$

Quiz: Show this is a valid kernel.

Kernels to connect generative/discriminative models(1)

- Generative models can naturally handle missing data and varying length in the case of hidden Markov models.
- Discriminative models perform better on discriminative tasks
- One way to combine them is to use a generative model to define a kernel and then use this kernel in a discriminative approach.
- One example:

$$k(x, x') = p(x)p(x')$$

  Two inputs are similar if they both have higher probabilities.

Kernels to connect generative/discriminative models(2)

- Two inputs are similar if they ave significant probability under a range of different components.

$$k(x, x') = \int p(x|z)p(x'|z)p(z)dz$$

where z is the latent variable.

- Suppose data consists of ordered sequence of length L, so an observation is

$$X = \{x_1, \cdots, x_L\}$$

- Hidden states $Z = \{z_1, \cdots, z_L\}$
- $K(X, X') = \sum_Z P(X|Z)P(X'|Z)P(Z)$
- This model can be easily extended to allow sequence of different length to be compared.

Fisher Kernel

- Consider the gradient with respect to θ, which defines a vector in a 'feature' space having the same dimensionality as θ.
- Fisher score:
$$g(\theta, x) = \nabla_\theta \ln p(x|\theta)$$
- Fisher kernel is defined by
$$k(x, x') = g(\theta, x)^t F^{-1} g(\theta, x') \quad (3)$$
where F is the *Fisher information matrix*, given by
$$F = E_x[g(\theta, x) g(\theta, x)^T] \quad (4)$$
- Empirically, F is estimated by the sample average, which corresponds to the covariance matrix of the Fisher scores.
- Has been applied to document retrieval.

Sigmodial Kernel

$$k(x, x') = tanh(ax^T x' + b)$$

- Its Gram matrix in general is not positive semidefinite, thus it's a invalid kernel.
- It gives SVM a superficial resemblance to neural network models.
- A Bayesian neural network with appropriate prior reduces to a Gaussian process. We'll discuss next time.

Radial Basis Function Network

- Regression based on a fixed basis functions.
- Radial basis function, which have the property that each basis function depends only on the radial distance (typically Euclidean) from a centre μ_j, so

$$\phi_j(x) = h(||x - \mu_j||)$$

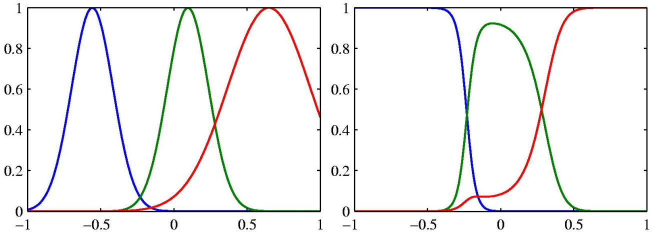
- Historically, radial basis functions were introduced for exact function interpolation.

$$f(x) = \sum_{n=1}^{N} w_n h(||x - x_n||) \tag{5}$$

- Same number of coefficients and constraints, the result will fit every target value exactly. Over-fitting!
- Motivation from other perspectives: regularization theory, noisy inputs.

Radial Basis Function Network

- Normalization might be required in practice.



- How to choose data point with large scale of training data?
  - Randomly choose subsets of data points
  - Orthogonal least squares: a sequential selection process in which each step the next data point to be chosen as a basis function entry corresponds to the one that gives the greatest reduction in the error.
- The same problem as Reduced SVM.

Nadaraya-Watson model

- Parzen density estimator to model the joint distribution $p(x,t)$

$$p(x,t) = \frac{1}{N}\sum_{n=1}^{N} f(x - x_n, t - t_n) \tag{6}$$

where $f(x,t)$ is the component density function and one component on each data point.

$$\begin{aligned} y(\mathbf{x}) &= \mathbb{E}[t|\mathbf{x}] = \int_{-\infty}^{\infty} tp(t|\mathbf{x})\,\mathrm{d}t \\ &= \frac{\displaystyle\int tp(\mathbf{x},t)\,\mathrm{d}t}{\displaystyle\int p(\mathbf{x},t)\,\mathrm{d}t} \\ &= \frac{\displaystyle\sum_n \int tf(\mathbf{x}-\mathbf{x}_n, t-t_n)\,\mathrm{d}t}{\displaystyle\sum_m \int f(\mathbf{x}-\mathbf{x}_m, t-t_m)\,\mathrm{d}t}. \end{aligned}$$

We now assume for simplicity that the component density functions have zero mean so that

$$\int_{-\infty}^{\infty} f(\mathbf{x}, t) t \, \mathrm{d}t = 0 \tag{6.44}$$

for all values of $\mathbf{x}$. Using a simple change of variable, we then obtain

$$\begin{aligned} y(\mathbf{x}) &= \frac{\sum_n g(\mathbf{x} - \mathbf{x}_n) t_n}{\sum_m g(\mathbf{x} - \mathbf{x}_m)} \\ &= \sum_n k(\mathbf{x}, \mathbf{x}_n) t_n \end{aligned} \tag{6.45}$$

where $n, m = 1, \ldots, N$ and the kernel function $k(\mathbf{x}, \mathbf{x}_n)$ is given by

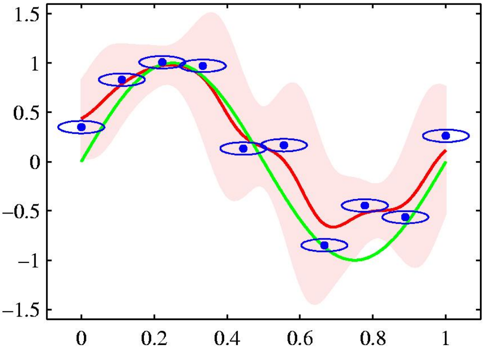
$$k(\mathbf{x}, \mathbf{x}_n) = \frac{g(\mathbf{x} - \mathbf{x}_n)}{\sum_m g(\mathbf{x} - \mathbf{x}_m)} \tag{6.46}$$

and we have defined

$$g(\mathbf{x}) = \int_{-\infty}^{\infty} f(\mathbf{x}, t) \, \mathrm{d}t. \tag{6.47}$$

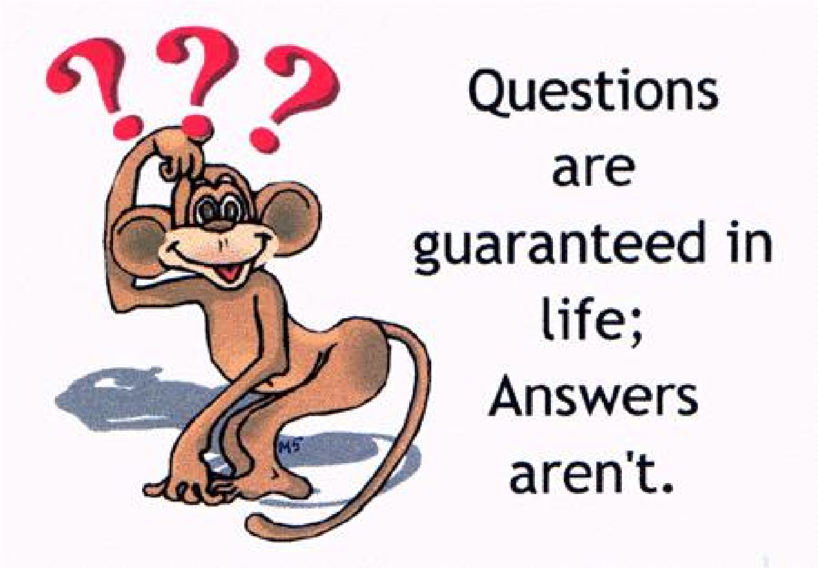
- The result $y(x) = \sum_n k(x, x_n)t_n$ is known as *Nadaraya-Watson* model or *kernel regression*.
- Notice that $\sum_{n=1}^{N} k(x, x_n) = 1$.
- The conditional probability can be calculated as

$$p(t|\mathbf{x}) = \frac{p(t, \mathbf{x})}{\int p(t, \mathbf{x})\,\mathrm{d}t} = \frac{\sum_n f(\mathbf{x} - \mathbf{x}_n, t - t_n)}{\sum_m \int f(\mathbf{x} - \mathbf{x}_m, t - t_m)\,\mathrm{d}t}$$


1.5
1
0.5
0
−0.5
−1
−1.5
0
0.2
0.4
0.6
0.8
1

Summary

- Dual Representation
- Kernel
- How to construct a kernel
- Various Kernels
- Radial Basis Functions
- Gaussian Process


Questions
are
guaranteed in
life;
Answers
aren't.