

# Kernel Methods

---

David S. Rosenberg

Bloomberg ML EDU

October 26, 2017

## Setup and Motivation

---

# The Input Space $\mathcal{X}$

- Our general learning theory setup: no assumptions about  $\mathcal{X}$
- But  $\mathcal{X} = \mathbf{R}^d$  for the specific methods we've developed:
  - Ridge regression
  - Lasso regression
  - Support Vector Machines
- Our hypothesis space for these was all affine functions on  $\mathbf{R}^d$ :

$$\mathcal{H} = \{x \mapsto w^T x + b \mid w \in \mathbf{R}^d, b \in \mathbf{R}\}.$$

- What if we want to do prediction on inputs not natively in  $\mathbf{R}^d$ ?

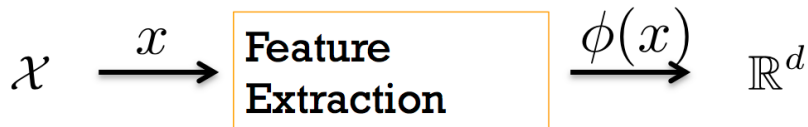
# Feature Extraction

## Definition

Mapping an input from  $\mathcal{X}$  to a vector in  $\mathbb{R}^d$  is called **feature extraction** or **featurization**.

**Raw Input**

**Feature Vector**

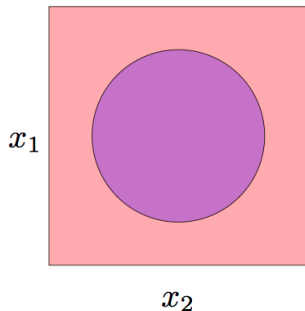


# Linear Models with Explicit Feature Map

- Input space:  $\mathcal{X}$  (no assumptions)
- Introduce **feature map**  $\psi : \mathcal{X} \rightarrow \mathbf{R}^d$
- The feature map maps into the **feature space**  $\mathbf{R}^d$ .
- Hypothesis space of affine functions on feature space:

$$\mathcal{H} = \{x \mapsto w^T \psi(x) + b \mid w \in \mathbf{R}^d, b \in \mathbf{R}\}.$$

## Geometric Example: Two class problem, nonlinear boundary



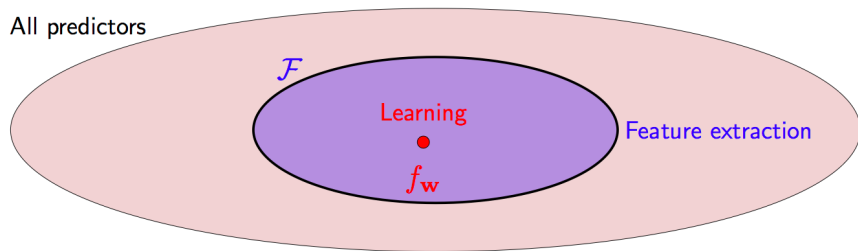
- With linear feature map  $\phi(x) = (x_1, x_2)$  and linear models, can't separate regions
- With appropriate nonlinearity  $\phi(x) = (x_1, x_2, x_1^2 + x_2^2)$ , piece of cake.
- Video: <http://youtu.be/3liCbRZPrZA>

From Percy Liang's "Lecture 3" slides from Stanford's CS221, Autumn 2014.

# Expressivity of Hypothesis Space

- Consider a linear hypothesis space with a feature map  $\phi : \mathcal{X} \rightarrow \mathbb{R}^d$ :

$$\mathcal{F} = \{f(x) = w^T \phi(x)\}$$



**Question: does  $\mathcal{F}$  contain a good predictor?**

We can grow the linear hypothesis space  $\mathcal{F}$  by adding more features.

From Percy Liang's "Lecture 3" slides from Stanford's CS221, Autumn 2014.

# Linear Models Need Big Feature Spaces

- To get **expressive** hypothesis spaces using linear models,
  - need high-dimensional feature spaces
- Suppose we start with  $x = (1, x_1, \dots, x_d) \in \mathbf{R}^{d+1} = \mathcal{X}$ .
- We want to add all monomials up to degree  $M$ :  $x_1^{p_1} \cdots x_d^{p_d}$ , with  $p_1 + \cdots + p_d \leq M$ .
- How many features will we end up with?
- $\binom{M+d}{M}$  (“flower shop problem” from combinatorics)
- For  $d = 40$  and  $M = 8$ , we get 377348994 features.
- That will make some extremely large matrices...



Very large feature spaces have two problems:

- ① Overfitting
  - ② Memory and computational costs
- Overfitting we handle with regularization.
  - “Kernel methods” can (sometimes) help with memory and computational costs.

## Kernel Methods: Motivation

## Review: Linear SVM and Dual

- The [featurized] SVM prediction function is the solution to

$$\min_{w \in \mathbf{R}^d, b \in \mathbf{R}} \frac{1}{2} \|w\|^2 + \frac{c}{n} \sum_{i=1}^n (1 - y_i [w^T \psi(x_i) + b])_+.$$

- Found it is equivalent to solve the dual problem to get  $\alpha^*$ :

$$\begin{aligned} \sup_{\alpha} \quad & \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \psi(x_j)^T \psi(x_i) \\ \text{s.t.} \quad & \sum_{i=1}^n \alpha_i y_i = 0 \\ & \alpha_i \in \left[0, \frac{c}{n}\right] \quad i = 1, \dots, n. \end{aligned}$$

- Notice:  $\psi(x)$ 's only show up as inner products with other  $x$ 's.

# Some Methods Can Be “Kernelized”

## Definition

A method is **kernelized** if inputs only appear inside inner products:  $\langle \psi(x), \psi(x') \rangle$  for  $x, x' \in \mathcal{X}$ .

- The **kernel function** corresponding to  $\psi$  and inner product  $\langle \cdot, \cdot \rangle$  is

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

- Why introduce this new notation  $k(x, x')$ ?
- Turns out, we can often evaluate  $k(x, x')$  directly,
  - without explicitly computing  $\psi(x)$  and  $\psi(x')$ .
- For large feature spaces, can be much faster.

# Kernel Evaluation Can Be Fast

## Example

Quadratic feature map

$$\phi(x) = (x_1, \dots, x_d, x_1^2, \dots, x_d^2, \sqrt{2}x_1x_2, \dots, \sqrt{2}x_1x_d, \dots, \sqrt{2}x_{d-1}x_d)^T$$

has dimension  $O(d^2)$ , but for any  $x^{(1)}, x^{(2)} \in \mathbb{R}^d$

$$k(x^{(1)}, x^{(2)}) = \langle \phi(x^{(1)}), \phi(x^{(2)}) \rangle = \langle x^{(1)}, x^{(2)} \rangle + \langle x^{(1)}, x^{(2)} \rangle^2$$

- Naively explicit computation of  $k(w, x)$ :  $O(d^2)$
- Implicit computation of  $k(w, x)$ :  $O(d)$

- Often useful to think of the kernel function as a **similarity score**.
- But this is not a mathematically precise statement.
- There are many ways to design a similarity score.
  - We will use **Mercer kernels**, which correspond to inner products in some feature space.
  - Has many mathematical benefits.

# What are the Benefits of Kernelization?

- ① Computational (e.g. when feature space dimension  $d$  larger than sample size  $n$ ).
- ② Access to infinite-dimensional feature spaces.
- ③ Allows thinking in terms of “similarity” rather than features.

## Example: SVM

---



- Recall the SVM dual optimization problem

$$\begin{aligned} \sup_{\alpha} \quad & \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j x_j^T x_i \\ \text{s.t.} \quad & \sum_{i=1}^n \alpha_i y_i = 0 \\ & \alpha_i \in \left[0, \frac{C}{n}\right] \quad i = 1, \dots, n. \end{aligned}$$

- Can replace  $x_j^T x_i$  by an arbitrary kernel  $k(x_j, x_i)$ .
- What kernel are we currently using?

- Input space:  $\mathcal{X} = \mathbf{R}^d$
- Feature space:  $\mathcal{H} = \mathbf{R}^d$ , with standard inner product
- Feature map

$$\psi(x) = x.$$

- Kernel:

$$k(x_1, x_2) = x_1^T x_2$$

# The Kernel Matrix (or the Gram Matrix)

## Definition

For a set of  $\{x_1, \dots, x_n\}$  and an inner product  $\langle \cdot, \cdot \rangle$  on the set, the **kernel matrix** or the **Gram matrix** is defined as

$$K = (\langle x_i, x_j \rangle)_{i,j} = \begin{pmatrix} \langle x_1, x_1 \rangle & \cdots & \langle x_1, x_n \rangle \\ \vdots & \ddots & \vdots \\ \langle x_n, x_1 \rangle & \cdots & \langle x_n, x_n \rangle \end{pmatrix}.$$

Then for the standard Euclidean inner product  $\langle x_i, x_j \rangle = x_i^T x_j$ , we have

$$K = XX^T$$

# SVM Dual with Kernel Matrix

$$\begin{aligned} \sup_{\alpha} \quad & \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j K_{ji} \\ \text{s.t.} \quad & \sum_{i=1}^n \alpha_i y_i = 0 \\ & \alpha_i \in \left[0, \frac{C}{n}\right] \quad i = 1, \dots, n. \end{aligned}$$

- Once our algorithm works with kernel matrices, we can change kernel just by changing the matrix.
- Size of matrix:  $n \times n$ , where  $n$  is the number of data points.
- Recall with ridge regression, we worked with  $X^T X$ , which is  $d \times d$ , where  $d$  is feature space dimension.

## Some Nonlinear Kernels

## Quadratic Kernel in $\mathbf{R}^d$

- Input space  $\mathcal{X} = \mathbf{R}^d$
- Feature space:  $\mathcal{H} = \mathbf{R}^D$ , where  $D = d + \binom{d}{2} \approx d^2/2$ .
- Feature map:

$$\phi(x) = (x_1, \dots, x_d, x_1^2, \dots, x_d^2, \sqrt{2}x_1x_2, \dots, \sqrt{2}x_ix_j, \dots, \sqrt{2}x_{d-1}x_d)^T$$

- Then for  $\forall w, x \in \mathbf{R}^d$

$$\begin{aligned} k(w, x) &= \langle \phi(w), \phi(x) \rangle \\ &= \langle w, x \rangle + \langle w, x \rangle^2 \end{aligned}$$

- Computation for inner product with explicit mapping:  $O(d^2)$
- Computation for implicit kernel calculation:  $O(d)$ .

## Polynomial Kernel in $\mathbf{R}^d$

- Input space  $\mathcal{X} = \mathbf{R}^d$
- Kernel function:

$$k(w, x) = (1 + \langle w, x \rangle)^M$$

- Corresponds to a feature map with all monomials up to degree  $M$ .
- For any  $M$ , computing the kernel has same computational cost
- Cost of explicit inner product computation grows rapidly in  $M$ .

# Radial Basis Function (RBF) / Gaussian Kernel

- Input space  $\mathcal{X} = \mathbf{R}^d$ .  $\forall w, x \in \mathbf{R}^d$ ,

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

where  $\sigma^2$  is known as the bandwidth parameter.

- Does it act like a similarity score?
- Why “radial”?
- Have we departed from our “inner product of feature vector” recipe?
  - Yes and no: corresponds to an infinite dimensional feature vector
- Probably the most common nonlinear kernel.



## Kernel Trick: Overview

# The “Kernel Trick”

- ➊ Given a kernelized ML algorithm.
- ➋ Can swap out the inner product for a new kernel function.
- ➌ New kernel may correspond to a high dimensional feature space.
- ➍ Once kernel matrix is computed, computational cost depends on number of data points, rather than the dimension of feature space.

Swapping out a linear kernel for a new kernel is called the **kernel trick**.

## Inner Product Spaces and Projections (Hilbert Spaces)

---

# Inner Product Space (or “Pre-Hilbert” Spaces)

An **inner product space** (over reals) is a vector space  $\mathcal{V}$  and an **inner product**, which is a mapping

$$\langle \cdot, \cdot \rangle : \mathcal{V} \times \mathcal{V} \rightarrow \mathbf{R}$$

that has the following properties  $\forall x, y, z \in \mathcal{V}$  and  $a, b \in \mathbf{R}$ :

- Symmetry:  $\langle x, y \rangle = \langle y, x \rangle$
- Linearity:  $\langle ax + by, z \rangle = a \langle x, z \rangle + b \langle y, z \rangle$
- Positive-definiteness:  $\langle x, x \rangle \geq 0$  and  $\langle x, x \rangle = 0 \iff x = 0$ .

## Norm from Inner Product

For an inner product space, we define a norm as

$$\|x\| = \sqrt{\langle x, x \rangle}.$$

### Example

$\mathbf{R}^d$  with standard Euclidean inner product is an inner product space:

$$\langle x, y \rangle := x^T y \quad \forall x, y \in \mathbf{R}^d.$$

Norm is

$$\|x\| = \sqrt{x^T x}.$$

# What norms can we get from an inner product?

## Theorem (Parallelogram Law)

A norm  $\|v\|$  can be generated by an inner product on  $\mathcal{V}$  iff  $\forall x, y \in \mathcal{V}$

$$2\|x\|^2 + 2\|y\|^2 = \|x + y\|^2 + \|x - y\|^2,$$

and if it can, the inner product is given by the **polarization identity**

$$\langle x, y \rangle = \frac{\|x\|^2 + \|y\|^2 - \|x - y\|^2}{2}.$$

## Example

$\ell_1$  norm on  $\mathbf{R}^d$  is NOT generated by an inner product. [Exercise]

Is  $\ell_2$  norm on  $\mathbf{R}^d$  generated by an inner product?

# Pythagorean Theorem

## Definition

Two vectors are **orthogonal** if  $\langle x, y \rangle = 0$ . We denote this by  $x \perp y$ .

## Definition

$x$  is orthogonal to a set  $S$ , i.e.  $x \perp S$ , if  $x \perp s$  for all  $s \in S$ .

## Theorem (Pythagorean Theorem)

If  $x \perp y$ , then  $\|x + y\|^2 = \|x\|^2 + \|y\|^2$ .

## Proof.

We have

$$\begin{aligned}\|x + y\|^2 &= \langle x + y, x + y \rangle \\ &= \langle x, x \rangle + \langle x, y \rangle + \langle y, x \rangle + \langle y, y \rangle \\ &= \|x\|^2 + \|y\|^2\end{aligned}$$

## Projection onto a Plane (Rough Definition)

- Choose some  $x \in \mathcal{V}$ .
- Let  $M$  be a subspace of inner product space  $\mathcal{V}$ .
- Then  $m_0$  is the **projection of  $x$  onto  $M$** ,
  - if  $m_0 \in M$  and is the closest point to  $x$  in  $M$ .
- In math: For all  $m \in M$ ,

$$\|x - m_0\| \leq \|x - m\|.$$



# Hilbert Space

- Projections exist for all finite-dimensional inner product spaces.
- We want to allow infinite-dimensional spaces.
- Need an extra condition called **completeness**.
- A space is **complete** if all Cauchy sequences in the space converge.

## Definition

A **Hilbert space** is a complete inner product space.

## Example

Any finite dimensional inner product space is a Hilbert space.

# The Projection Theorem

## Theorem (Classical Projection Theorem)

- $\mathcal{H}$  a Hilbert space
- $M$  a closed subspace of  $\mathcal{H}$  (picture a hyperplane through the origin)
- For any  $x \in \mathcal{H}$ , there **exists a unique**  $m_0 \in M$  for which

$$\|x - m_0\| \leq \|x - m\| \quad \forall m \in M.$$

- This  $m_0$  is called the **[orthogonal] projection of  $x$  onto  $M$** .
- Furthermore,  $m_0 \in M$  is the projection of  $x$  onto  $M$  iff

$$x - m_0 \perp M.$$

# Projection Reduces Norm

## Theorem

Let  $M$  be a closed subspace of  $\mathcal{H}$ . For any  $x \in \mathcal{H}$ , let  $m_0 = \text{Proj}_M x$  be the projection of  $x$  onto  $M$ . Then

$$\|m_0\| \leq \|x\|,$$

with equality only when  $m_0 = x$ .

## Proof.

$$\begin{aligned}\|x\|^2 &= \|m_0 + (x - m_0)\|^2 \text{ (note: } x - m_0 \perp m_0 \text{ by Projection theorem)} \\ &= \|m_0\|^2 + \|x - m_0\|^2 \text{ by Pythagorean theorem} \\ \|m_0\|^2 &= \|x\|^2 - \|x - m_0\|^2\end{aligned}$$

If  $\|x - m_0\|^2 = 0$ , then  $x = m_0$ , by definition of norm. □

# Representer Theorem

---

## Generalize from SVM Objective

- Featurized SVM objective:

$$\min_{w \in \mathbf{R}^d} \frac{1}{2} \|w\|^2 + \frac{c}{n} \sum_{i=1}^n (1 - y_i [\langle w, \psi(x_i) \rangle])_+.$$

- Generalized objective:

$$\min_{w \in \mathcal{H}} R(\|w\|) + L(\langle w, \psi(x_1) \rangle, \dots, \langle w, \psi(x_n) \rangle),$$

where

- $R: \mathbf{R}^{\geq 0} \rightarrow \mathbf{R}$  is nondecreasing (**Regularization term**)
- and  $L: \mathbf{R}^n \rightarrow \mathbf{R}$  is arbitrary. (**Loss term**)

# Generalized Linear Objective Function (Details)

- Generalized objective:

$$\min_{w \in \mathcal{H}} R(\|w\|) + L(\langle w, \psi(x_1) \rangle, \dots, \langle w, \psi(x_n) \rangle),$$

where

- $w, \psi(x_1), \dots, \psi(x_n) \in \mathcal{H}$  for some Hilbert space  $\mathcal{H}$ . (We typically have  $\mathcal{H} = \mathbf{R}^d$ .)
- $\|\cdot\|$  is the norm corresponding to the inner product of  $\mathcal{H}$ . (i.e.  $\|w\| = \sqrt{\langle w, w \rangle}$ )
- $R: \mathbf{R}^{\geq 0} \rightarrow \mathbf{R}$  is nondecreasing (**Regularization term**), and
- $L: \mathbf{R}^n \rightarrow \mathbf{R}$  is arbitrary (**Loss term**).

# Generalized Linear Objective Function

- **Generalized objective:**

$$\min_{w \in \mathcal{H}} R(\|w\|) + L(\langle w, \psi(x_1) \rangle, \dots, \langle w, \psi(x_n) \rangle),$$

- Why “linear”?  $\langle w, \psi(x_i) \rangle$  is a generalization of predictions  $w^T \psi(x_i)$ 
  - a linear function of  $\psi(x_i) \in \mathbf{R}^d$ .
- Ridge regression and SVM are of this form.
- What if we penalize with  $\lambda \|w\|_2$  instead of  $\lambda \|w\|_2^2$ ? Yes!.
- What if we use lasso regression? No!  $\ell_1$  norm does not correspond to an inner product.

# The Representer Theorem

## Theorem (Representer Theorem)

Let

$$J(w) = R(\|w\|) + L(\langle w, \psi(x_1) \rangle, \dots, \langle w, \psi(x_n) \rangle),$$

where

- $w, \psi(x_1), \dots, \psi(x_n) \in \mathcal{H}$  for some Hilbert space  $\mathcal{H}$ . (We typically have  $\mathcal{H} = \mathbf{R}^d$ .)
- $\|\cdot\|$  is the norm corresponding to the inner product of  $\mathcal{H}$ . (i.e.  $\|w\| = \sqrt{\langle w, w \rangle}$ )
- $R: \mathbf{R}^{\geq 0} \rightarrow \mathbf{R}$  is nondecreasing (**Regularization term**), and
- $L: \mathbf{R}^n \rightarrow \mathbf{R}$  is arbitrary (**Loss term**).

If  $J(w)$  has a minimizer, then it has a minimizer of the form  $w^* = \sum_{i=1}^n \alpha_i \psi(x_i)$ .  
[If  $R$  is strictly increasing, then all minimizers have this form. (Proof in homework.)]



# The Representer Theorem (Proof)

- 1 Let  $w^*$  be a minimizer.
- 2 Let  $M = \text{span}(\psi(x_1), \dots, \psi(x_n))$ . [the “span of the data”]
- 3 Let  $w = \text{Proj}_M w^*$ . So  $\exists \alpha$  s.t.  $w = \sum_{i=1}^n \alpha_i \psi(x_i)$ .
- 4 Then  $w^\perp := w^* - w$  is orthogonal to  $M$ .
- 5 Projections decrease norms:  $\|w\| \leq \|w^*\|$ .
- 6 Since  $R$  is nondecreasing,  $R(\|w\|) \leq R(\|w^*\|)$ .
- 7 By (4),  $\langle w^*, \psi(x_i) \rangle = \langle w + w^\perp, \psi(x_i) \rangle = \langle w, \psi(x_i) \rangle$ .
- 8  $L(\langle w^*, \psi(x_1) \rangle, \dots, \langle w^*, \psi(x_n) \rangle) = L(\langle w, \psi(x_1) \rangle, \dots, \langle w, \psi(x_n) \rangle)$
- 9  $J(w) \leq J(w^*)$ .
- 10 Therefore  $w = \sum_{i=1}^n \alpha_i \psi(x_i)$  is also a minimizer.

Q.E.D.

## Using Representer Theorem to Kernelize

---

## Kernelized Predictions

- Consider  $w = \sum_{i=1}^n \alpha_i \psi(x_i)$ . (As representer theorem implies.)
- How do we make predictions for a given  $x \in \mathcal{X}$ ?

$$\begin{aligned} f(x) = \langle w, \psi(x) \rangle &= \left\langle \sum_{i=1}^n \alpha_i \psi(x_i), \psi(x) \right\rangle \\ &= \sum_{i=1}^n \alpha_i \langle \psi(x_i), \psi(x) \rangle \\ &= \sum_{i=1}^n \alpha_i k(x_i, x) \end{aligned}$$

**Note:**  $f(x)$  is a linear combination of  $k(x_1, x), \dots, k(x_n, x)$ , all considered as functions of  $x$ .

## Kernelized Regularization

- Consider  $w = \sum_{i=1}^n \alpha_i \psi(x_i)$ .
- What does  $R(\|w\|)$  look like?

$$\begin{aligned}\|w\|^2 &= \langle w, w \rangle \\ &= \left\langle \sum_{i=1}^n \alpha_i \psi(x_i), \sum_{j=1}^n \alpha_j \psi(x_j) \right\rangle \\ &= \sum_{i,j=1}^n \alpha_i \alpha_j \langle \psi(x_i), \psi(x_j) \rangle \\ &= \sum_{i,j=1}^n \alpha_i \alpha_j k(x_i, x_j)\end{aligned}$$

(You should recognize the last expression as a quadratic form.)

# The Kernel Matrix (a.k.a. Gram Matrix)

## Definition

The **kernel matrix** or **Gram matrix** for a kernel  $k$  on a set  $\{x_1, \dots, x_n\}$  is

$$K = (k(x_i, x_j))_{i,j} = \begin{pmatrix} k(x_1, x_1) & \cdots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \cdots & k(x_n, x_n) \end{pmatrix} \in \mathbf{R}^{n \times n}.$$

## Kernelized Regularization: Matrix Form

- Consider  $w = \sum_{i=1}^n \alpha_i \psi(x_i)$ .
- What does  $R(\|w\|)$  look like?

$$\begin{aligned}\|w\|^2 &= \sum_{i,j=1}^n \alpha_i \alpha_j k(x_i, x_j) \\ &= \alpha^T K \alpha\end{aligned}$$

- So  $R(\|w\|) = R\left(\sqrt{\alpha^T K \alpha}\right)$ .

## Kernelized Predictions

- Write  $f_\alpha(x) = \sum_{i=1}^n \alpha_i k(x, x_i)$ . (Switched from  $k(x_i, x)$  by symmetry of inner product.)
- Predictions on the training points have a particularly simple form:

$$\begin{aligned} \begin{pmatrix} f_\alpha(x_1) \\ \vdots \\ f_\alpha(x_n) \end{pmatrix} &= \begin{pmatrix} \alpha_1 k(x_1, x_1) + \cdots + \alpha_n k(x_1, x_n) \\ \vdots \\ \alpha_1 k(x_n, x_1) + \cdots + \alpha_n k(x_n, x_n) \end{pmatrix} \\ &= \begin{pmatrix} k(x_1, x_1) & \cdots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \cdots & k(x_n, x_n) \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix} \\ &= K\alpha \end{aligned}$$

- Substituting

$$w = \sum_{i=1}^n \alpha_i \psi(x_i)$$

into generalized objective, we get

$$\min_{\alpha \in \mathbf{R}^n} R\left(\sqrt{\alpha^T K \alpha}\right) + L(K\alpha).$$

- No direct access to  $\psi(x_i)$ .
- All references are via kernel matrix  $K$ .
- This is the **kernelized objective function**.



- The SVM objective:

$$\min_{w \in \mathcal{H}} \frac{1}{2} \|w\|^2 + \frac{c}{n} \sum_{i=1}^n (1 - y_i [\langle w, \psi(x_i) \rangle])_+.$$

- Kernelizing yields

$$\min_{\alpha \in \mathbf{R}^n} \frac{1}{2} \alpha^T K \alpha + \frac{c}{n} \sum_{i=1}^n (1 - y_i (K \alpha)_i)_+$$

# Kernelized Ridge Regression

- Ridge Regression:

$$\min_{w \in \mathbf{R}^d} \frac{1}{n} \sum_{i=1}^n (w^T x_i - y_i)^2 + \lambda \|w\|^2$$

- Featurized Ridge Regression

$$\min_{w \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n (\langle w, \psi(x_i) \rangle - y_i)^2 + \lambda \|w\|^2$$

- Kernelized Ridge Regression

$$\min_{\alpha \in \mathbf{R}^n} \frac{1}{n} \|K\alpha - y\|^2 + \lambda \alpha^T K \alpha,$$

where  $y = (y_1, \dots, y_n)^T$ .

## Prediction Functions with RBF Kernel

# Radial Basis Function (RBF) / Gaussian Kernel

- Input space  $\mathcal{X} = \mathbf{R}^d$

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

where  $\sigma^2$  is known as the bandwidth parameter.

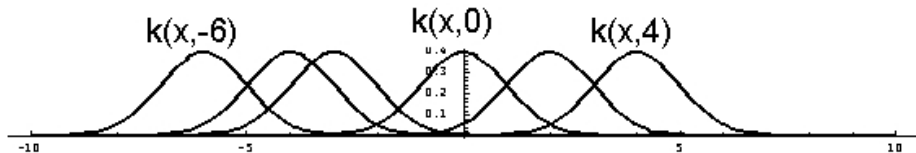
- Does it act like a similarity score?
- Why “radial”?
- Have we departed from our “inner product of feature vector” recipe?
  - Yes and no: corresponds to an infinite dimensional feature vector
- Probably the most common nonlinear kernel.

# RBF Basis

- Input space  $\mathcal{X} = \mathbb{R}$
- Output space:  $\mathcal{Y} = \mathbb{R}$
- RBF kernel  $k(w, x) = \exp\left(- (w - x)^2\right)$ .
- Suppose we have 6 training examples:  $x_i \in \{-6, -4, -3, 0, 2, 4\}$ .
- If representer theorem applies, then

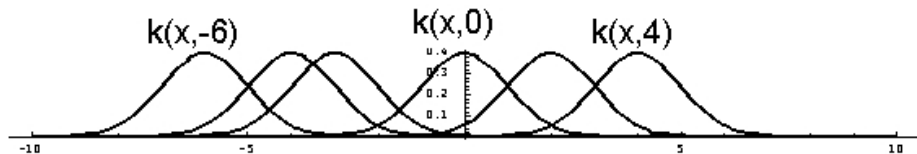
$$f(x) = \sum_{i=1}^6 \alpha_i k(x_i, x).$$

- $f$  is a linear combination of 6 basis functions of form  $k(x_i, \cdot)$ :

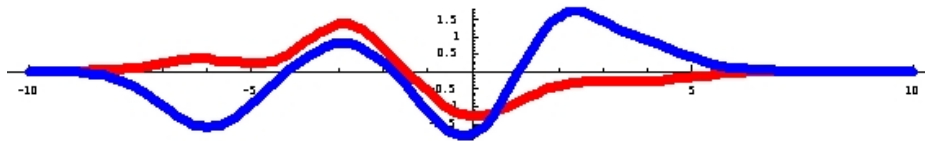


# RBF Predictions

- Basis functions



- Predictions of the form  $f(x) = \sum_{i=1}^6 \alpha_i k(x_i, x)$ :



- When kernelizing with RBF kernel, prediction functions always look this way.
- (Whether we get  $w$  from SVM, ridge regression, etc...)

## RBF Feature Space: The Sequence Space $\ell_2$

- To work with infinite dimensional feature vectors, we need a space with certain properties.
  - an inner product
  - a norm related to the inner product
  - projection theorem:  $x = x_{\perp} + x_{\parallel}$  where  $x_{\parallel} \in S = \text{span}(w_1, \dots, w_n)$  and  $\langle x_{\perp}, s \rangle = 0 \quad \forall s \in S$ .
- Basically, we need a Hilbert space.

### Definition

$\ell_2$  is the space of all real-valued sequences:  $(x_0, x_1, x_2, x_3, \dots)$  with  $\sum_{i=0}^{\infty} x_i^2 < \infty$ .

### Theorem

*With the inner product  $\langle x, x' \rangle = \sum_{i=0}^{\infty} x_i x'_i$ ,  $\ell_2$  is a **Hilbert space**.*

# The Infinite Dimensional Feature Vector for RBF

- Consider RBF kernel (1-dim):  $k(w, x) = \exp\left(-(w - x)^2 / 2\right)$
- We claim that  $\psi : \mathbf{R} \rightarrow \ell_2$  defined by

$$[\psi(x)]_n = \frac{1}{\sqrt{n!}} e^{-x^2/2} x^n$$

gives the “infinite-dimensional feature vector” corresponding to RBF kernel.

- Is this mapping even well-defined? Is  $\psi(x)$  even an element of  $\ell_2$ ?
- Yes:

$$\sum_{n=0}^{\infty} \frac{1}{n!} e^{-x^2} x^{2n} = e^{-x^2} \sum_{n=0}^{\infty} \frac{(x^2)^n}{n!} = 1 < \infty$$



# The Infinite Dimensional Feature Vector for RBF

- Does feature vector  $[\psi(x)]_n = \frac{1}{\sqrt{n!}} e^{-x^2/2} x^n$  actually correspond to the RBF kernel?
- Yes! Proof:

$$\begin{aligned}\langle \psi(w), \psi(x) \rangle &= \sum_{n=0}^{\infty} \frac{1}{n!} e^{-(x^2+w^2)/2} x^n w^n \\ &= e^{-(x^2+w^2)/2} \sum_{n=0}^{\infty} \frac{(xw)^n}{n!} \\ &= \exp\left(-[x^2 + w^2]/2\right) \exp(xw) \\ &= \exp\left(-[(x-w)^2/2]\right)\end{aligned}$$

QED

When is  $k(x, w)$  a kernel function? (Mercer's Theorem)

# How to Get Kernels?

- 1 Explicitly construct  $\psi(x) : \mathcal{X} \rightarrow \mathbf{R}^d$  and define  $k(x, w) = \psi(x)^T \psi(w)$ .
- 2 Directly define the kernel function  $k(x, w)$ , and verify it corresponds to  $\langle \psi(x), \psi(w) \rangle$  for some  $\psi$ .

There are many theorems to help us with the second approach

# Positive Semidefinite Matrices

## Definition

A real, symmetric matrix  $M \in \mathbf{R}^{n \times n}$  is **positive semidefinite (psd)** if for any  $x \in \mathbf{R}^n$ ,

$$x^T M x \geq 0.$$

## Theorem

*The following conditions are each necessary and sufficient for  $M$  to be positive semidefinite:*

- *$M$  has a “square root”, i.e. there exists  $R$  s.t.  $M = R^T R$ .*
- *All eigenvalues of  $M$  are greater than or equal to 0.*

# Positive Semidefinite Function

## Definition

A symmetric kernel function  $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbf{R}$  is **positive semidefinite (psd)** if for any finite set  $\{x_1, \dots, x_n\} \in \mathcal{X}$ , the kernel matrix on this set

$$K = (k(x_i, x_j))_{i,j} = \begin{pmatrix} k(x_1, x_1) & \cdots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \cdots & k(x_n, x_n) \end{pmatrix}$$

is a positive semidefinite matrix.

# Mercer's Theorem

## Theorem

*A symmetric function  $k(w, x)$  can be expressed as an inner product*

$$k(w, x) = \langle \psi(w), \psi(x) \rangle$$

*for some  $\psi$  if and only if  $k(w, x)$  is **positive semidefinite**.*

# Generating New Kernels from Old

- Suppose  $k, k_1, k_2 : \mathcal{X} \times \mathcal{X} \rightarrow \mathbf{R}$  are psd kernels. Then so are the following:

$$k_{\text{new}}(w, x) = k_1(w, x) + k_2(w, x)$$

$$k_{\text{new}}(w, x) = \alpha k(w, x)$$

$$k_{\text{new}}(w, x) = f(w)f(x) \text{ for any function } f(x)$$

$$k_{\text{new}}(w, x) = k_1(w, x)k_2(w, x)$$

- See Appendix for details.
- Lots more theorems to help you construct new kernels from old...

## Details on New Kernels from Old



- Suppose  $k_1$  and  $k_2$  are psd kernels with feature maps  $\phi_1$  and  $\phi_2$ , respectively.
- Then

$$k_1(w, x) + k_2(w, x)$$

is a psd kernel.

- Proof: Concatenate the feature vectors to get

$$\phi(x) = (\phi_1(x), \phi_2(x)).$$

Then  $\phi$  is a feature map for  $k_1 + k_2$ .

## Closure under Positive Scaling

- Suppose  $k$  is a psd kernel with feature maps  $\phi$ .
- Then for any  $\alpha > 0$ ,

$$\alpha k$$

is a psd kernel.

- Proof: Note that

$$\phi(x) = \sqrt{\alpha}\phi(x)$$

is a feature map for  $\alpha k$ .

## Scalar Function Gives a Kernel

- For any function  $f(x)$ ,

$$k(w, x) = f(w)f(x)$$

is a kernel.

- Proof: Let  $f(x)$  be the feature mapping. (It maps into a 1-dimensional feature space.)

$$\langle f(x), f(w) \rangle = f(x)f(w) = k(w, x).$$

## Closure under Hadamard Products

- Suppose  $k_1$  and  $k_2$  are psd kernels with feature maps  $\phi_1$  and  $\phi_2$ , respectively.
- Then

$$k_1(w, x) k_2(w, x)$$

is a psd kernel.

- Proof: Take the outer product of the feature vectors:

$$\phi(x) = \phi_1(x) [\phi_2(x)]^T.$$

Note that  $\phi(x)$  is a matrix.

- Continued...

# Closure under Hadamard Products

- Then

$$\begin{aligned}\langle \phi(x), \phi(w) \rangle &= \sum_{i,j} \phi(x) \phi(w) \\ &= \sum_{i,j} \left[ \phi_1(x) [\phi_2(x)]^T \right]_{ij} \left[ \phi_1(w) [\phi_2(w)]^T \right]_{ij} \\ &= \sum_{i,j} [\phi_1(x)]_i [\phi_2(x)]_j [\phi_1(w)]_i [\phi_2(w)]_j \\ &= \left( \sum_i [\phi_1(x)]_i [\phi_1(w)]_i \right) \left( \sum_j [\phi_2(x)]_j [\phi_2(w)]_j \right) \\ &= k_1(w, x) k_2(w, x)\end{aligned}$$