Kernel methods IV
The kernel function

Sanjoy Dasgupta

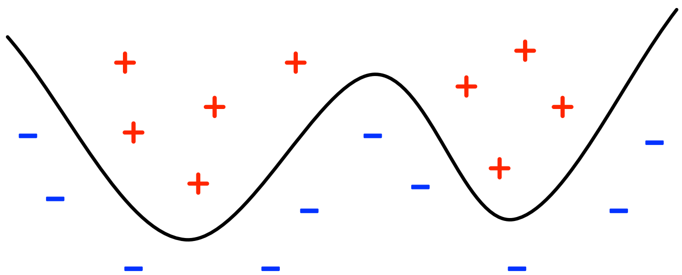
University of California, San Diego

Topics we'll cover

1. The kernel function
2. The RBF kernel

Basis expansion

Suppose we want a decision boundary that is a polynomial of order p:



Add new features to data vectors x:

- Let $\Phi(x)$ consist of all terms of order $\leq p$, such as $x_1 x_2^2 x_3^{p-3}$.
- Degree-p polynomial in $x \Leftrightarrow$ linear in $\Phi(x)$.
- $\Phi(x) \cdot \Phi(z) = (1 + x \cdot z)^p$.

Kernel SVM

1. **Basis expansion.** Mapping $x \mapsto \Phi(x)$.
2. **Learning.** Solve the dual problem:

$$\boxed{\begin{array}{c} \max_{\alpha \in \mathbb{R}^n} \quad \sum_{i=1}^n \alpha_i - \sum_{i,j=1}^n \alpha_i \alpha_j y^{(i)} y^{(j)} (\Phi(x^{(i)}) \cdot \Phi(x^{(j)})) \\ \text{s.t.:} \quad \sum_{i=1}^n \alpha_i y^{(i)} = 0 \\ 0 \leq \alpha_i \leq C \end{array}}$$

   This yields $\alpha = (\alpha_1, \ldots, \alpha_n)$. Offset b also follows.

3. **Classification.** Given a new point x, classify as

$$\text{sign}\left(\sum_i \alpha_i y^{(i)} (\Phi(x^{(i)}) \cdot \Phi(x)) + b \right).$$

Kernel SVM, revisited

1. **Kernel function.** Define a similarity function $k(x, z)$.
2. **Learning.** Solve the dual problem:

$$\boxed{\begin{aligned} \max_{\alpha \in \mathbb{R}^n} \quad & \sum_{i=1}^{n} \alpha_i - \sum_{i,j=1}^{n} \alpha_i \alpha_j y^{(i)} y^{(j)} k(x^{(i)}, x^{(j)}) \\ \text{s.t.:} \quad & \sum_{i=1}^{n} \alpha_i y^{(i)} = 0 \\ & 0 \leq \alpha_i \leq C \end{aligned}}$$

   This yields α. Offset b also follows.
3. **Classification.** Given a new point x, classify as

$$\operatorname{sign}\left(\sum_i \alpha_i y^{(i)} k(x^{(i)}, x) + b \right).$$

The kernel function

We never explicitly construct the embedding $\Phi(x)$.

- What we actually use is the **kernel function** $k(x, z) = \Phi(x) \cdot \Phi(z)$.
- Can think of $k(x, z)$ as a **measure of similarity** between x and z.
- Rewrite learning algorithm and final classifier in terms of k.

Kernel Perceptron:

- $\alpha = 0$ and $b = 0$
- while some i has $y^{(i)} \left(\sum_j \alpha_j y^{(j)} k(x^{(j)}, x^{(i)}) + b \right) \leq 0$:
  - $\alpha_i = \alpha_i + 1$
  - $b = b + y^{(i)}$

To classify a new point x: sign $\left(\sum_j \alpha_j y^{(j)} k(x^{(j)}, x) + b \right)$.

Choosing the kernel function

The final classifier is a **similarity-weighted vote**,

$$F(x) = \alpha_1 y^{(1)} k(x^{(1)}, x) + \cdots + \alpha_n y^{(n)} k(x^{(n)}, x)$$

(plus an offset term, b).

Can we choose k to be **any** similarity function?

- Not quite: need $k(x, z) = \Phi(x) \cdot \Phi(z)$ for *some* embedding Φ.
- **Mercer's condition**: same as requiring that for any finite set of points $x^{(1)}, \ldots, x^{(m)}$, the $m \times m$ similarity matrix K given by

  $$K_{ij} = k(x^{(i)}, x^{(j)})$$

  is positive semidefinite.

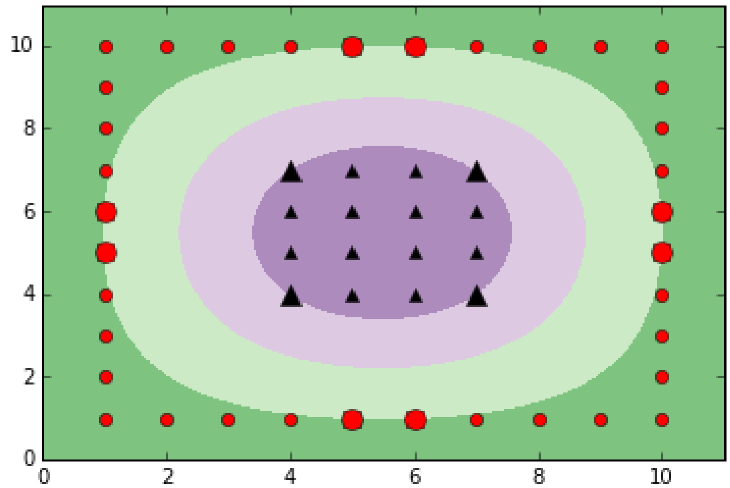
The RBF kernel

A popular similarity function: the **Gaussian kernel** or **RBF kernel**

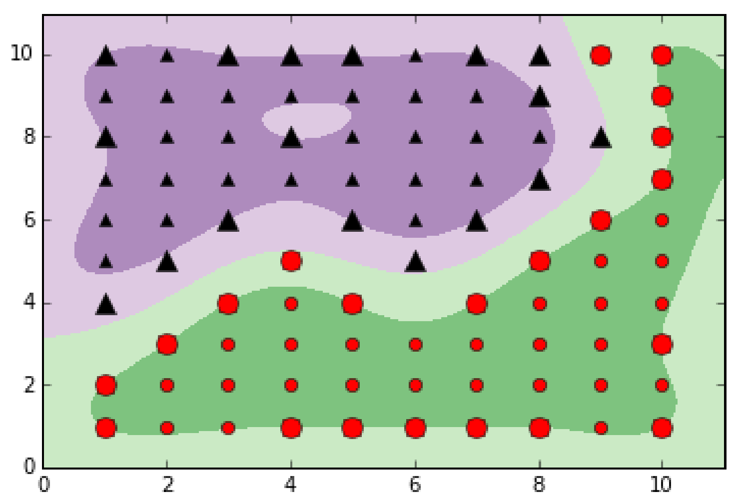
$$k(x, z) = e^{-\|x-z\|^2/s^2},$$

where s is an adjustable scale parameter.

RBF kernel: examples



RBF kernel: examples



The scale parameter

Recall prediction function: $F(x) = \alpha_1 y^{(1)} k(x^{(1)}, x) + \cdots + \alpha_n y^{(n)} k(x^{(n)}, x)$.

For the RBF kernel, $k(x, z) = e^{-\|x-z\|^2/s^2}$,

1. How does this function behave as $s \uparrow \infty$?
2. How does this function behave as $s \downarrow 0$?
3. As we get more data, should we increase or decrease s?