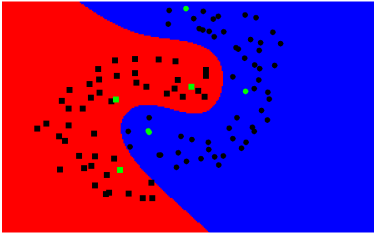
Nonlinear SVM

COMP4211

THE DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
計算機科學及工程學系

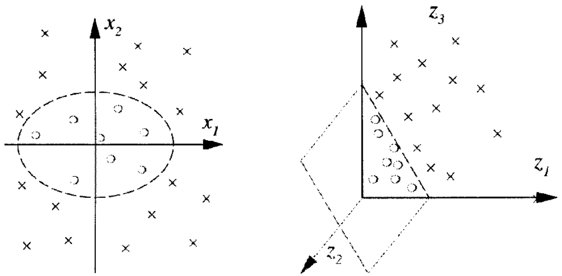
Nonlinear Decision Surface



Naive method:

- instead of a line, use a curve
- not efficient

Preprocess the data with $\varphi : \mathbb{R}^m \rightarrow \mathcal{H}$, $\mathbf{x} \mapsto \varphi(\mathbf{x})$



- $\mathbb{R}^m$: input space
- $\mathcal{H}$: feature space

Example: All Degree 2 Monomials

- $\varphi(\mathbf{x}) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$
- $(\varphi(\mathbf{x}_1), y_1), \ldots, (\varphi(\mathbf{x}_N), y_N) \in \mathcal{H} \times \{\pm 1\}$
- $\varphi(\mathbf{x}) \mapsto y$
- $f = \mathrm{sign}(\mathbf{w}'\varphi(\mathbf{x}) + b)$

Problem

for $m = 256, d = 5 \rightarrow$ dimensionality 10^{10}

Shortcut

Recall that the training data only appear (in both training and testing) in the form of dot products between vectors

- training

$$\max \quad \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i' \mathbf{x}_j$$

$$\text{s.t.} \quad \alpha_i \geq 0, \sum_{i=1}^{N} \alpha_i y_i = 0$$

- testing

$$\text{sign}\left(\sum_{i=1}^{N_S} \alpha_i y_i \mathbf{x}_i' \mathbf{x} + b \right)$$

Shortcut...

Example (degree 2 case)

- $d = 2$
- $\mathbf{x} = (x_1, x_2) \mapsto \varphi(\mathbf{x}) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$

$\mathbf{a}, \mathbf{b} \in \mathbb{R}^2$

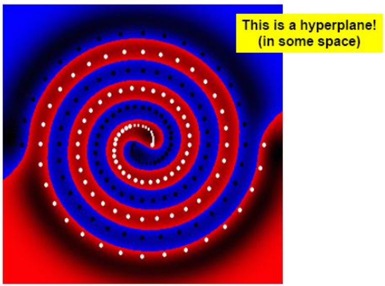
$$\begin{aligned}
\varphi(\mathbf{a})'\varphi(\mathbf{b}) &= (a_1^2, \sqrt{2}a_1a_2, a_2^2)'(b_1^2, \sqrt{2}b_1b_2, b_2^2) \\
&= a_1^2b_1^2 + 2a_1a_2b_1b_2 + a_2^2b_2^2 \\
&= (\mathbf{a}'\mathbf{b})^2
\end{aligned}$$

dot product can be computed in $\mathbb{R}^2$ (without going to $\mathcal{H}$)

Kernels

The dot product in $\mathcal{H}$ can be computed in $\mathbb{R}^m$

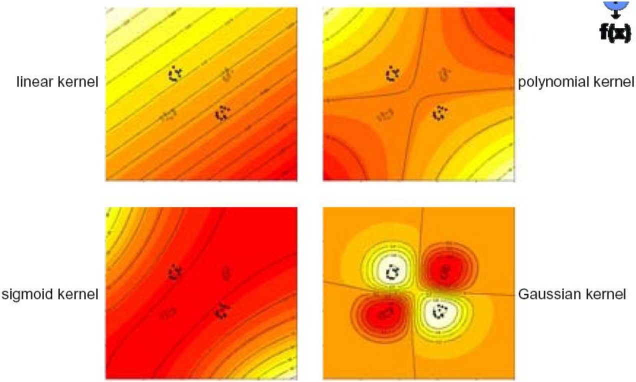
- define $k(\mathbf{x}, \mathbf{y}) \equiv (\mathbf{x}'\mathbf{y})^d$, k: kernel function
- kernels are functions that return inner products between the images of data points in some space
- by replacing inner products with kernels in linear algorithms, we can obtain very flexible representations



- choosing k is equivalent to choosing the feature map

Examples of Kernels

- inhomogeneous polynomial: $k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}'\mathbf{y} + 1)^d$
- Gaussian: $k(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|^2/(2\sigma^2))$
  - radial basis function (RBF) network
  - corresponds to an infinite-dimensional feature space
- sigmoid: $k(\mathbf{x}, \mathbf{y}) = \tanh(\kappa(\mathbf{x}'\mathbf{y}) + \theta)$
  - a valid kernel only for certain κ and θ

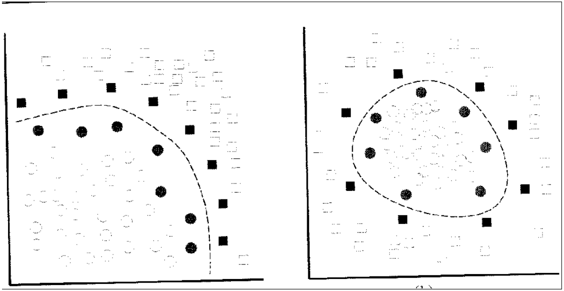


“Kernel Trick”

Kernel: substitutes the dot product and act as a nonlinear similarity measure

> Any algorithm that depends only on dot products can use the kernel trick!

Example



usually support vectors are very few in number

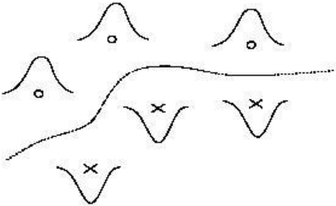
- data compression

Gaussian Kernel

Recall that the decision rule uses

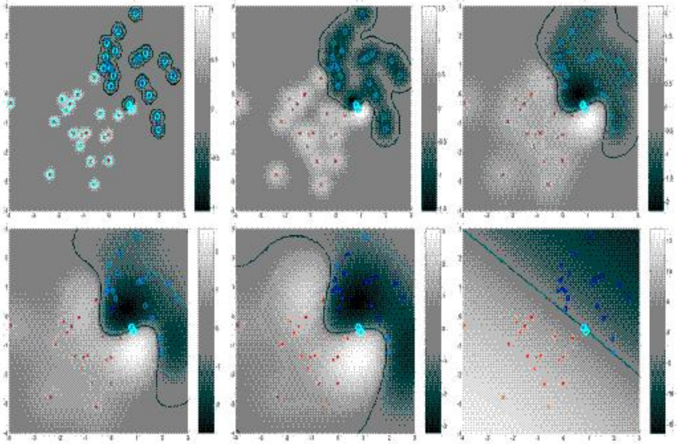
$$\mathbf{w}'\mathbf{x} + b = \sum_{i=1}^{N} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + b$$

- Gaussian kernel: $k(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\|\mathbf{x}_1 - \mathbf{x}_2\|^2 / 2\sigma^2)$
- amounts to putting bumps of various sizes on the training set



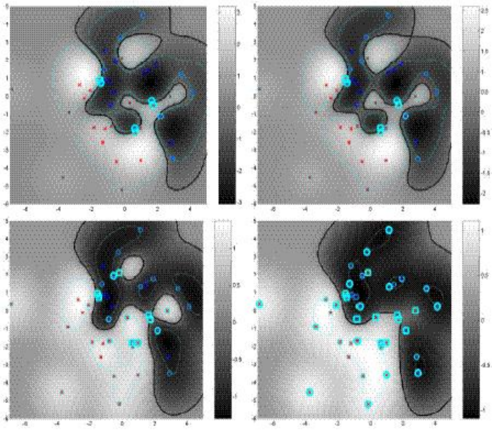
Example: Two overlapping Gaussians belonging to two classes with means $(-1, -1)$ and $(1, 1)$ and standard deviation 1

- $\sigma = 0.1, 0.25, 0.5, 0.75, 1.10$



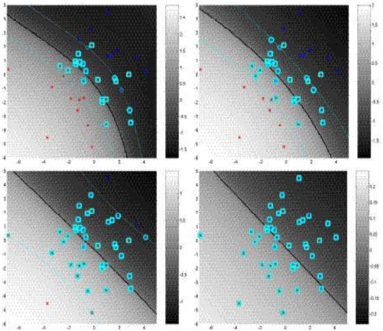
Varying C

$(\sigma^* = 2, \sigma = 1)$ $C = \text{inf}, 100, 10, 1$



Varying C (More)

$(\sigma^* = 2, \sigma = 10)$ $C = 100, 10, 1$ and $\sigma = 100$, $C = 1$



Heuristic: $\sigma^2 = \frac{1}{N(N-1)} \sum_{i,j=1}^{N} \|\mathbf{x}_i - \mathbf{x}_j\|^2$