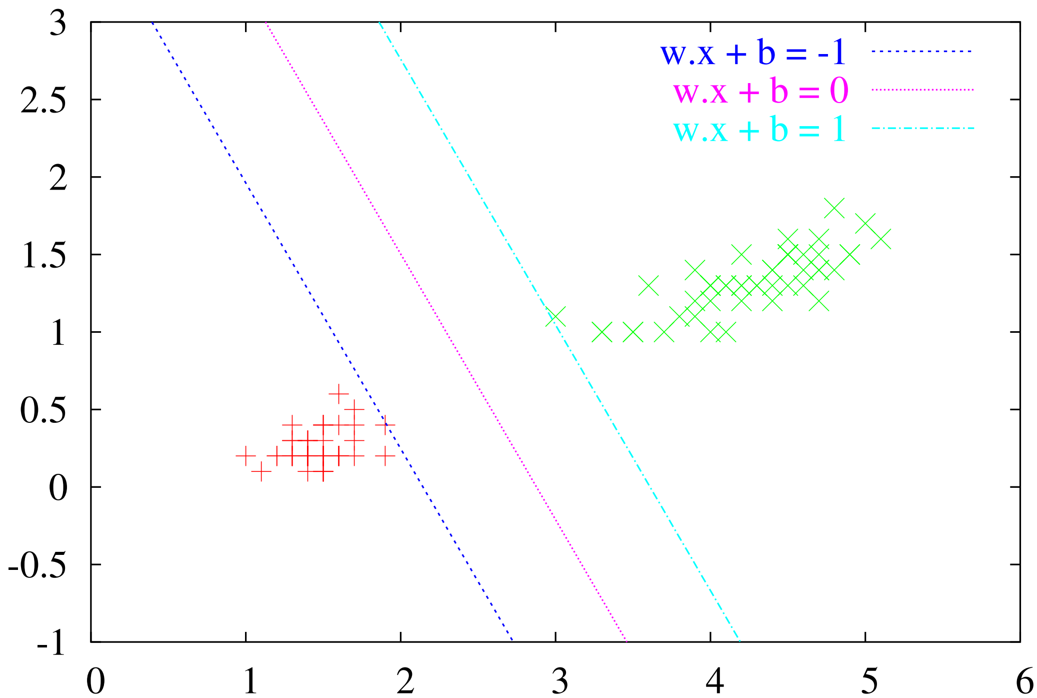
Learning Problem

Given training data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$, determine how to predict y for new inputs $\mathbf{x}$.

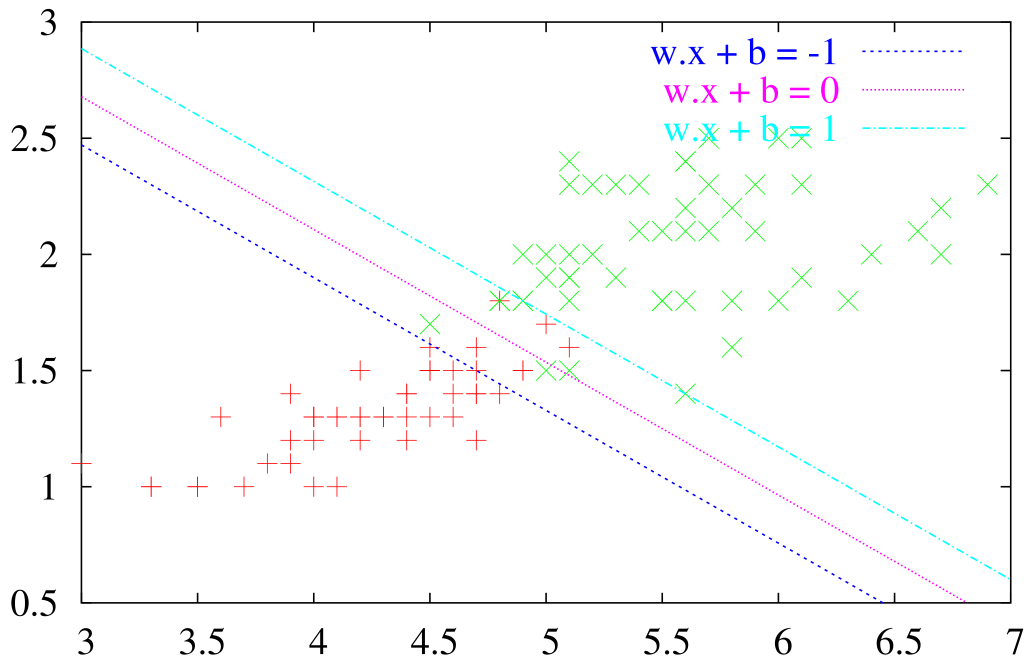
For binary classification, $y_i \in \{-1, 1\}$. For regression, $y \in \mathbf{R}$.

We assume that the prediction y for $\mathbf{x}$ should be based on $(\mathbf{x}_i, y_i)$ in the training data where $\mathbf{x}_i$ is "similar" to $\mathbf{x}$.

Example SVM for Separable Examples



Example SVM for Nonseparable Examples



Support Vector Machines

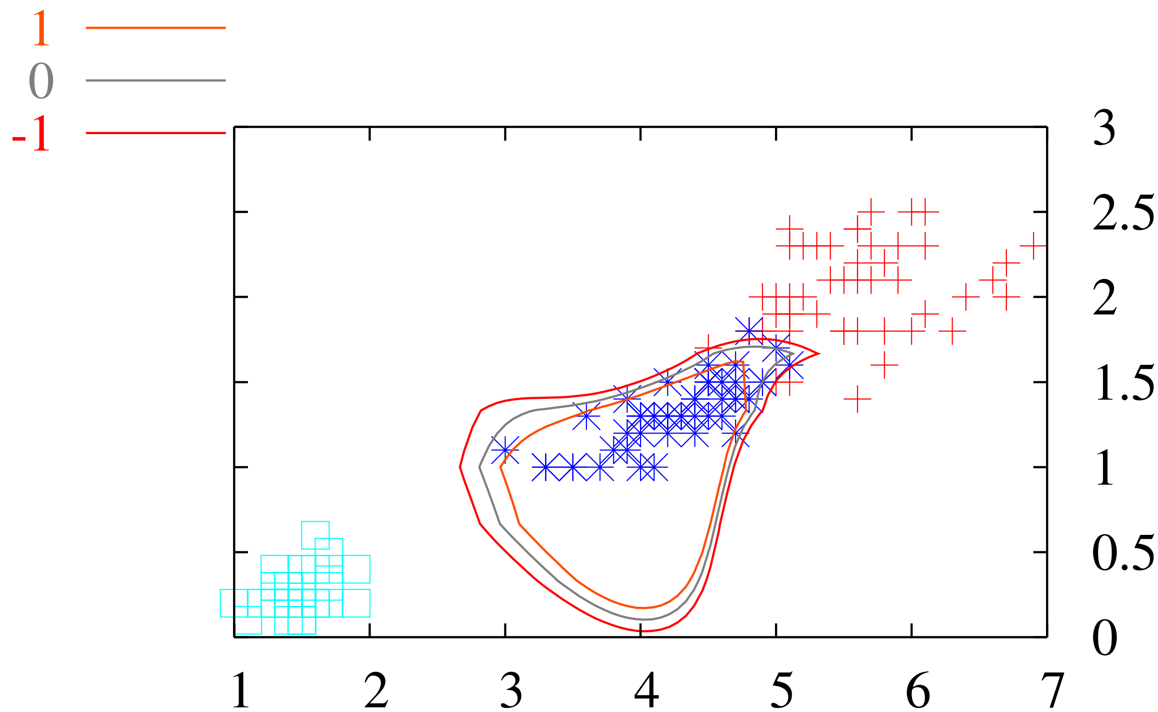
If possible, a SVM separates positive from negative exs. with a hyperplane (similar to perceptrons, adalines).

A SVM seeks to keep the weights as small as possible with positive exs. ≥ 1 and negative exs. ≤ -1.

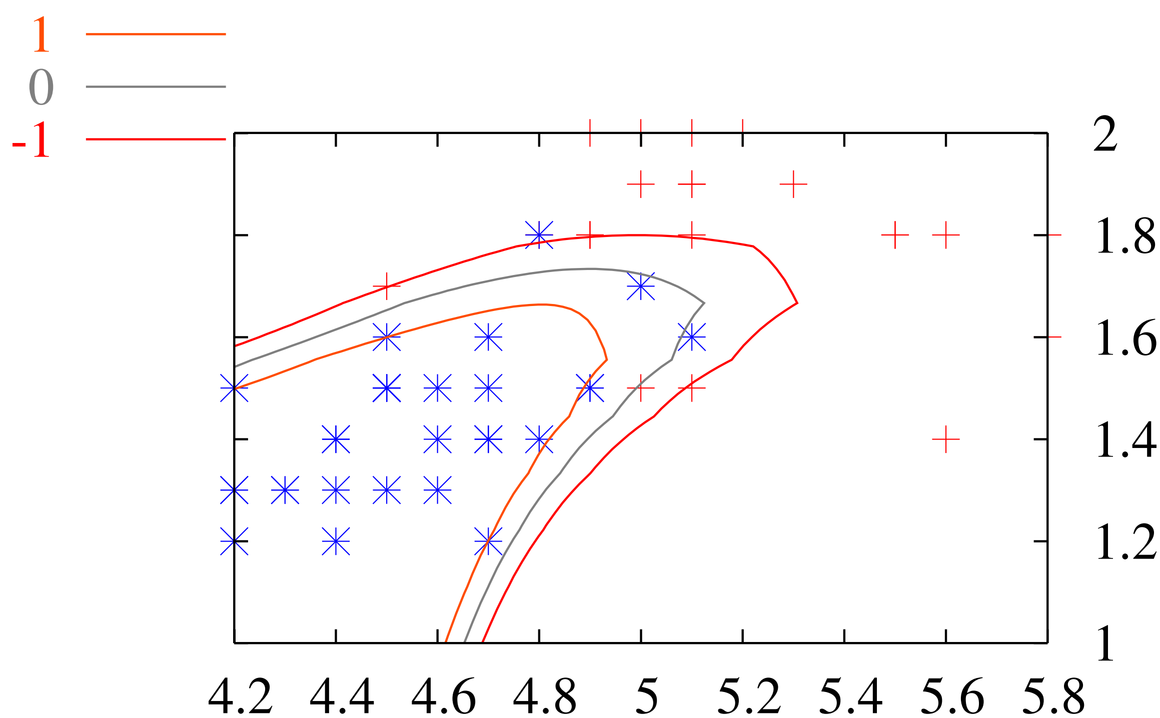
A SVM is defined by its support vectors, the positive exs. ≤ 1 and negative exs. ≥ -1.

Kernel fns. allow nonlinear decision boundaries.

Example Gaussian Kernel SVM



Example Gaussian Kernel, Zoomed In



Kernel Functions

Let $(\mathbf{a} \cdot \mathbf{b})$ represent the dot product of vectors $\mathbf{a}$ and $\mathbf{b}$, where $(\mathbf{a} \cdot \mathbf{b}) = \sum_j a_j b_j$.

Let $\mathbf{\Phi}$ be a set of basis functions $\Phi_1, \Phi_2, \ldots$. A *kernel function* $k(\mathbf{x}, \mathbf{x}') = (\mathbf{\Phi}(\mathbf{x}) \cdot \mathbf{\Phi}(\mathbf{x}'))$ is a way of measuring the similarity of two input vectors.

The "trick" of certain kernel functions is a simple computation that corresponds to a large number of basis functions.

Example Kernel Functions

Dot product: $(\mathbf{x} \cdot \mathbf{x}')$

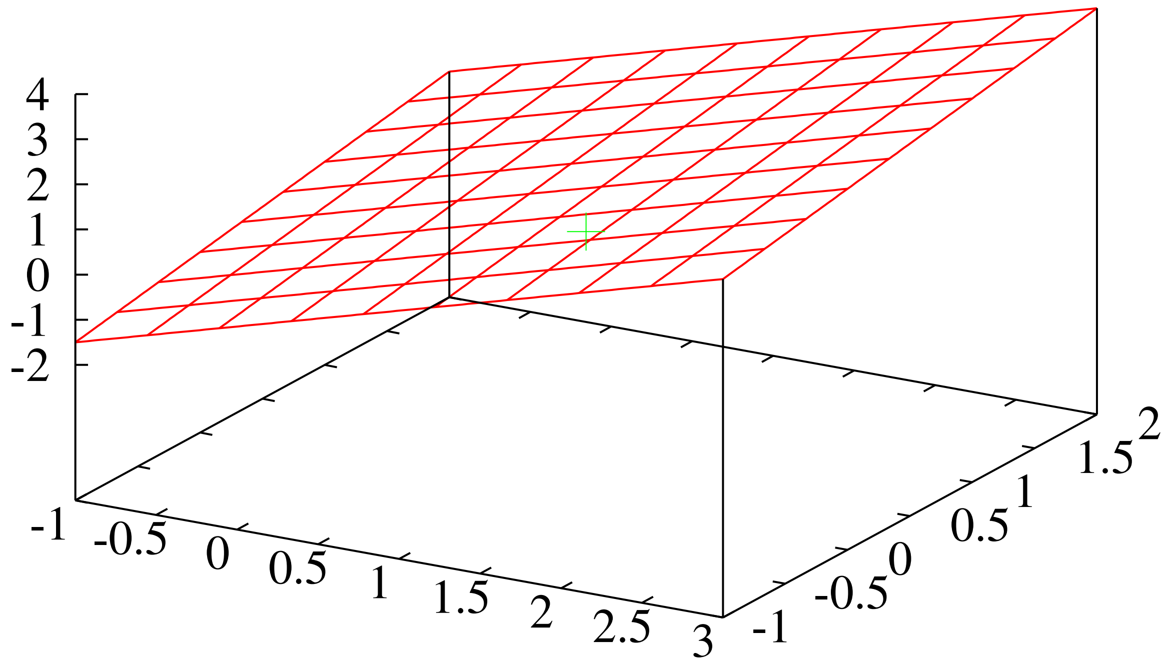
Polynomial: $(\mathbf{x} \cdot \mathbf{x}')^d$, where $d \in \{2, 3, \ldots\}$
$(a(\mathbf{x} \cdot \mathbf{x}') + c)^d$, where $d \in \{2, 3, \ldots\}$, $a, c > 0$

Gaussian: $e^{-\|\mathbf{x}-\mathbf{x}'\|^2/(2\sigma^2)}$, where $\sigma > 0$.

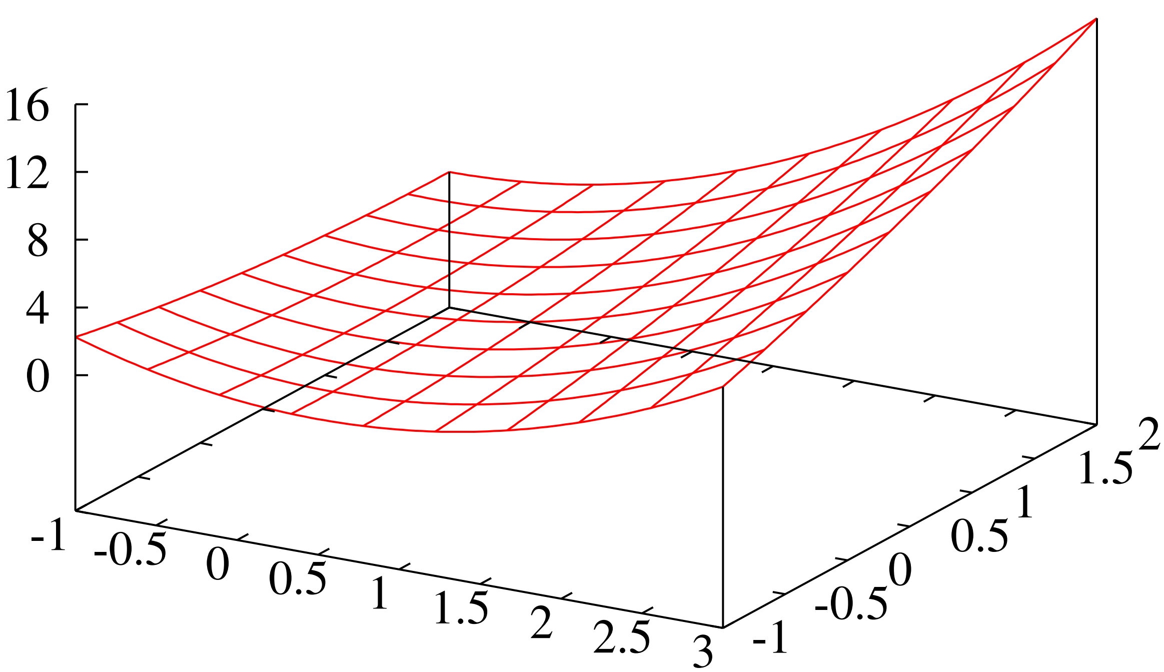
Hypertangent: $\tanh(\kappa(\mathbf{x} \cdot \mathbf{x}') + \theta)$, where $\kappa > 0$, $\theta \in \mathbf{R}$

The following figures illustrates these functions with one point fixed at $(1, 0.5)$.

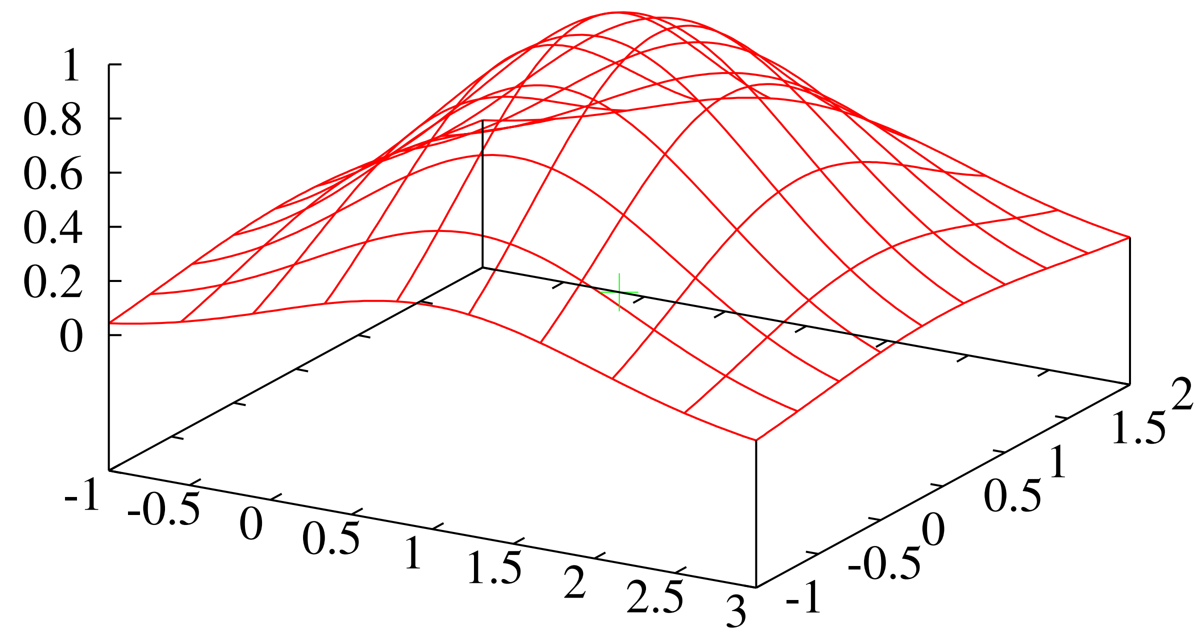
Dot Product



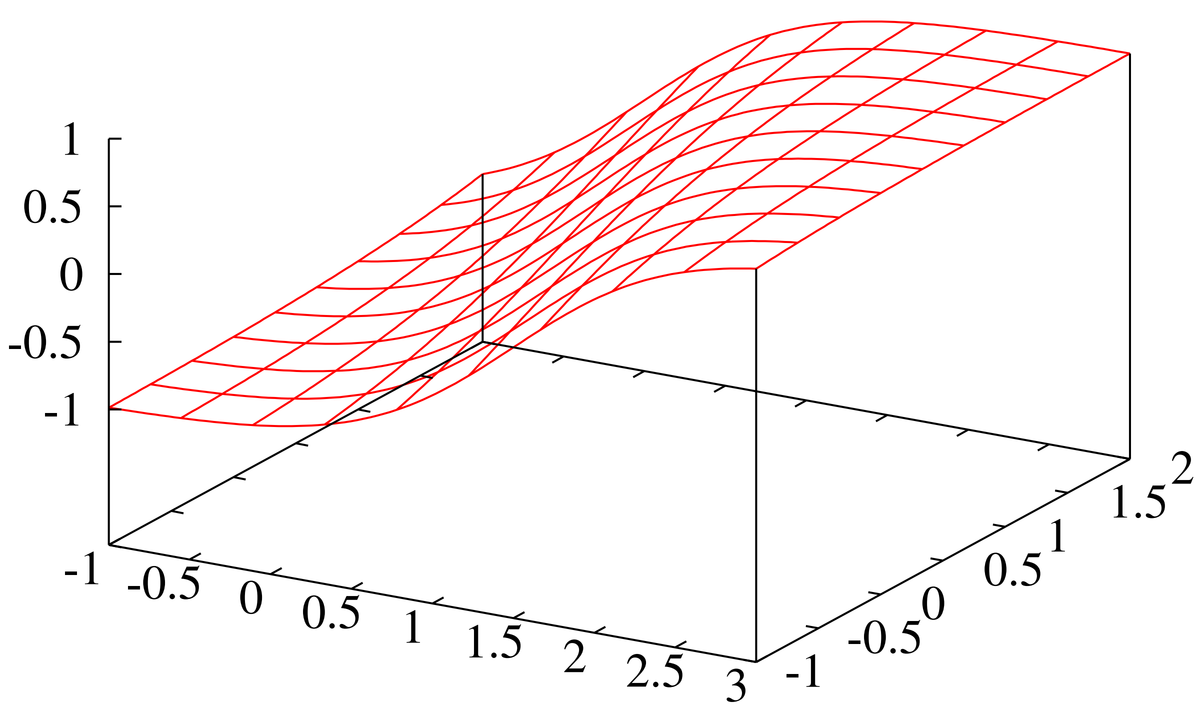
Polynomial, Degree 2



Gaussian



Hypertangent



Basis Functions

Except for tanh, these functions have the form $k(\mathbf{x}, \mathbf{x'}) = (\boldsymbol{\Phi}(\mathbf{x}) \cdot \boldsymbol{\Phi}(\mathbf{x'}))$. For example:

$$\begin{aligned}
&((u_1, u_2) \cdot (v_1, v_2))^2 \\
&= (u_1 v_1 + u_2 v_2)^2 \\
&= (u_1^2 v_1^2 + 2 u_1 u_2 v_1 v_2 + u_2^2 v_2^2) \\
&= (u_1^2, \sqrt{2} u_1 u_2, u_2^2) \cdot (v_1^2, \sqrt{2} v_1 v_2, v_2^2)
\end{aligned}$$

Roughly, more basis functions implies richer representation, but more opportunities for overfitting. The Gaussian kernel corresponds to an infinite number of basis functions!
