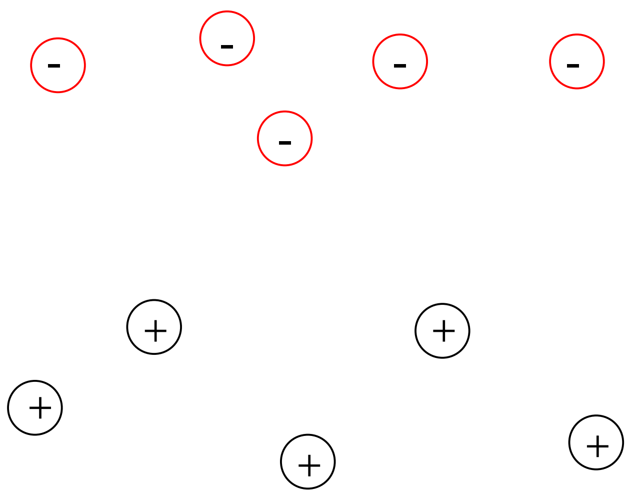
⋄ Binary Classification ⋄

Given some training data

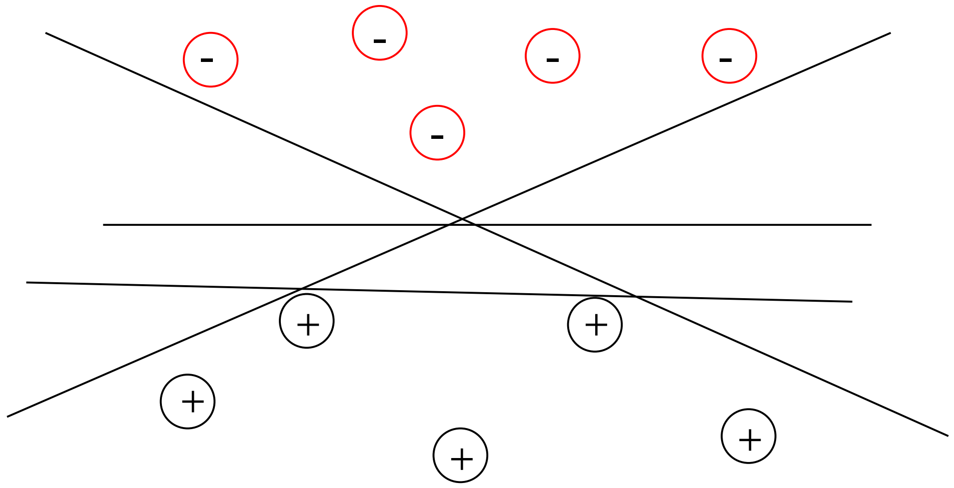
$$(x_1, y_1), \ldots, (x_l, y_l), \qquad y_i \in \{-1, 1\}$$

find the function $f(x, \alpha_0) \in f(x, \alpha)$, which best approximates the unknown function $y = f(x)$.



⋄ Separating Hyperplane ⋄

If the data is linearly separable one can separate it by an infinite number of linear hyperplanes.

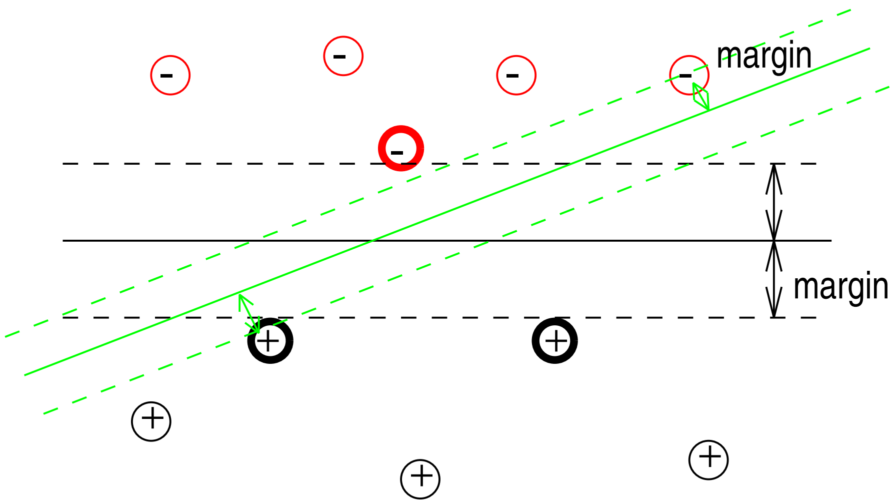


We can write these hyperplanes as

$$f(x, \alpha) = (w_\alpha \cdot x) + b$$

◇ Optimal Separating Hyperplane ◇

Among these hyperplanes there is one with the maximum margin: Optimal Separating Hyperplane



This hyperplane is uniquely determined by the vectors on the margin, the support vectors.

[Vapnik, Chervonenkis '74]

⋄ Optimal Separating Hyperplane ⋄

The hyperplane separating the data

$$(x_1, y_1), \ldots, (x_l, y_l)$$

is one that satifies the conditions

$$(w \cdot x_i) + b \geq 1, \quad \text{if } y_i = 1$$

$$(w \cdot x_i) + b \leq -1, \quad \text{if } y_i = -1$$

or in short

$$y_i[(w \cdot x_i) + b] \geq 1$$

The optimal hyperplane satisfies the above conditions and has the minimal norm

$$||w||^2 = (w \cdot w)$$

[Vapnik, Chervonenkis '74]

⋄ Soft Margin Hyperplane ⋄

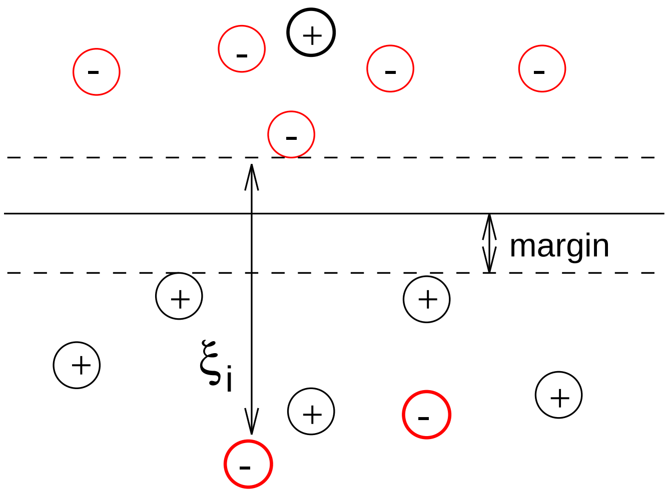
When the data is not linearly separable, we minimize

$$(w \cdot w) + C \sum_i \xi_i^\delta, \qquad \delta \geq 0$$

under constraints

$$y_i[(w \cdot x_i) + b] \geq 1 - \xi_i, \qquad \xi_i \geq 0$$

where the ξ_i allow for some error.



To minimize the number of errors δ should be close to 0.

This is not a convex optimization problem and is hard. The smallest δ to make this simple is $\delta = 1$. [Cortes, Vapnik '95]

◇ The Lagrangian ◇

To construct the optimal separating hyperplane in the separable and non-separable case find the saddle point of the lagrangian

$$L(w, b, \alpha, \xi) =$$

$$\frac{1}{2}(w\cdot w)+C\sum_i \xi_i-\sum_i \alpha_i(\xi_i+y_i[(w\cdot x_i)+b]-1)$$

(minimizing with respect to w, b, ξ and maximizing with respect to α)

under constraint

$$\xi_i \geq 0, \alpha_i \geq 0$$

$\diamond$ Solution $\diamond$

The saddle point is defined as follows:

$$w = \sum_i \alpha_i y_i x_i$$

where α is the maximum point of

$$W(\alpha) = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \quad (1)$$

subject to constraints

$$\sum_i \alpha_i y_i = 0$$

and

$$0 \leq \alpha_i \leq C$$

The separating hyperplane has the form

$$f(x) = \sum_i \alpha_i y_i (x_i \cdot x) + b \qquad (2)$$

IMPORTANT:

(1) and (2) do not explicitly depend on the dimensionality.

◇ Properties of the ◇

◇ Optimal Separating ◇

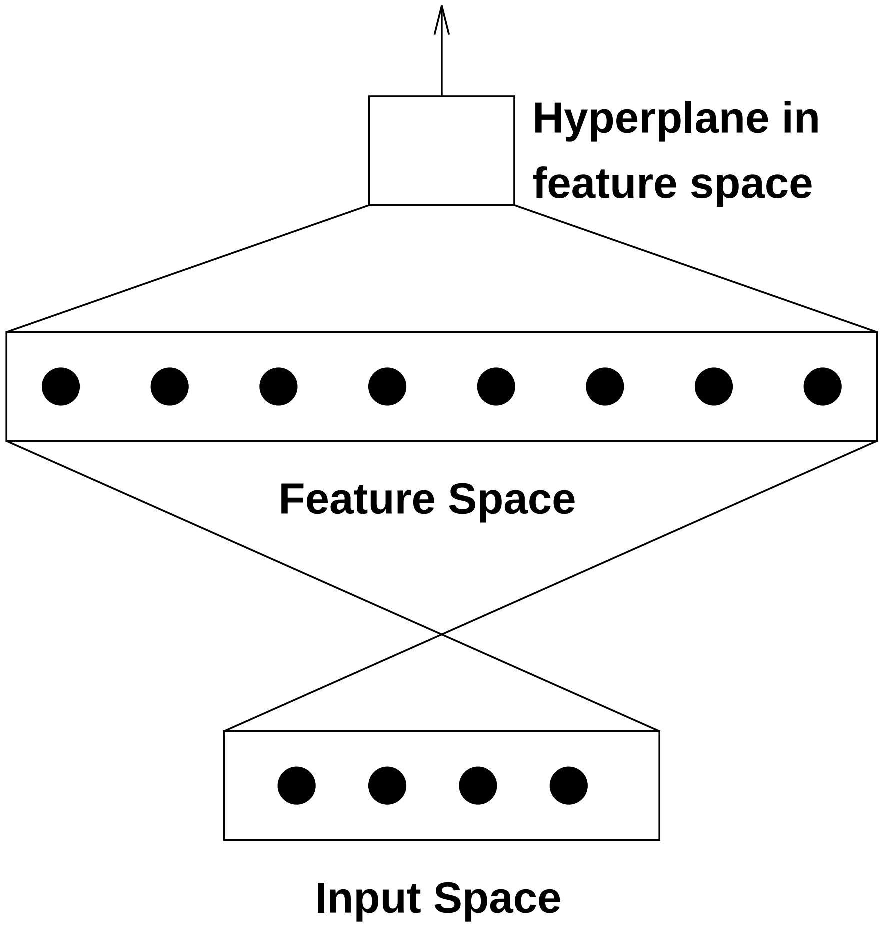
◇ Hyperplane ◇

1. The optimal separating hyperplane is defined by the so-called support vectors. For these $\alpha_i \neq 0$.
2. Construction of the optimal separating hyperplane does not depend explicitly on the dimensionality of the problem.
3. The description of the optimal separating hyperplane does not explicitly depend on the dimensionality of the problem.

◇ Feature Spaces ◇

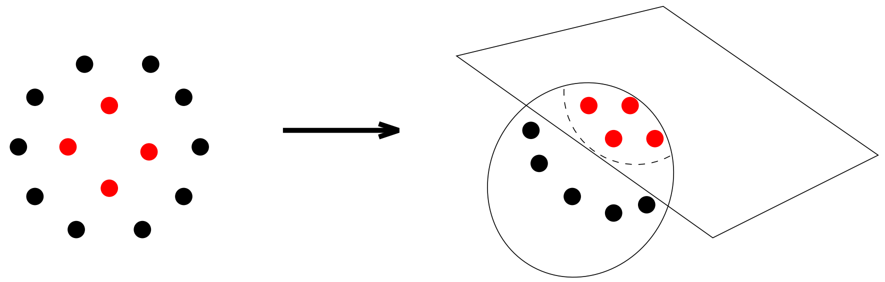
Idea of the SV machine:

- ◇ Map input vectors non-linearly into a high dimensional feature space.
- ◇ Construct the optimal separating hyperplane in the high dimensional feature space.

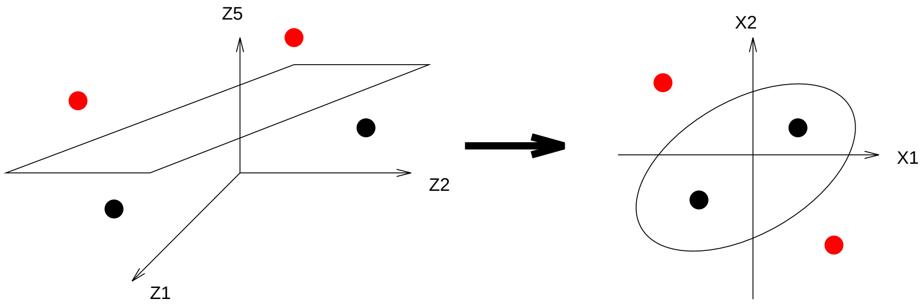


⋄ Polynomial decision rule ⋄

For two dimensional input space



Constructing a decision surface of polynomial degree 2 is equivalent to a linear hyperplane in 5 dimensional space.



$$\begin{array}{cccccc}\alpha_1 z_1 & +\alpha_2 z_2 & +\alpha_3 z_3 & +\alpha_4 z_4 & +\alpha_5 z_5 & +b=0 \\ \Downarrow & \Downarrow & \Downarrow & \Downarrow & \Downarrow & \\ x^1 & x^2 & (x^1)^2 & (x^2)^2 & x^1 x^2 & \end{array}$$

$\diamond$ Exponential explosion of dimensionality of feature space $\diamond$

Constructing a decision surface corresponding to a polynomial of degree two gives the mapping $z = \phi(x)$ from input to feature space:

$$x^1, \ldots, x^n$$

$$\Downarrow$$

$$z^1 = x^1, \ldots, z^n = x^n,$$

$$z^{n+1} = (x^1)^2, \ldots, z^{2n} = (x^n)^2,$$

$$z^{2n+1} = x^1 x^2, \ldots, z^N = x^n x^{n-1},$$

where $N = \frac{n(n+3)}{2}$ co-ordinates in feature space Z.

When constructing polynomials of degree 5 in 256-dimensional space the feature space is billion-dimensional.

⋄ Curse of Dimensionality ⋄

We can overcome the curse of dimensionality because we only need to calculate inner products between vectors in feature space.

$$(z_i \cdot z_j) = (\phi(x_i) \cdot \phi(x_j)) \Leftrightarrow K(x_i, x_j)$$

where $\phi(x)$ is the transformation from input space into feature space.

We do not need to perform the mapping explicitly. If function K satisfies Mercer's condition it describes an inner product.

◇ Examples of Kernels ◇

◇ A polynomial machine is constructed using:

$$K(x_i, x_j) = ((x_i \cdot x_j) + 1)^d, d = 1, \ldots$$

◇ A radial basis function machine with convolution function:

$$K(x_i, x_j) = \exp\{-\frac{|x_i - x_j|^2}{\sigma^2}\}$$

◇ A two layer neural network machine with convolution function:

$$K(x_i, x_j) = \tanh(b(x_i \cdot x_j) - c)$$

◇ Generalization Ability ◇

There are three reasons possible for generalization in an SV machine:

- ◇ Small dimensionality of feature space
- ◇ Large separating margin
- ◇ Small number of Support Vectors

SV machines rely on the last two reasons.

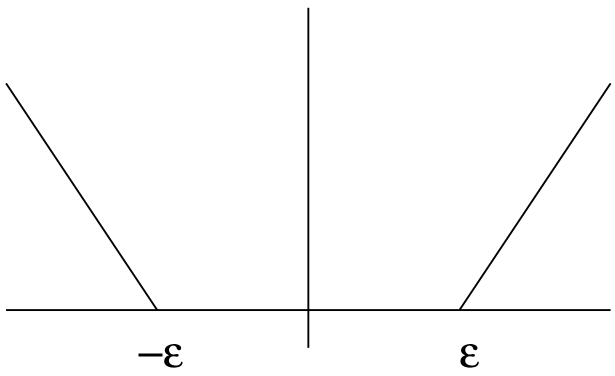
Classical statistical methods rely on the effective number of free parameters.

◇ Regression Estimation ◇

We can generalize the SV method for estimating real valued functions.

One new idea:

Errors in the approximation less than epsilon are ignored.



This is called the epsilon insensitive zone.

This plays the role of the margin and controls:

ACCURACY vs COMPLEXITY

(number of SVs)

◇ New Kernels ◇

- ◇ We can construct kernels that generate n-dimensional splines in high dimensional input space.
- ◇ We can construct kernels that generate Fourier expansions.
- ◇ Potentially, any series expansion.

IMPORTANT:

In many cases, the Kernel for high dimensional input space is a product of one dimensional Kernels.

◇ Support Vector Machines ◇

◇ SV machines are a general technique.

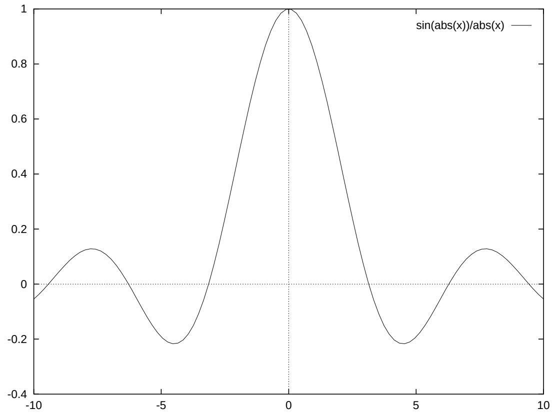
◇ SVM can estimate any type of function.

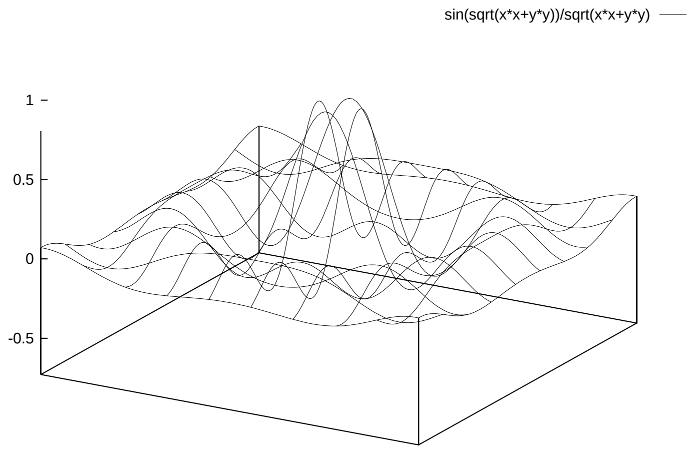
◇ SVM can solve linear operator equations.

◇ SVM can perform Positron Emission Tomography.

⋄ Toy examples, 1d and 2d Mexican Hat ⋄

$$y = \frac{sin(|x|)}{|x|}$$



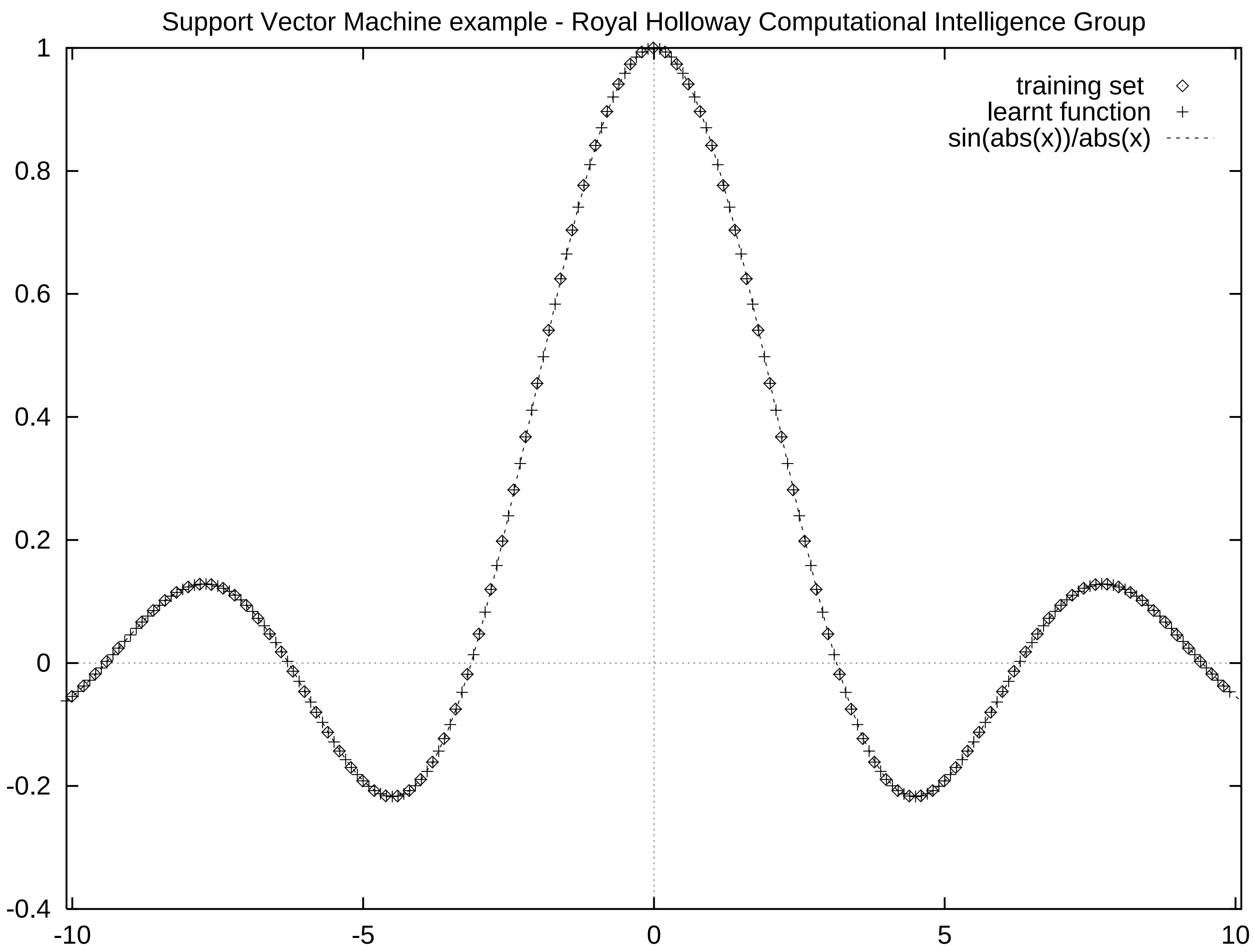


For our experiments we generated all points in a one or two dimensional input space (x_1) or (x_1, x_2) on a fixed square grid in the range $[-10 : 10]$.

[Vapnik, Golovich, Smola '96]

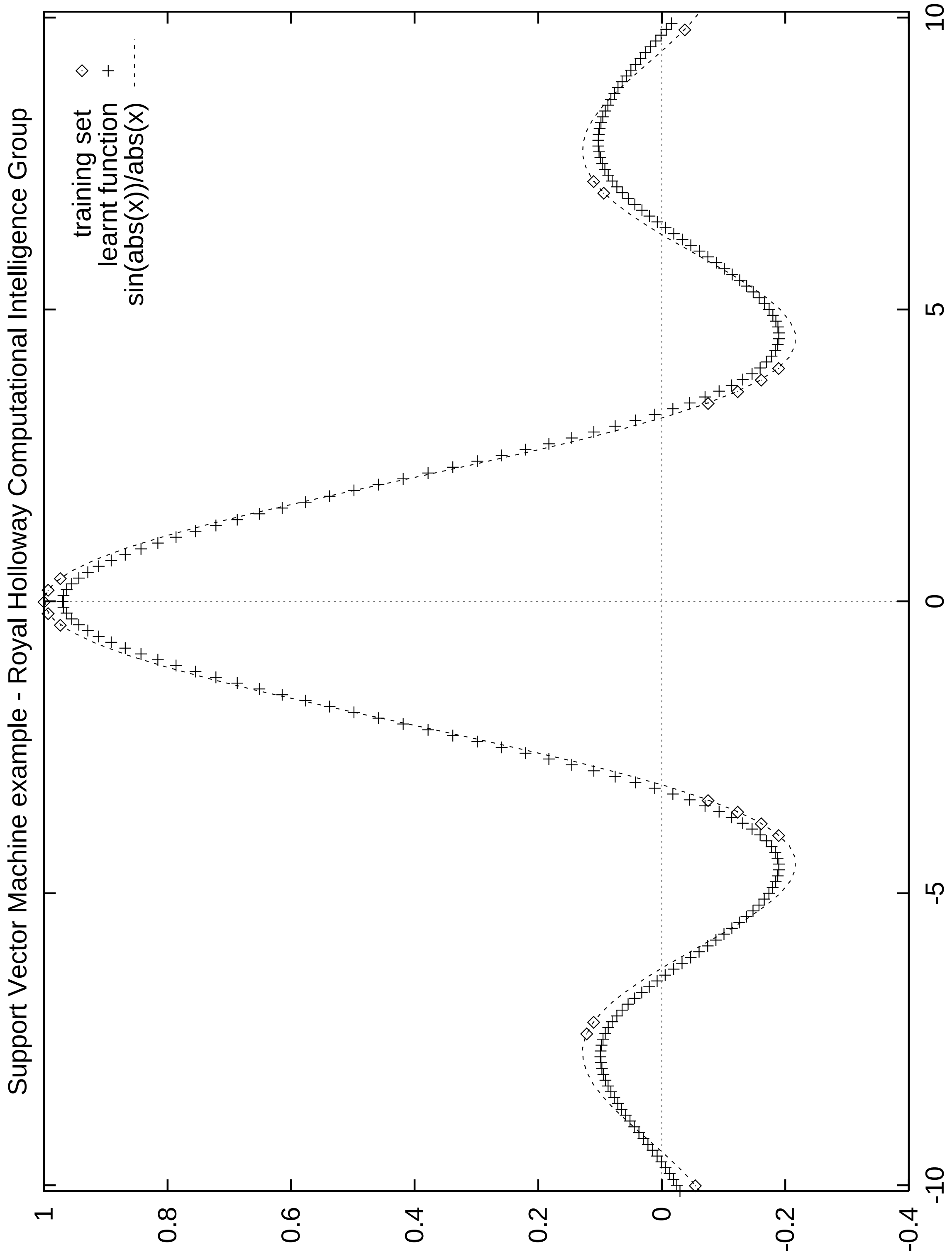
◇ Mexican Hat ◇

$\epsilon = 0$



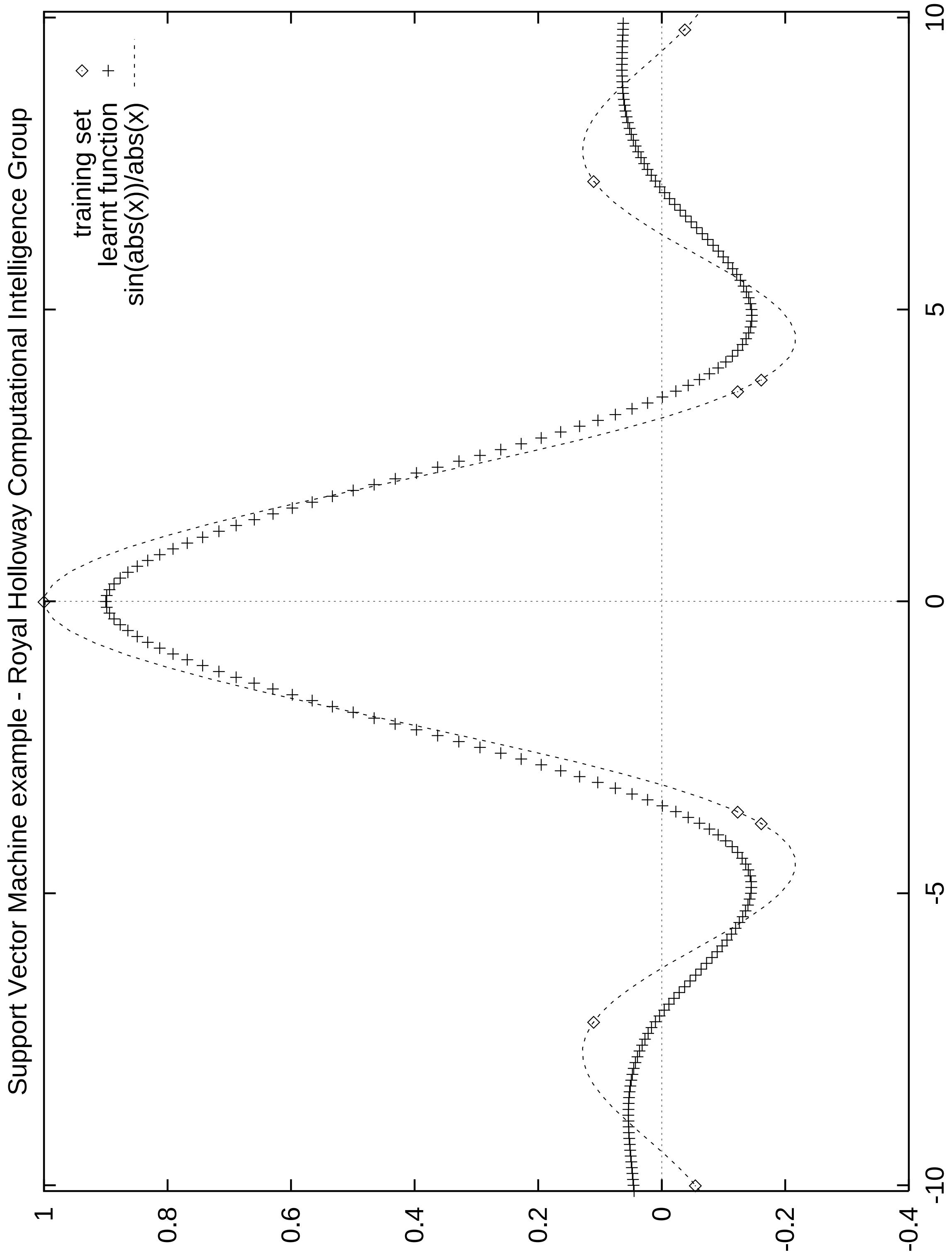
◇ Mexican Hat ◇

$\epsilon = 0.03$



◇ Mexican Hat ◇

$\epsilon = 0.1$



◇ Mexican Hat ◇

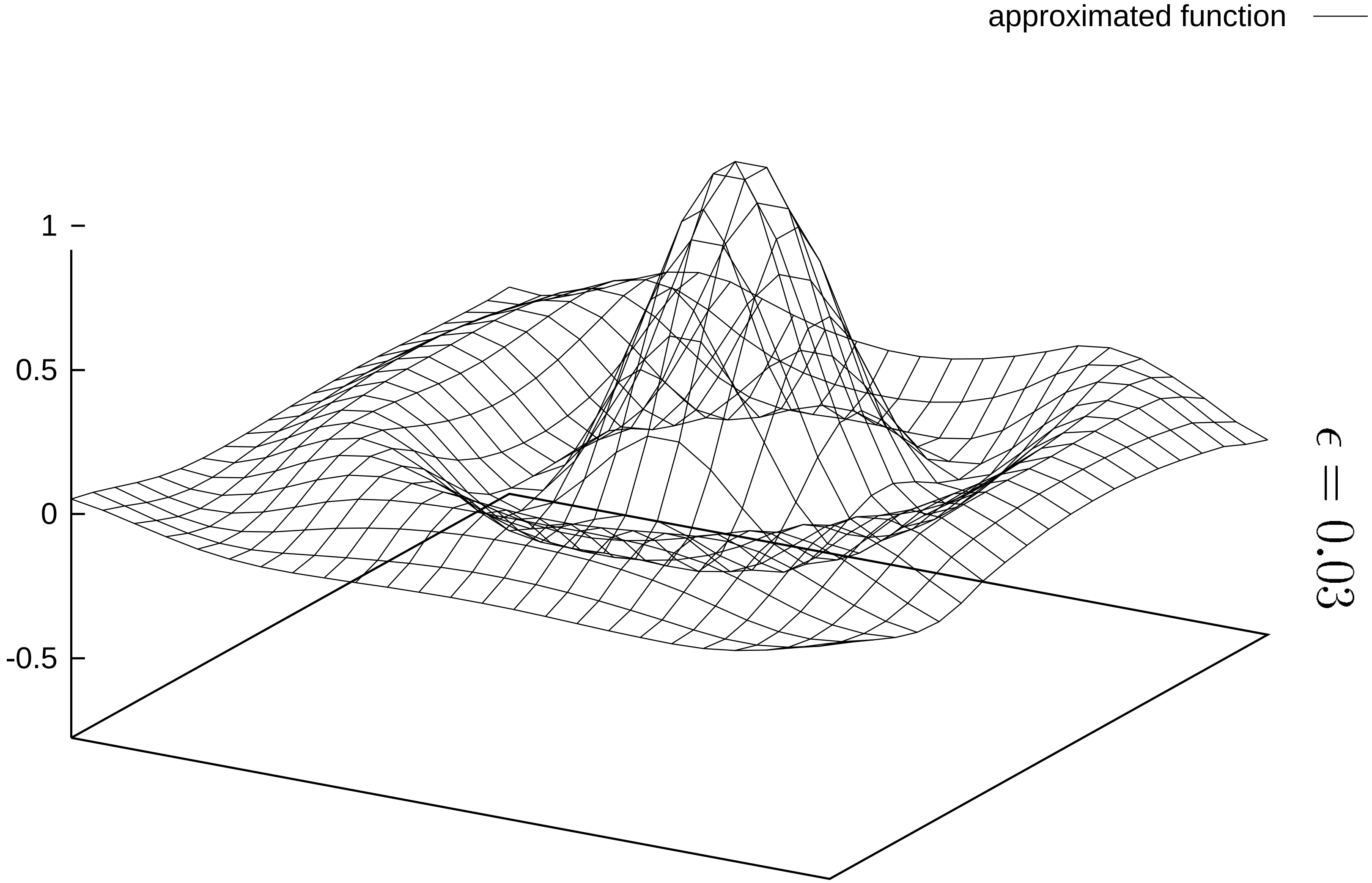
Support Vectors for the 1d case:

ϵ	Number of SVs
0	100
0.01	35
0.03	19
0.05	12
0.1	9
0.2	7

These results were obtained using 100 training examples.

⋄ Mexican Hat ⋄

$\epsilon = 0.03$



⋄ Mexican Hat ⋄

Support Vectors for the 3d case:

Training set size	Number of SVs
400	100
2025	201
7921	267

The accuracy for these experiments was $\epsilon = 0.03$.

⋄ Real Life Examples ⋄

⋄ US Postal Service Digits ⋄

(AT&T Research; Cortes, Burges, Schölkopf)

Training data: 7,300
Test data: 2,000
Input space dimensionality: 256

Classifier		Raw Error
Human Performance		2.5%
Decision tree, C4.5		16.2%
Best two layer neural network		5.9%
5 layer neural network (LeNet 1)		5.1%
SVM with kernel	Number of SV	
Polynomial	274	4.0%
RBF	291	4.1%
Neural Network	254	4.2%

⋄ NIST Digits ⋄

[Drucker, Schapire, Simard]
[LeCun et al]

Training data: 60,000
Test data: 10,000
Input space dimensionality: 400

Classifier	Error rate
LeNet 1	1.7%
LeNet 4	1.1%
LeNet 5	0.9%
SVM	0.84%
3 LeNet 4s w. boosting	0.7%

◇ The Boston Housing problem ◇

In this house price estimation problem, our spline SV machine is only out performed by another SVM.

Classifier	Error
MARS-1	14.37
MARS-3	15.91
POLYMARS	14.07
BAGGING	12.4
SV 13-dim splines	8.8
SV polynomial	7.2

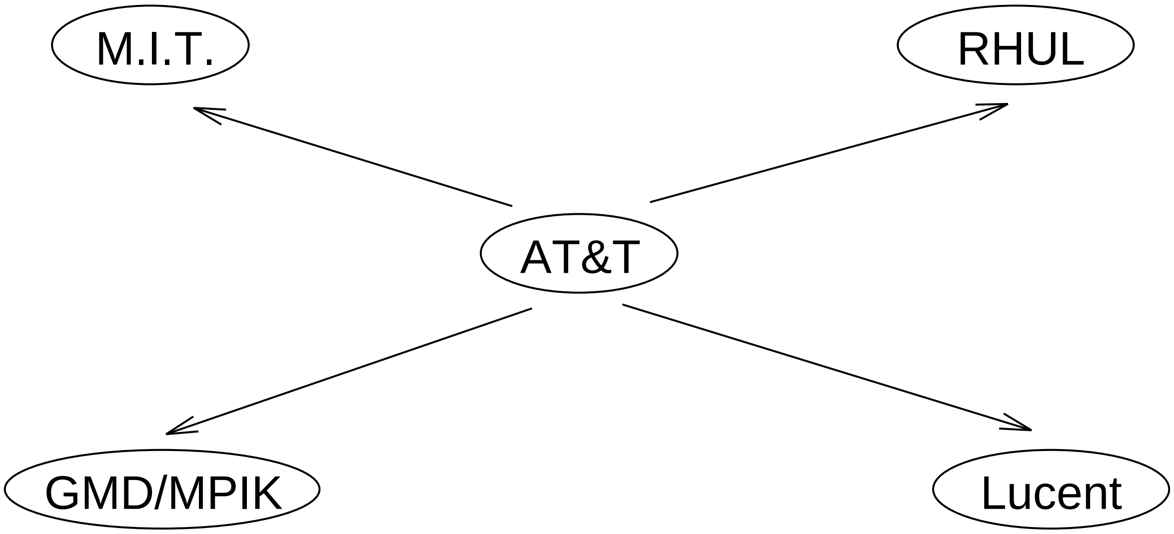
[Friedman]

[Brieman]

[Druker,Burges,Kaufman,Smola,Vapnik]

⋄ Who is working on SV machines? ⋄

We know of 4 state of the art implementations of the SV machine.



We are part of a large international research group, headed by the research team at AT&T.

◇ Implementational Issues ◇

- ◇ We expect to have 2,000-100,000 training examples.
- ◇ Optimization packages consider 200 variables as large.
- ◇ SV machines solve a simple QP under box constraints. With decomposition we can treat up to 4,000 variables or SVs (100,000 examples).
- ◇ A special type of decomposition was developed by E.Osuna at MIT to treat up to 100,000 SVs. This is slow though.

We are developing an approach which

- ◇ can treat a large number of training data; and
- ◇ is fast for a large number of SVs.

◇ Summary ◇

- ◇ SVM is a conceptually elegant machine.
- ◇ It is an effective and general method for representing complex functions in high dimensional space.
- ◇ It relies on a non-classical statistical justification (VC theory).
- ◇ It avoids the curse of dimensionality.
- ◇ It can control accuracy vs complexity in function estimation.
- ◇ SV machines still have many unexplored uses and applications.