Lecture 18: Multiclass Support Vector Machines

Hao Helen Zhang

Outlines

- Traditional Methods for Multiclass Problems
  - One-vs-rest approaches
  - Pairwise approaches
- Recent development for Multiclass Problems
  - Simultaneous Classification
  - Various loss functions
- Extensions of SVM

Multiclass Classification Setup

- Label: $\{-1, +1\} \to \{1, 2, \ldots, K\}$.
- Classification decision rule:

$$f : R^d \Longrightarrow \{1, 2, \ldots, K\}.$$

- Classification accuracy is measured by
  - Equal-cost: Generalization Error (GE)

$$\text{Err}(\mathbf{f}) = P(Y \neq f(\mathbf{X})).$$

  - Unequal-cost: the risk

$$R(f) = E_{Y,\mathbf{X}} C(Y, f(\mathbf{X})).$$

Traditional Methods

Main ideas:

(i) Decompose the multiclass classification problem into multiple binary classification problems.

(ii) Use the majority voting principle (a combined decision from the committee) to predict the label

Common approaches: simple but effective

- One-vs-rest (one-vs-all) approaches
- Pairwise (one-vs-one, all-vs-all) approaches

One-vs-rest Approach

One of the simplest multiclass classifier; commonly used in SVMs; also known as the one-vs-all (OVA) approach

(i) Solve K different binary problems: classify "class k" versus "the rest classes" for $k = 1, \cdots, K$.

(ii) Assign a test sample to the class giving the largest $f_k(x)$ (most positive) value, where $f_k(x)$ is the solution from the kth problem

Properties:

- Very simple to implement, perform well in practice
- Not optimal (asymptotically): the decision rule is not Fisher consistent if there is no dominating class (i.e. $\arg\max p_k(x) < \frac{1}{2}$).

Read: Rifkin and Klautau (2004) "In Defense of One-vs-all Classification"

Pairwise Approach

Also known as all-vs-all (AVA) approach

(i) Solve $\binom{K}{2}$ different binary problems: classify "class k" versus "class j" for all $j \neq k$. Each classifier is called g_{ij}.

(ii) For prediction at a point, each classifier is queried once and issues a vote. The class with the maximum number of (weighted) votes is the winner.

Properties:

- Training process is efficient, by dealing with small binary problems.
- If K is big, there are too many problems to solve. If $K = 10$, we need to train 45 binary classifiers.
- Simple to implement; perform competitively in practice.

Read: Park and Furnkranz (2007) "Efficient Pairwise Classification"

One Single SVM approach: Simultaneous Classification

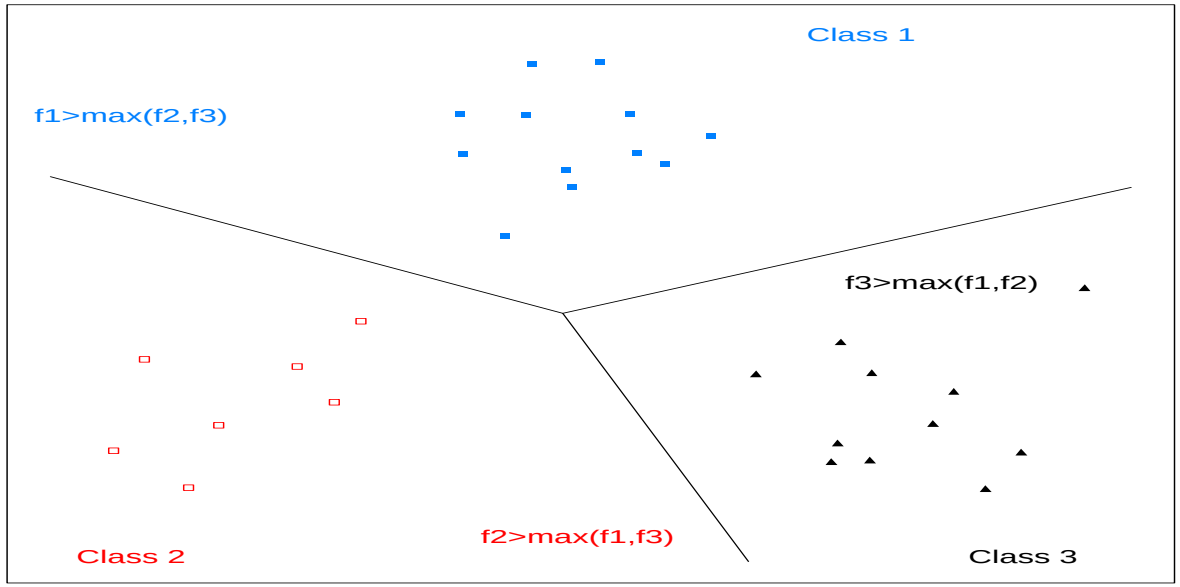
- Label: $\{-1, +1\} \to \{1, 2, \ldots, K\}$.
- Use one single SVM to construct a decision function vector

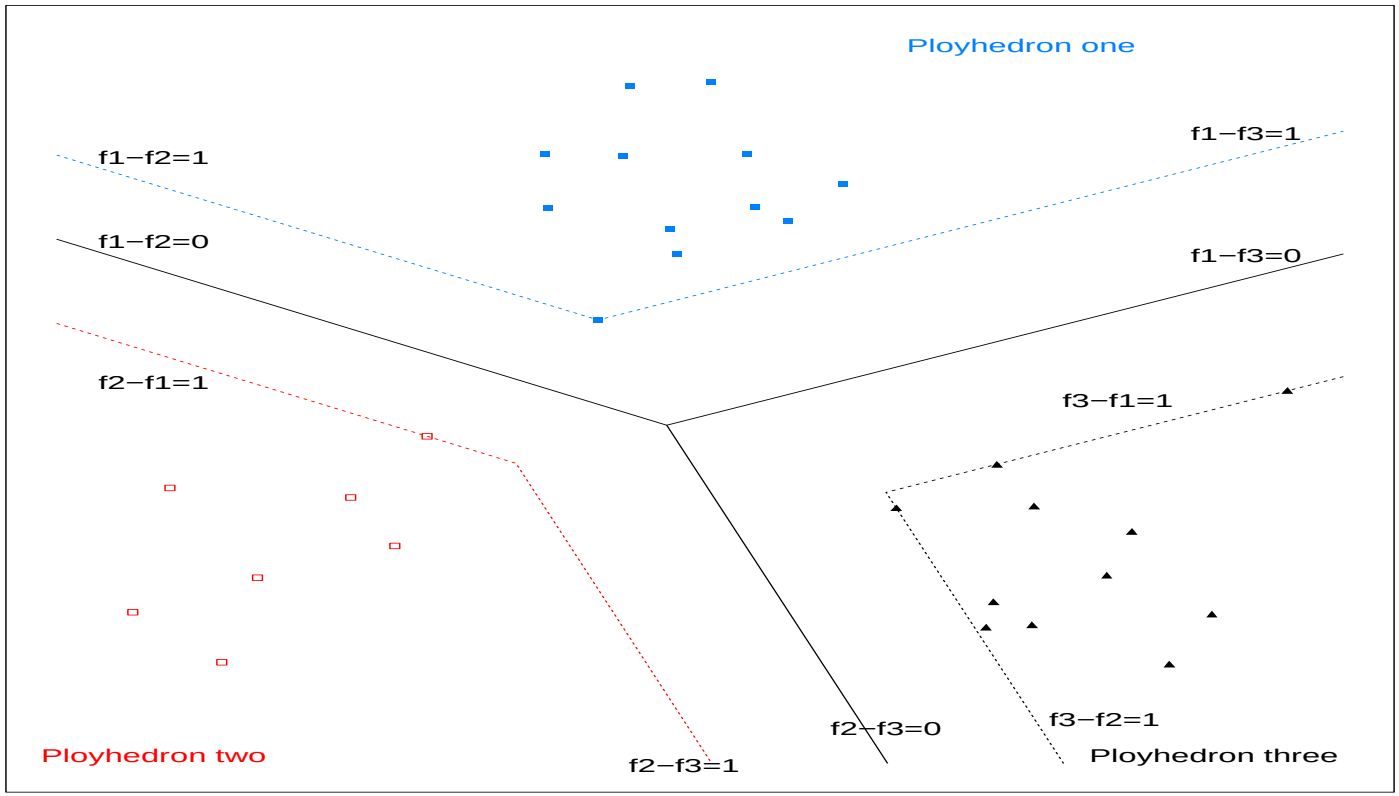
$$\mathbf{f} = (f_1, \ldots, f_K).$$

- Classifier (Decision rule):

$$f(\mathbf{x}) = \text{argmax}_{k=1,\cdots,K} f_k(\mathbf{x}).$$

- If $K = 2$, there is one f_k and the decision rule is $\text{sign}(f_k)$.
- In some sense, multiple logistic regression is a simultaneous classification procedure


Class 1
f1>max(f2,f3)
f3>max(f1,f2)
f2>max(f1,f3)
Class 2
Class 3
x2
x1


Ployhedron one
f1−f2=1
f1−f2=0
f2−f1=1
f1−f3=1
f1−f3=0
f3−f1=1
f3−f2=1
Ployhedron three
Ployhedron two
f2−f3=1
f2−f3=0

SVM for Multiclass Problem

Multiclass SVM: solving one single regularization problem by imposing a penalty on the values of $f_y(\mathbf{x}) - f_l(\mathbf{x})$'s.

- Weston and Watkins (1999)
- Cramer and Singer (2002)
- Lee et al. (2004)
- Liu and Shen (2006); multiclass ψ-learning: Shen et al. (2003)

Various Multiclass SVMs

- Weston and Watkins (1999):
  a penalty is imposed only if $f_y(\mathbf{x}) < f_k(\mathbf{x}) + 2$ for $k \neq y$.
  - Even if $f_y(\mathbf{x}) < 1$, a penalty is not imposed as long as $f_k(\mathbf{x})$ is sufficiently small for $k \neq y$;
  - Similarly, if $f_k(\mathbf{x}) > 1$ for $k \neq y$, we do not pay a penalty if $f_y(\mathbf{x})$ is sufficiently large.

$$L(y, \mathbf{f}(\mathbf{x})) = \sum_{k \neq y} [2 - (f_y(\mathbf{x}) - f_k(\mathbf{x}))]_+ .$$

- Lee et al. (2004): $L(y, \mathbf{f}(\mathbf{x})) = \sum_{k \neq y} [f_k(\mathbf{x}) + 1]_+$.
- Crammer and Singer (2002): Liu and Shen (2006),
  $L(y, \mathbf{f}(\mathbf{x})) = [1 - \min_k \{f_y(\mathbf{x}) - f_k(\mathbf{x})\}]_+$.

To avoid the redundancy, a sum-to-zero constraint $\sum_{k=1}^{K} f_k = 0$ is sometimes enforced.

Linear Multiclass SVMs

For linear classification problems, we have

$$f_k(\mathbf{x}) = \boldsymbol{\beta}_k \mathbf{x} + \beta_{0k}, \quad k = 1, \cdots, K.$$

The sum-to-zero constraint can be replaced by

$$\sum_{k=1}^{K} \beta_{0k} = 0, \quad \sum_{k=1}^{K} \boldsymbol{\beta}_k = \mathbf{0}.$$

The optimization problem becomes

$$\min_{\mathbf{f}} \sum_{i=1}^{n} L\left(y_i, \mathbf{f}(\mathbf{x}_i)\right) + \lambda \sum_{k=1}^{K} \|\boldsymbol{\beta}_k\|^2$$

subject to the sum-to-zero constraint.

Nonlinear Multiclass SVMs

To achieve the nonlinear classification, we assume

$$f_k(\mathbf{x}) = \boldsymbol{\beta}_k' \phi(\mathbf{x}) + \beta_{k0}, \quad k = 1, \cdots, K.$$

where $\phi(\mathbf{x})$ represents the basis functions in the feature space $\mathcal{F}$.

- Similar to the binary classification, the nonlinear MSVM can be conveniently solved using a kernel function.

Regularization Problems for Nonlinear MSVMs

We can represent the MSVM as the solution to a regularization problem in the RKHS.

- Assume that

$$\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_K(\mathbf{x})) \in \prod_{k=1}^{K} (\{1\} + \mathcal{H}_k)$$

  under the sum-to-zero constraint.

- Then a MSVM classifier can be derived by solving

$$\min_{\mathbf{f}} \sum_{i=1}^{n} L\left(y_i, \mathbf{f}(\mathbf{x}_i)\right) + \lambda \sum_{k=1}^{K} \|g_k\|_{\mathcal{H}_k}^2,$$

where $f_k(\mathbf{x}) = g_k(\mathbf{x}) + \beta_{0k}, g_k \in \mathcal{H}_k, \beta_{0k} \in \mathcal{R}$.

Generalized Functional Margin

Given $(\mathbf{x}, y)$, a reasonable decision vector $\mathbf{f}(\mathbf{x})$ should

- encourage a large value for $f_y(\mathbf{x})$
- have small values for $f_k(\mathbf{x}), k \neq y$.

Define the $K-1$-vector of relative differences as

$$\mathbf{g} = (f_y(\mathbf{x}) - f_1(\mathbf{x}), \cdots, f_y(\mathbf{x}) - f_{y-1}(\mathbf{x}), f_y(\mathbf{x}) - f_{y+1}(\mathbf{x}), \cdots, f_y(\mathbf{x}) - f_K(\mathbf{x})).$$

Liu et al. (2004) called the vector $\mathbf{g}$ the *generalized functional margin* of $\mathbf{f}$

- $\mathbf{g}$ characterizes correctness and strength of classification of $\mathbf{x}$ by $\mathbf{f}$.
- $\mathbf{f}$ indicates a correct classification of $(\mathbf{x}, y)$ if

$$\mathbf{g}(\mathbf{f}(\mathbf{x}), y) > \mathbf{0}_{k-1}.$$

0-1 Loss with Functional Margin

A point $(\boldsymbol{x}, y)$ is misclassified if $y \neq \arg\max_k f_k(\boldsymbol{x})$.
Define the multivariate sign function as

$$\begin{aligned} \text{sign}(\mathbf{u}) = 1 \quad & \text{if} \quad \mathbf{u}_{\min} = \min(u_1, \cdots, u_m) > 0, \\ -1 \quad & \text{if} \quad \mathbf{u}_{\min} \leq 0. \end{aligned}$$

where $\mathbf{u} = (u_1, \cdots, u_m)$. Using the functional margin,

- The 0-1 loss becomes

$$I(\min \boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x}), y) < 0) = \frac{1}{2}\left[1 - \text{sign}(\boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x}), y))\right].$$

- The GE becomes to

$$R[f] = \frac{1}{2} E\left[1 - \text{sign}(\boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x}), y))\right].$$

Generalized Loss Functions Using Functional Margin

A natural way to generalize the binary loss is

$$\sum_{i=1}^{n} \ell(\min \boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x}_i), y_i)).$$

In particular, the loss function $L(y, \mathbf{f}(\mathbf{x}))$ can be expressed as $V(\mathbf{g}(\mathbf{f}(\mathbf{x}), y))$ with

- Weston and Watkins (1999): $V(\mathbf{u}) = \sum_{j=1}^{K-1} [2 - u_j]_+$.
- Lee et al. (2004): $V(\mathbf{u}) = \sum_{j=1}^{K-1} [\frac{\sum_{c=1}^{K-1} u_c}{K} - u_j + 1]_+$.
- Liu and Shen (2006): $V(\mathbf{u}) = [1 - \min_j u_j]_+$.

All of these loss functions are the upper bounds of the 0-1 loss.

Small Round Blue Cell Tumors of Childhood

- Khan et al. (2001) in *Nature Medicine*
- Tumor types: neuroblastoma (NB), rhabdomyosarcoma (RMS), non-Hodgkin lymphoma (NHL) and the Ewing family of tumors (EWS)
- Number of genes : 2308
- Class distribution of data set

Data set	EWS	BL(NHL)	NB	RMS	total
Training set	23	8	12	20	63
Test set	6	3	6	5	20
Total	29	11	18	25	83

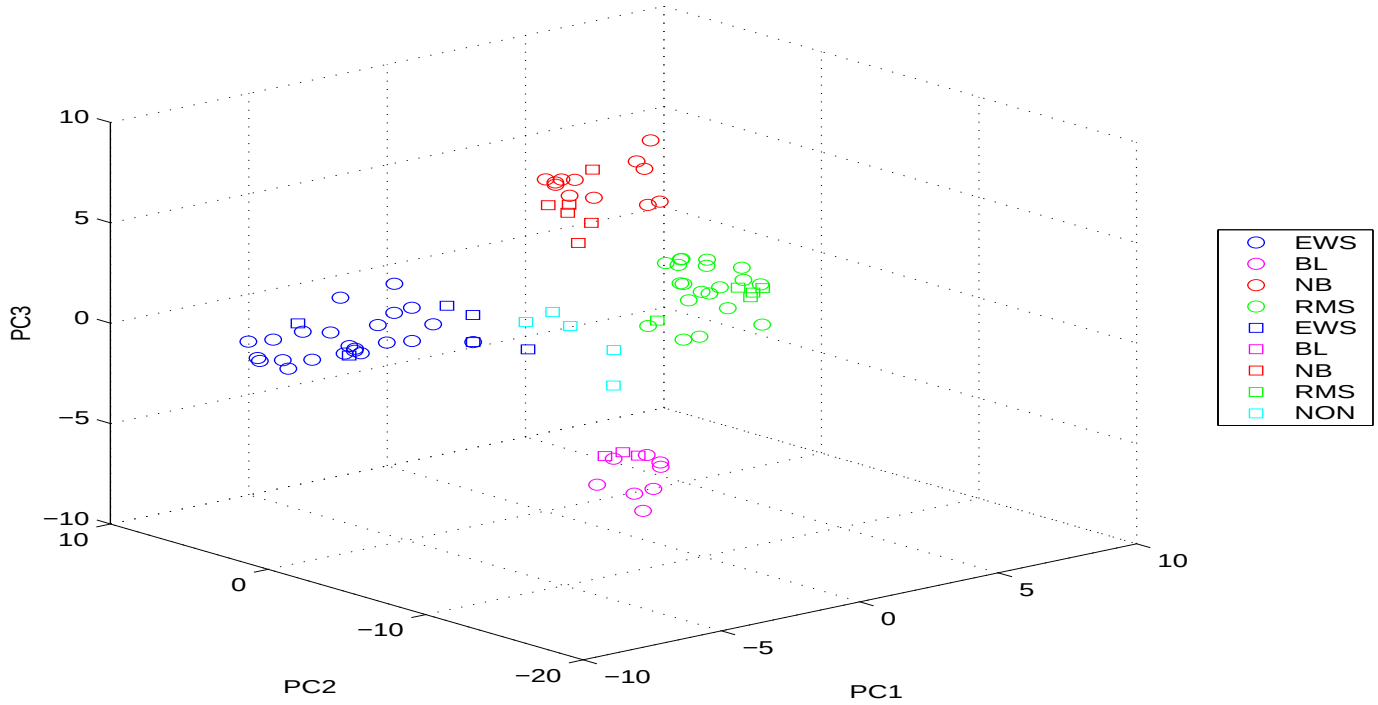


Figure 3: Three principal components of 100 gene expression levels (circles: training samples, squares: test samples including non SRBCT samples). The tumor types are distinguished by colors.

Characteristics of Support Vector Machines

- High accuracy, high flexibility
- Naturally handle large dimensional data
- Sparse representation of the solutions (via support vectors): fast for making future prediction
- No probability estimates (hard classifiers)

Other Active Problems in SVM

- Variable/Feature Selection
  - Linear SVM: Bradley and Mangasarian (1998), Guyon et al. (2000), Rakotomamonjy (2003), Jebara and Jaakkola (2000)
  - Nonlinear SVM: Weston et al. (2002), Grandvalet (2003), Basis pursuit (Zhang 2003), COSSO selection (Lin & Zhang (2003)
- Proximal SVM – faster computation
- Robust SVM – get rid of outliers
- Choice of kernels

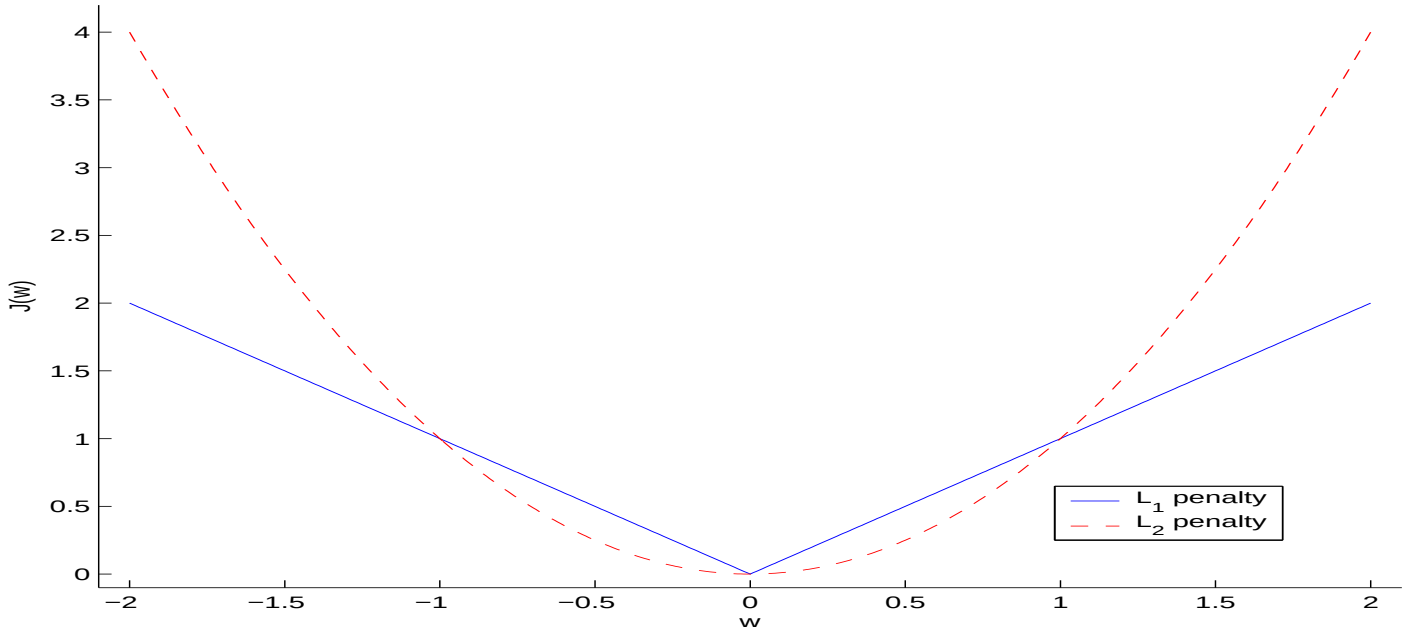
The L_1 SVM

- Replace the L_2 penalty by the L_1 penalty.
- The L_1 penalty tends to give sparse solutions.
- For $f(\mathbf{x}) = h(\mathbf{x})^T\boldsymbol{\beta} + \beta_0$, the L_1 SVM solves

$$\min_{\beta_0,\boldsymbol{\beta}} \sum_{i=1}^{n}[1 - y_i f(\mathbf{x}_i)]_+ + \lambda \sum_{j=1}^{d} |\beta_j|. \tag{1}$$

- The solution will have at most n nonzero coefficients β_j.

L_1 Penalty versus L_2 Penalty



Robust Support Vector Machines

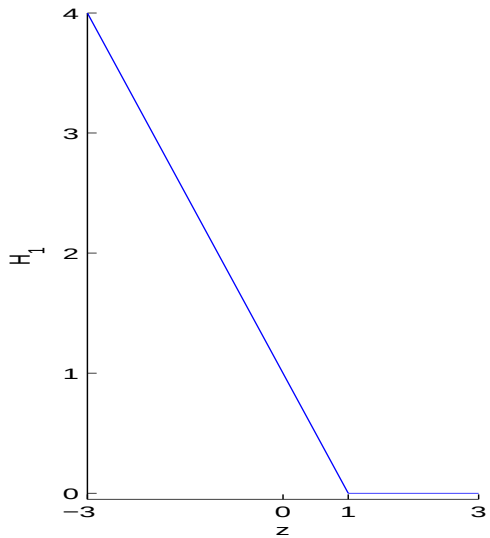
- Hinge loss is unbounded; sensitive to outliers (e.g. wrong labels etc)
- Support Vectors: $y_i f(\mathbf{x}_i) \leq 1$.
- Truncated hinge loss: $T_s(u) = H_1(u) - H_s(u)$, where

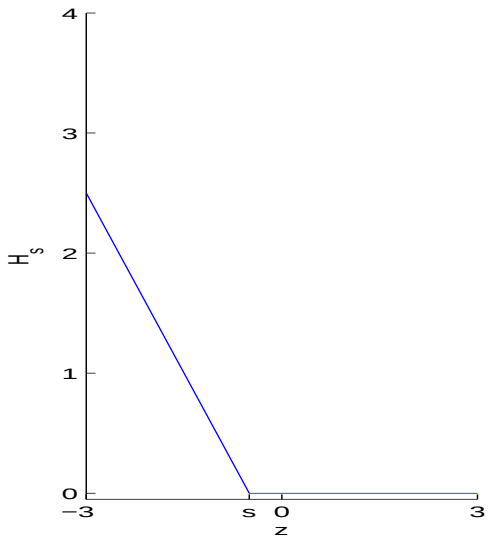
$$H_s(u) = [s - u]_+ .$$

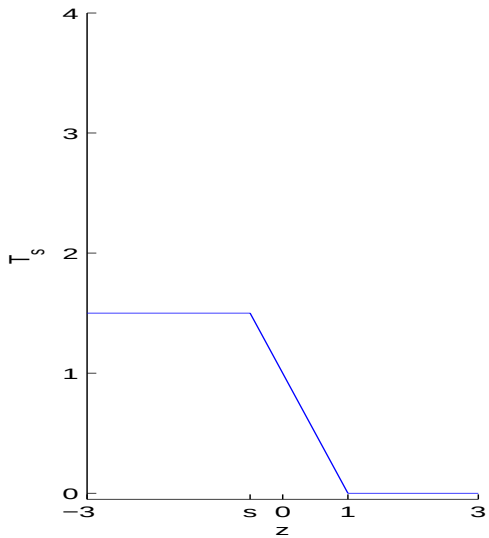
- Remove some "bad" SVs (Wu and Liu, 2006).

Decomposition: Difference of Convex Functions

- Key: D.C. decomposition (Diff. Convex functions).
- $T_s(u) = H_1(u) - H_s(u)$.







D.C. Algorithm

D.C. Algorithm: The Difference Convex Algorithm for minimizing
$J(\Theta) = J_{vex}(\Theta) + J_{cav}(\Theta)$

1. Initialize Θ_0.
2. Repeat $\Theta_{t+1} = \text{argmin}_{\Theta}(J_{vex}(\Theta) + \langle J'_{cav}(\Theta_t), \Theta - \Theta_t \rangle)$ until convergence of Θ_t.

- The algorithm converges in finite steps (Liu et al. (2005)).
- Choice of initial values: Use SVM's solution.
- RSVM: The set of SVs is a only a SUBSET of the original one!
- Nonlinear learning can be achieved by the kernel trick.