

Robust Truncated-Hinge-Loss SVM

Yufeng Liu

Department of Statistics and Operations Research

Carolina Center for Genome Sciences

University of North Carolina at Chapel Hill

`http://www.unc.edu/~yfliu`

Joint work with Yichao Wu, UNC-CH and Princeton

Outline

- General Framework
- Binary Support Vector Machine
- Binary ψ -Learning
- The New Methodology
- Multicategory Classification
- Variable Selection
- Algorithm
- Numerical Examples and Discussions

General Framework

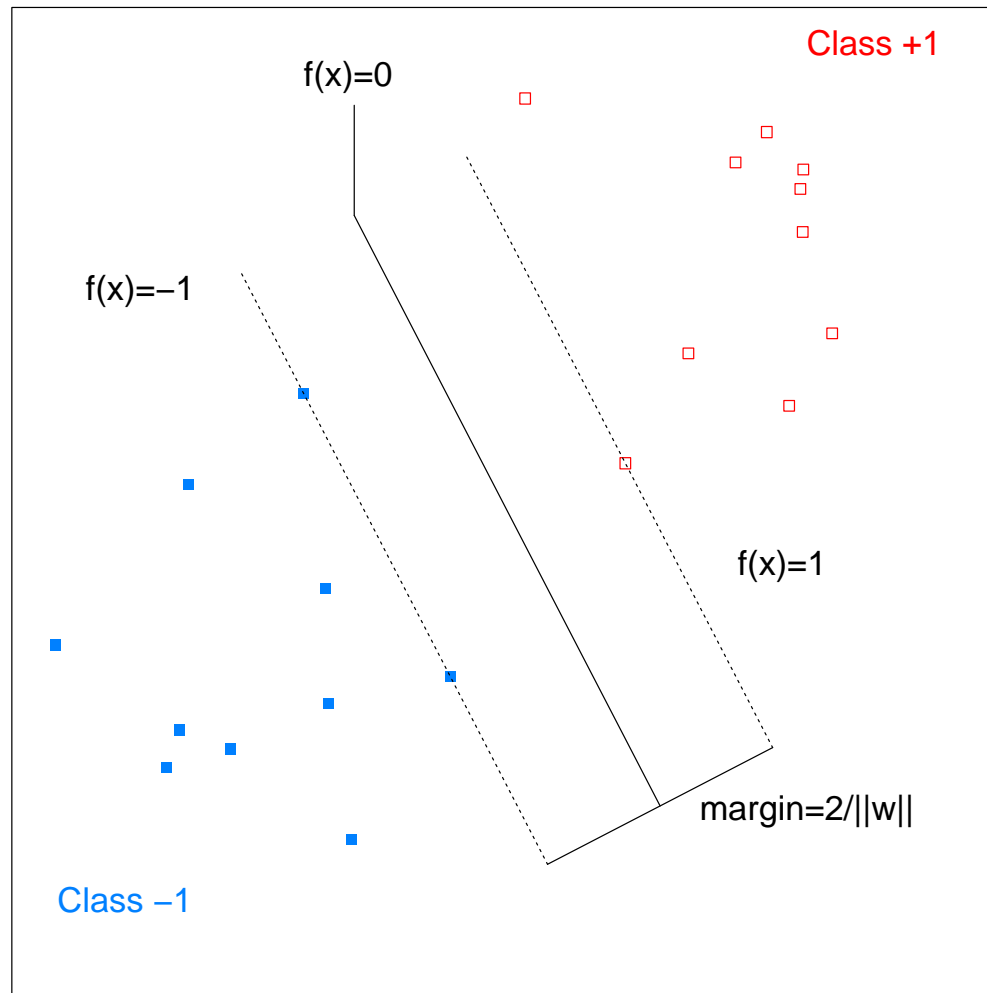
- Supervised learning: Given training data $\{(\mathbf{x}_i, y_i)_{i=1}^n\}$, i.i.d. \sim unknown $P(\mathbf{x}, y)$.
 - input $\mathbf{x}_i \in R^d$ as predictor;
 - outcome y_i as class.
- Build a prediction model, or classifier:
 - enable us to do prediction.
- Good classifier: accurately predicts the class y for given \mathbf{x} .

Classification Methods

- Traditional statistical methods
Linear/Quadratic Discriminant Analysis, Nearest Neighbor, Logistic Regression, etc.
- Machine learning
Margins → SVM (Boser, Guyon, & Vapnik, 1992, Vapnik, 1995),
Boosting (Freund & Schapire, 1997),
Distance Weighted Discrimination (Marron, Todd,
& Ahn 2004),
Import Vector Machine (Zhu & Hastie, 2005),
 ψ -Learning (Shen, Tseng, Zhang, & Wong, 2003, Liu &
Shen, 2006), etc.

Binary SVM

- A powerful classification method (Vapnik (1998)).
 - $y \in \{\pm 1\}$;
Estimate $f(\mathbf{x})$ with classification rule $\text{sign}[f(\mathbf{x})] : R^d \rightarrow \{\pm 1\}$.
 - Separable: linear SVM searches for optimal hyperplane $f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$ ($\mathbf{w} \in R^d, b \in R^1$) s.t.
 - $y_i f(\mathbf{x}_i) \geq 1; \forall i$ ($y_i f(\mathbf{x}_i)$: functional margin).
 - Geometric margin $2/\|\mathbf{w}\|$ is maximized.
- “Support Vectors”: points on $f(\mathbf{x}) = \pm 1$; determine the solution.



Plot of the decision boundary defined by $f(x) = 0$, the geometric margin $\gamma = \frac{2}{\|w\|}$, and three SVs on $f(x) = \pm 1$.

Binary SVM

- Nonseparable: “zero”-error not attainable \rightarrow “slack variables” $\{\xi_i\}_{i=1}^n$

$$\min_{b, \mathbf{w}, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i$$

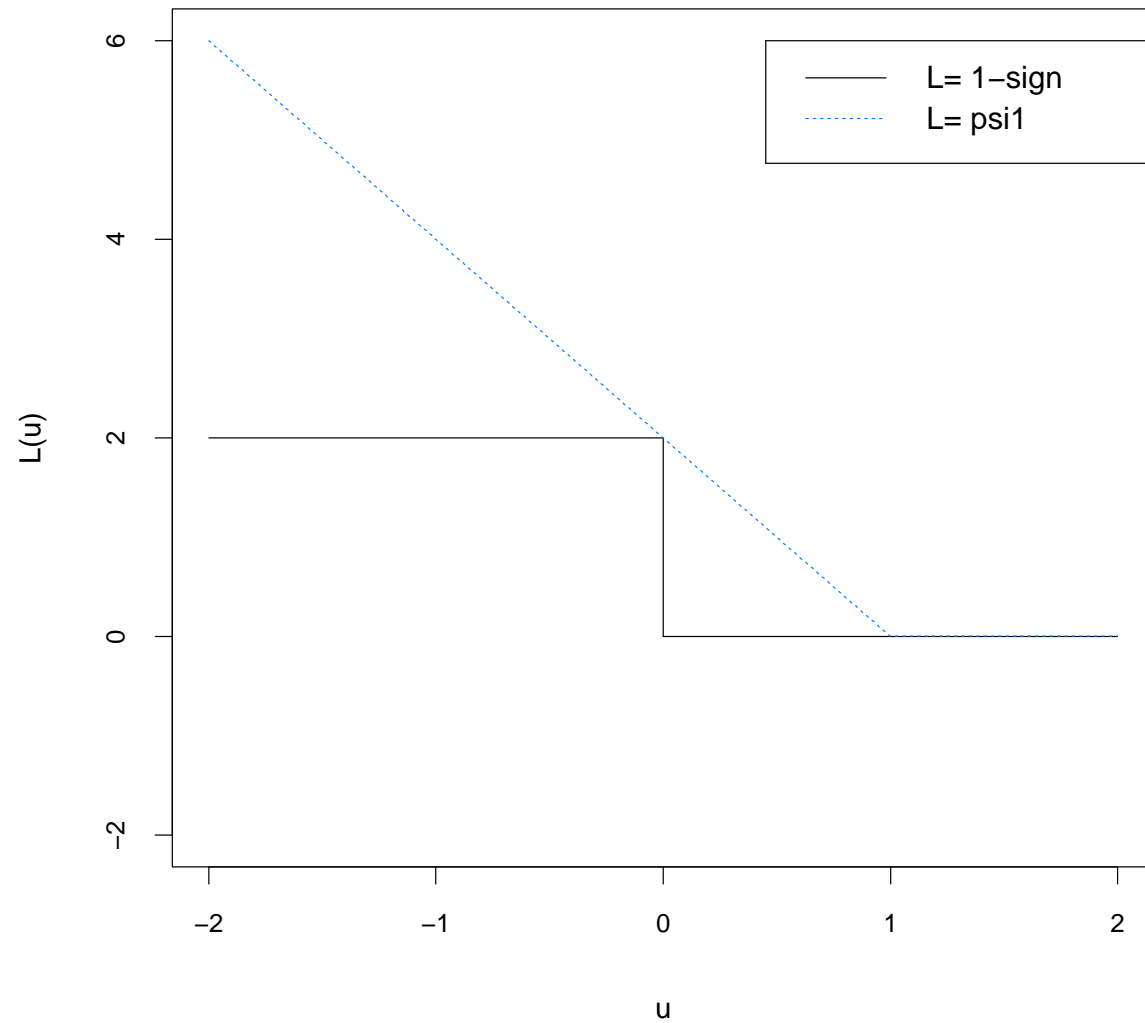
subject to

$$y_i f(\mathbf{x}_i) \geq (1 - \xi_i), \quad \xi_i \geq 0; \quad i = 1, \dots, n,$$

where $C > 0$ is a tuning parameter.

- Equivalent to $\min_{b, \mathbf{w}} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n H_1(y_i f(\mathbf{x}_i))$,
 $H_1(u) = [1 - u]_+$ (Hinge Loss).

The Hinge Loss for SVM



The dual problem for SVM

$$\min L_D(\boldsymbol{\alpha}) = \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle - \sum_{i=1}^n \alpha_i$$

subject to: $0 \leq \alpha_i \leq C, \quad i = 1, 2, \dots, n$

$$\sum_{i=1}^n \alpha_i y_i = 0$$

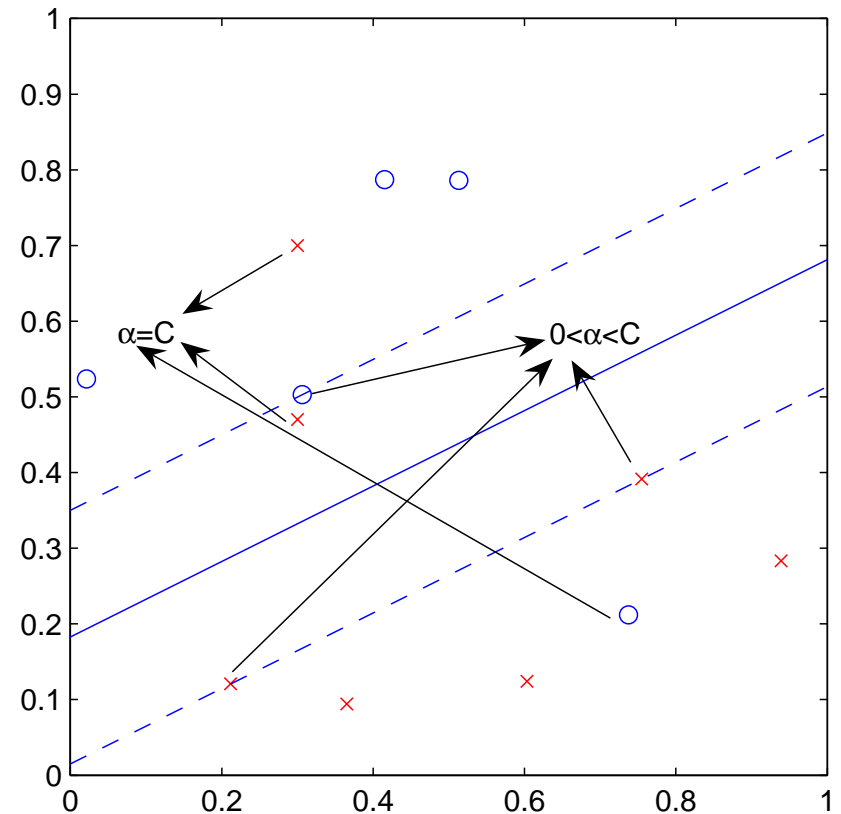
- Can be solved by quadratic programming.
- Recover \mathbf{w} : $\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i$
- Kernel Trick: Replace $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ by $K(\mathbf{x}_i, \mathbf{x}_j)$ and $f(\mathbf{x}) = \sum_{i=1}^n y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b$.

Support Vectors

- $\alpha_i = 0 \rightarrow y_i f(\mathbf{x}_i) > 1$;
not needed in constructing $f(\mathbf{x})$.

Support vectors:

- $0 < \alpha_i < C \rightarrow y_i f(\mathbf{x}_i) = 1$ (Solve b).
- $\alpha_i = C \rightarrow y_i f(\mathbf{x}_i) < 1$.



Regularization

- Regularization framework

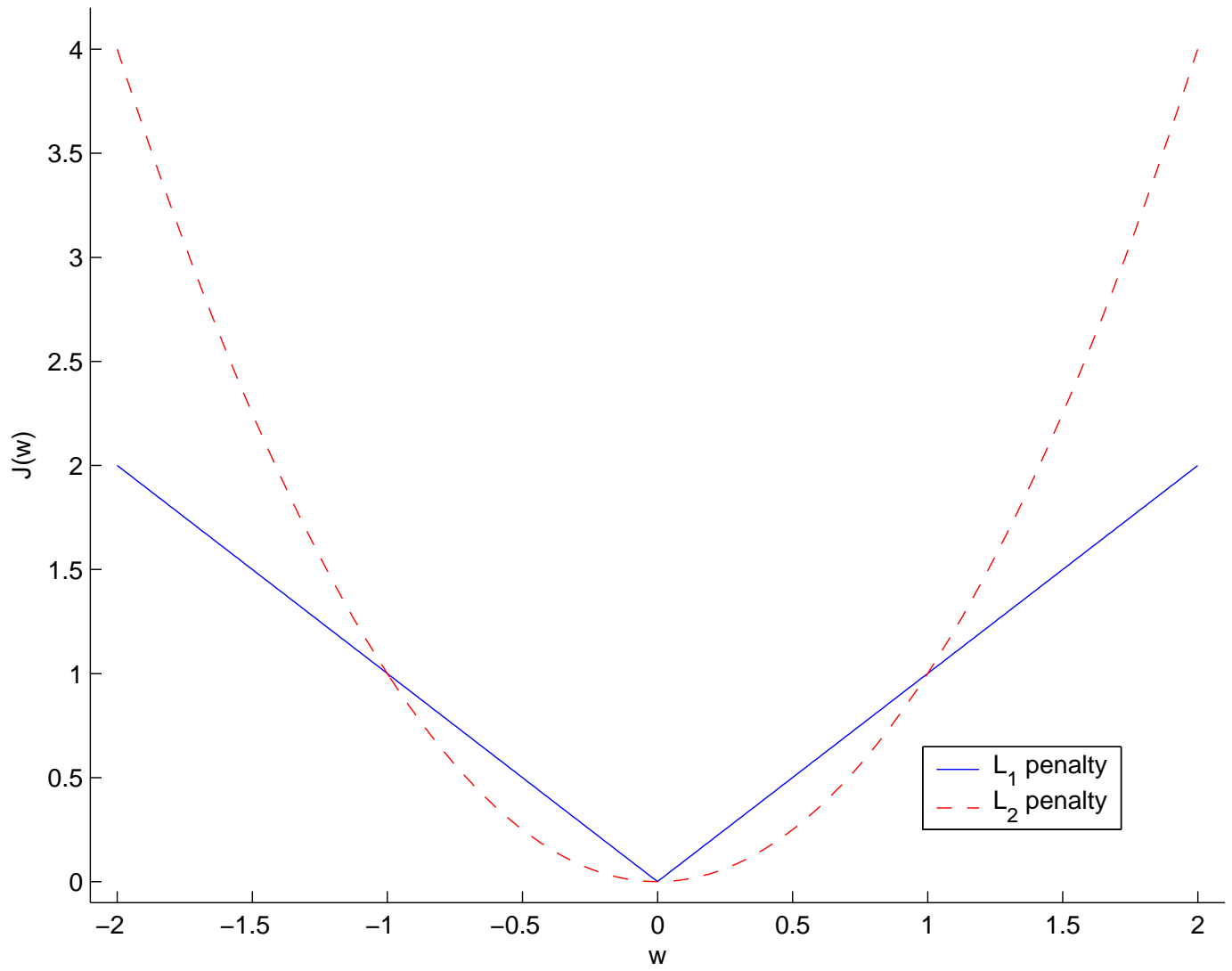
$$\min_f J(f) + C \sum_{i=1}^n l(f(\mathbf{x}_i), y_i).$$

- Regularization term $J(f)$: the roughness penalty of f ;
Loss l : data fit measure.
- $J(f)$:
 - Linear case : $\frac{1}{2} \|\mathbf{w}\|^2$ (L_2 penalty) or $\sum_{j=1}^d |w_j|$ (L_1 penalty, LASSO-type penalty, Tibshirani (1996), Zhu et al. (2003)), L_q penalty (Frank and Friedman (1993), Liu et al. (2006)), Elastic Net (Zou and Hastie, 2005, Wang, Zhu, and Zou (2006)), SCAD (Zhang et al. (2006)), Hybrid of L_0 and L_1 (Liu and Wu (2006b)).

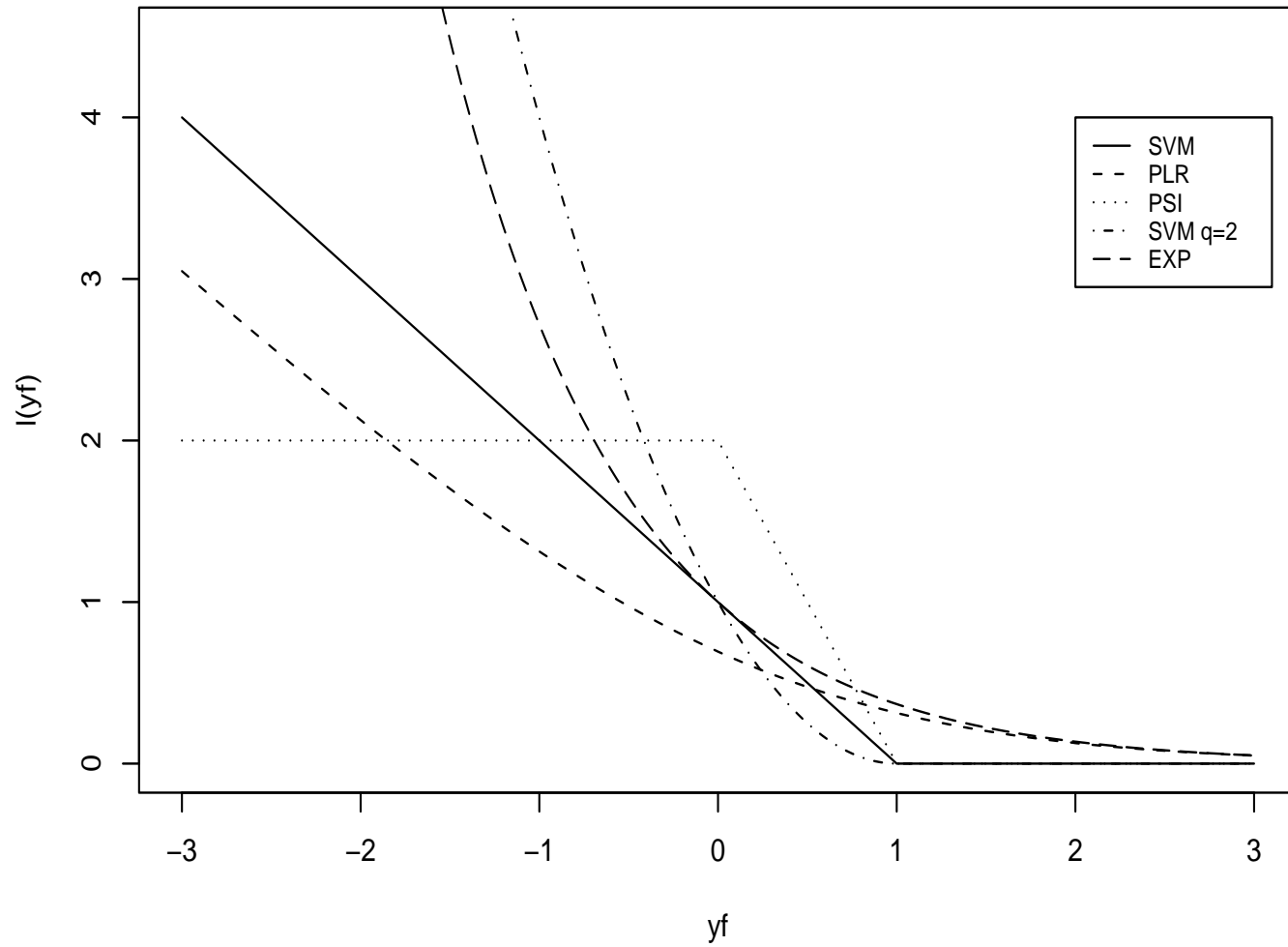
Regularization

- Nonlinear: Apply linear learning to a nonlinear feature space \mathcal{F} induced by kernel $K(\cdot, \cdot) : S \times S \rightarrow R$.
 - K satisfies Mercer's Theorem;
 - Regularization: $f(\mathbf{x}) = h(\mathbf{x}) + b$ with $h(\mathbf{x}) \in \mathcal{H}_K$ (Wahba, 1998);
 - Representer Theorem (Kimeldorf & Wahba, 1971)
 $h(\mathbf{x}) = \sum_{i=1}^n v_i K(\mathbf{x}_i, \mathbf{x});$
 $J(f) = \|h\|_{\mathcal{H}_K}^2 = \frac{1}{2} \mathbf{v}' \mathbf{K} \mathbf{v}$, where $\mathbf{K}(i, j) = K(\mathbf{x}_i, \mathbf{x}_j)$.

L_1 Penalty versus L_2 Penalty



Different Losses



Binary ψ -Learning

- Generalization error (GE):

$$P(Y f(\mathbf{X}) < 0) = \frac{1}{2} E(1 - \text{sign}(Y f(\mathbf{X}))).$$

- Goal: Targets on GE directly (Shen et al., 2003)

$$l(f(\mathbf{x}_i), y_i) = \psi(y_i f(\mathbf{x}_i))$$

- $1 - \text{sign} \rightarrow \psi$ (non-increasing): Solve scaling problem.

$$R \geq \psi(u) > 0 \quad u \in (0, \tau],$$

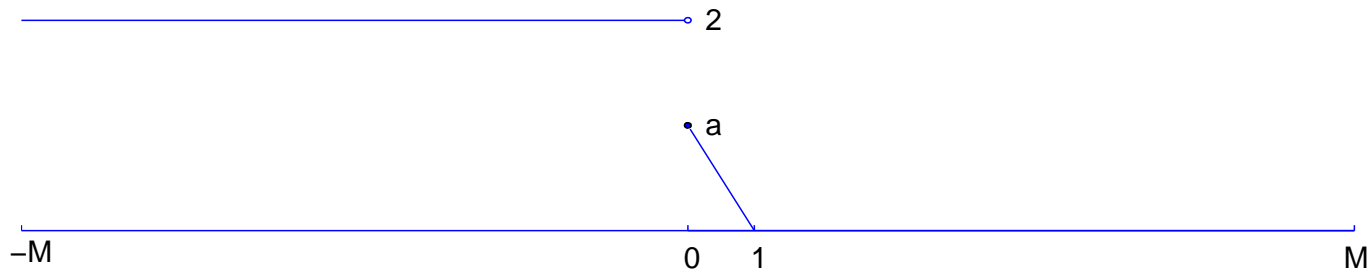
$$\psi(u) = 1 - \text{sign}(u) \quad \text{o.w.}$$

- Has potentials over SVM.

Piecewise Linear ψ -Loss

$$\begin{aligned}\psi(u) &= a - au \text{ if } u \in (0, 1], \\ &= 1 - \text{sign}(u) \text{ if } u \notin (0, 1],\end{aligned}$$

where a is a constant in $(0, 2]$.

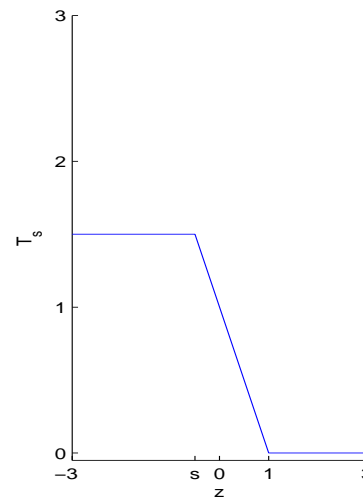
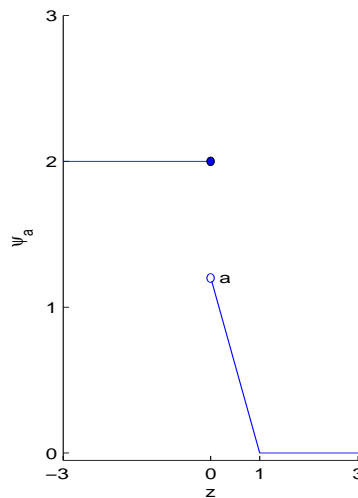
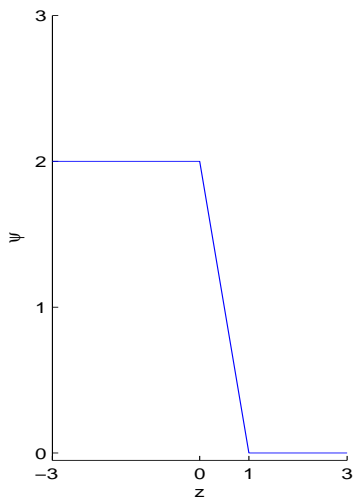


Implementation and Choice of a

- When $a = 2$, ψ loss is continuous: **D.C. programming** (Liu, Shen, and Doss, 2005)
Key: D.C. decomposition (Diff. Convex func.).
 - DCA (An and Tao, J. Global Optimization, 1997).
 - Sequence of convex optimization; feasible for large-scale problems.
- With any $a \in (0, 2]$, **Mixed Integer Programming (MIP)** (Liu and Wu, 2006)
 - More flexible than DCA.
 - Numerical results suggest $a \in [0.5, 1.5]$ can be better than $a = 2$ since the corresponding losses distinguish points close to boundary better.
 - MIP may not be suitable for large-scale problems.

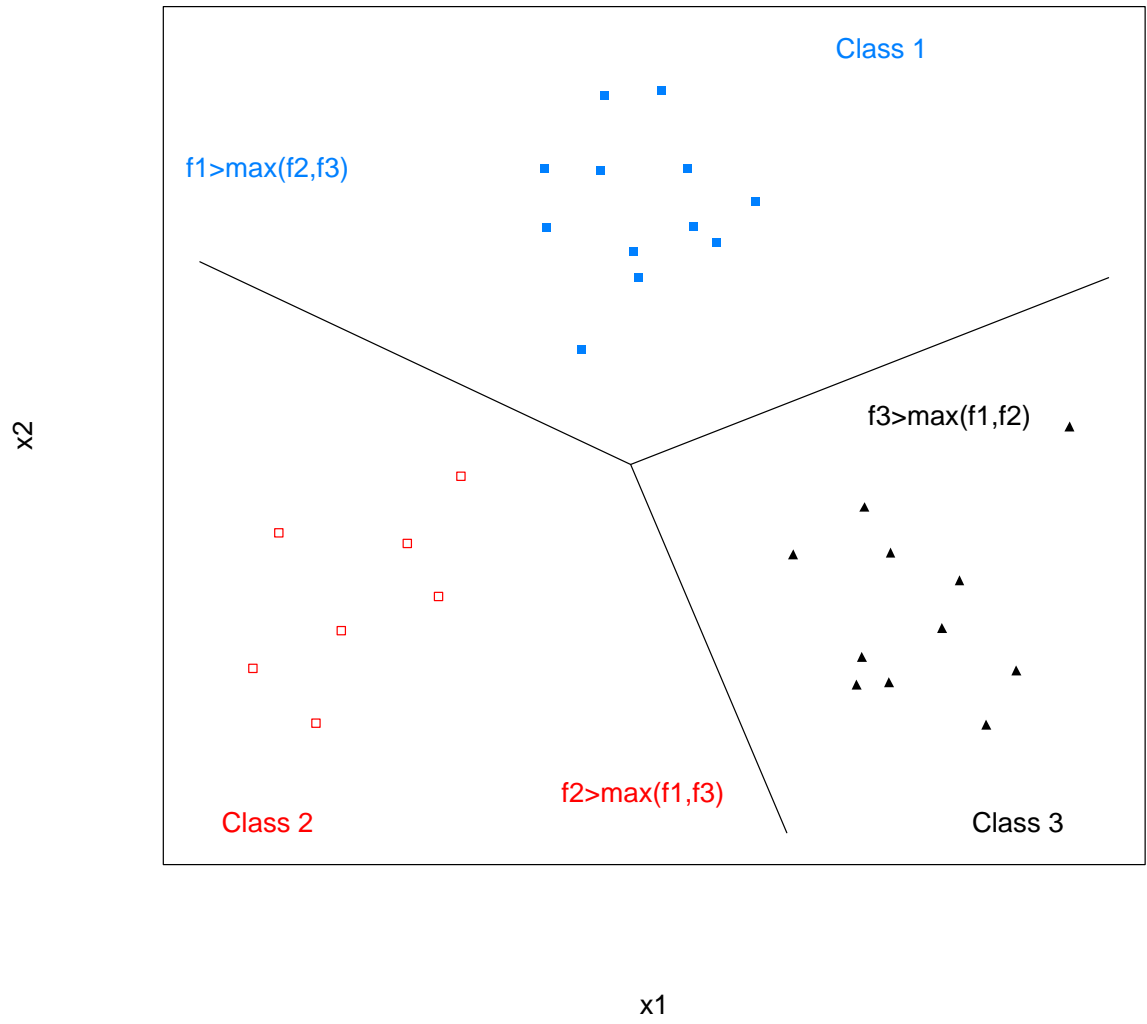
A New Approach

- Truncated hinge loss: $T_s(u) = H_1(u) - H_s(u)$, where $H_s(u) = [s - u]_+$ (also studied by Collobert et al. (2006) for binary problems).
- Choice of s is important (especially for multcategory classification).
- Special cases:
 - The original hinge loss $H_1(u)$ when $s = -\infty$.
 - The ψ loss with $a = 2$ when $s = 0$.



From Binary to Multicategory

- Label: $\{-1, +1\} \rightarrow \{1, 2, \dots, k\}$.
- k -class
 - Construct decision function vector $\mathbf{f} = (f_1, \dots, f_k)$.
($k = 2$ only one f)
 - Classifier: $\operatorname{argmax}_{j=1, \dots, k} f_j(\mathbf{x})$. ($k = 2$: $\operatorname{sign}(f)$)
- Accuracy
Generalization Error (GE): $\operatorname{Err}(\mathbf{f}) = P(Y \neq \operatorname{argmax}_j f_j(\mathbf{X}))$.



Find $\mathbf{f} = (f_1, f_2, f_3)$ and use $\operatorname{argmax}_j f_j(\mathbf{x})$ to do classification.

Multicategory Framework

- **Multiple comparison:** $\mathbf{g}(\mathbf{x}, y) = \{f_y(\mathbf{x}) - f_j(\mathbf{x}), \forall j \neq y\}$. (Liu and Shen, 2006)
 - Compare class y with rest $k - 1$ classes.
 - $\mathbf{g}(\mathbf{x}, y) = f_y(\mathbf{x}) - f_{3-y}(\mathbf{x})$ when $k = 2$.
- \mathbf{f} yields correct classification for (\mathbf{x}, y) if $\mathbf{g}(\mathbf{x}, y) > \mathbf{0}_{k-1}$, i.e., $\min(\mathbf{g}(\mathbf{x}, y)) > 0$.
- **Generalized functional margin:** $\min(\mathbf{g}(\mathbf{x}, y))$; reduced to $yf(\mathbf{x})$ for binary case with $y \in \{-1, +1\}$.

Multicategory Learning

- Regularization: Find \mathbf{f} via minimizing
$$\sum_{j=1}^k J(f_j) + C \sum_{i=1}^n l(\mathbf{f}(\mathbf{x}_i), y_i),$$
 with constraint
$$\sum_{j=1}^k f_j(\mathbf{x}) = 0.$$
- Examples of $l(\mathbf{f}(\mathbf{x}_i), y_i)$
 - Multicategory ψ -learning (Liu and Shen (2006)): $\psi(\min(\mathbf{g}(\mathbf{x}_i, y_i)))$.
 - Multicategory SVM (Crammer and Singer (2001), Liu and Shen (2006)): $H_1(\min(\mathbf{g}(\mathbf{x}_i, y_i)))$.
 - Multicategory SVM with truncation: $H_s(\min(\mathbf{g}(\mathbf{x}_i, y_i)))$.

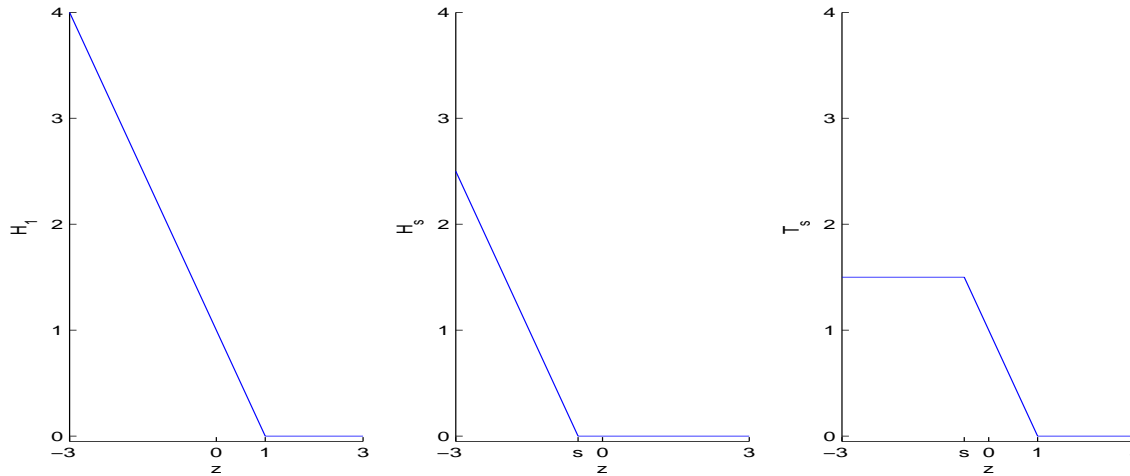
Multicategory SVM

- Other extensions: Weston & Watkins (1998), Vapnik (1998), Lee et al. (2004), etc.
- Loss by Lee et al.:
 - $l(\mathbf{f}(\mathbf{x}_i), y_i) = \sum_{j=1}^k I(y_i \neq j) [f_j(\mathbf{x}_i) + 1]_+$,
 $\sum_{j=1}^k f_j(\mathbf{x}) = 0$.
 - Fisher consistent: estimate $\operatorname{argmax}_j P_j(\mathbf{x})$ asymptotically, where $P_j(\mathbf{x}) = P(Y = j|\mathbf{x})$.

Properties of the Truncated Loss

- Loss $H_s(\min(\mathbf{g}(\mathbf{x}_i, y_i)))$ has Fisher consistency when $0 \geq s \geq -\frac{1}{k-1}$.
- For $s < -\frac{1}{k-1}$:
 - Consistent if $\operatorname{argmax}_j P_j \geq 0.5$.
 - May not guarantee consistency if $\operatorname{argmax}_j P_j < 0.5$.
- Choice of s : $-\frac{1}{k-1}$.

D.C. Algorithm



- 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
 $J(\Theta) = J_{\text{vex}}(\Theta) + J_{\text{cav}}(\Theta)$

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

- The algorithm converges in finite steps (Liu et al. (2005)).
- Choice of initial values: Use SVM's solution.

D.C. Algorithm

The dual problem of the optimization problem at each iteration:

$$\begin{aligned}
 \min \quad & \frac{1}{2} \sum_{m=1}^k \left\| \sum_{i: y_i=m} \sum_{m' \neq y_i} (\alpha_{im'} - \beta_{im'}) \mathbf{x}_i^T - \sum_{i: y_i \neq m} (\alpha_{im} - \beta_{im}) \mathbf{x}_i^T \right\|_2^2 - \sum_{i=1}^n \sum_{m' \neq y_i} \alpha_{i'} \\
 \text{s.t.} \quad & \sum_{i: y_i=m} \sum_{m' \neq y_i} (\alpha_{im'} - \beta_{im'}) - \sum_{i: y_i \neq m} (\alpha_{im} - \beta_{im}) = 0, \quad m = 1, 2, \dots, k \\
 & 0 \leq \sum_{m \neq y_i} \alpha_{im} \leq C, \quad i = 1, 2, \dots, n \\
 & \alpha_{im} \geq 0, \quad i = 1, 2, \dots, n; \quad m \neq y_i,
 \end{aligned}$$

\mathbf{w}_m 's can be recovered by

$$\mathbf{w}_m = \sum_{i: y_i=m} \sum_{m' \neq y_i} (\alpha_{im'} - \beta_{im'}) \mathbf{x}_i - \sum_{i: y_i \neq m} (\alpha_{im} - \beta_{im}) \mathbf{x}_i. \quad (1)$$

The set of SVs is a only a SUBSET of the original one!

Nonlinear learning can be achieved by the kernel trick.

Variable Selection

- Important for getting sparse classifiers and good interpretation.
- Linear learning: replace the L_2 penalty by the L_1 penalty.

$$J^s(\Theta) = \sum_{j=1}^d \sum_{m=1}^k \delta_{jm} |w_{jm}| + C \sum_{i=1}^n H_1(\min \mathbf{g}(\mathbf{x}_i, y_i)) \\ + \sum_{m=1}^k \left\langle \frac{\partial}{\partial \mathbf{w}_m} J_{cav}^s(\Theta_t), \mathbf{w}_m \right\rangle + \sum_{m=1}^k \frac{\partial}{\partial b_m} J_{cav}^s(\Theta_t) b_m$$

- δ_{jm} is the weight for coefficient w_{jm} ;
 $\delta_{jm} = 1$ for standard L_1 penalty.

Variable Selection

- L_1 penalty: shrinkage on both important and unimportant variables.
- Weighted L_1 penalty: try to penalize unimportant variables more.
- Solution: Adaptive L_1 penalty
 - Derive the weights
 - Apply the weighted L_1 penalty
 - References: linear regression (Breiman (1995), Zou (2005)), survival analysis and related settings (Zhang and Lu (2005), Lu and Zhang (2006a), Lu and Zhang (2006b)), binary SVM (Zou (2006)), multiclass SVM (joint work with Y. Wu, H. Zhang, and J. Zhu).
- Procedure for truncated SVM:
 - Solve the coefficients for the truncated SVM with L_2 penalty.
 - Solve the truncated SVM using weighted L_1 penalty with $\delta_{jm} = 1/|w_{jm}|$.

Numerical Examples

- Generate (x_1, x_2) uniformly from the circle $\{(x_1, x_2) : x_1^2 + x_2^2 \leq 1\}$.
- $y = \lfloor \frac{k\vartheta}{2\pi} \rfloor + 1$, where ϑ is the angle between the ray from $(0, 0)$ to $(1, 0)$ and another ray from $(0, 0)$ to (x_1, x_2) .
- Randomly select some points and flip their labels to the remaining $k - 1$ classes with equal probabilities.
- Add independent noise variables generated from $\text{Uniform}[-1, 1]$.
- Sizes of training, tuning and testing data are 100, 100 and 10000.
- Choose C based on the tuning set.

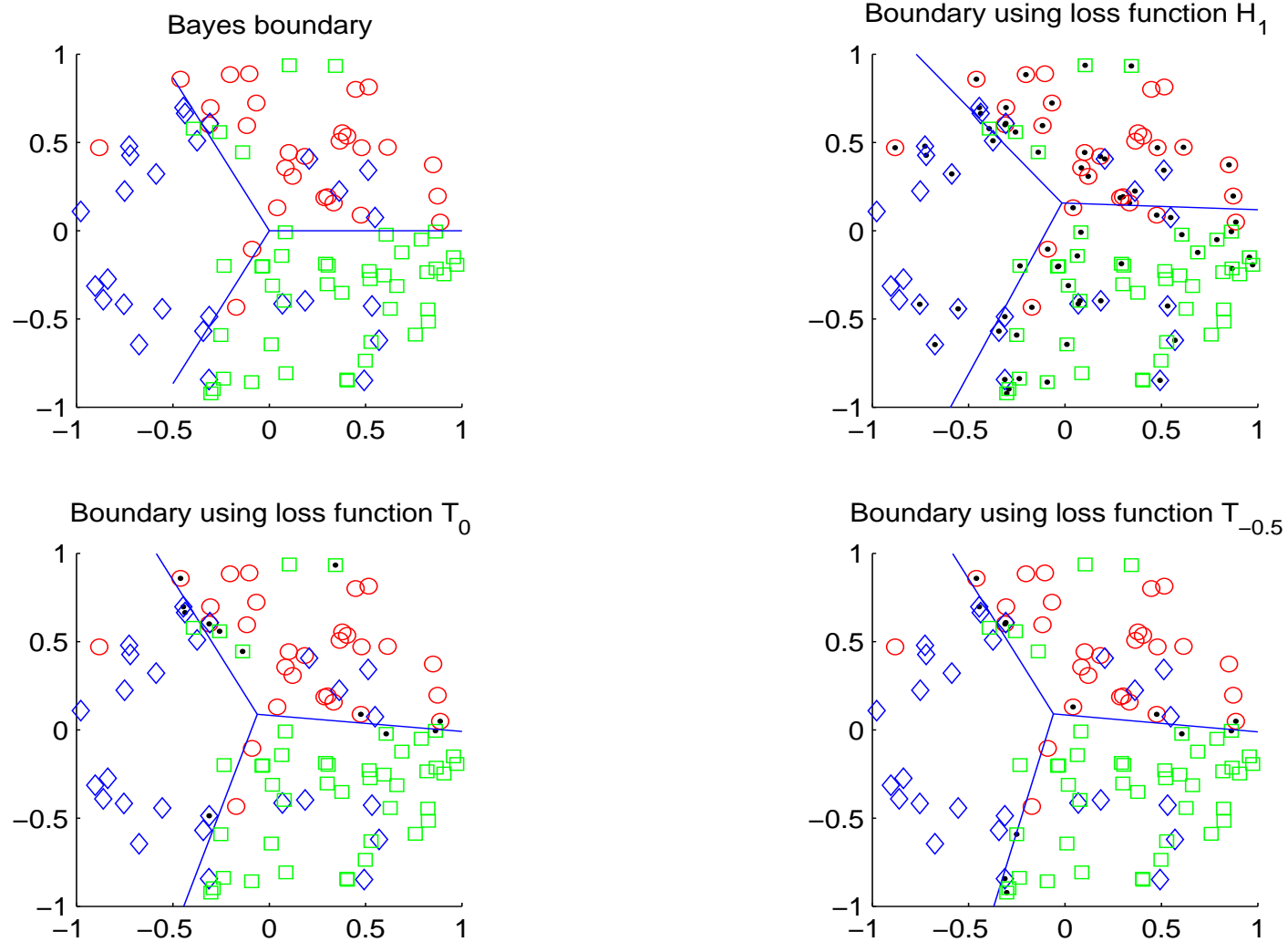


Figure 1: Red circles, blue diamonds, and green squares denote observations in classes 1, 2, and 3 respectively. Observations with black dots are SVs.

Table 1: Linear example with 10% flipping

k	d	Method	L_2		L_1		Adaptive L_1	
			Testing Error	# of SV's	Testing Error	Avg # of zeros	Testing Error	Avg # of zeros
4	2	H_1 (SVM)	0.1737 (0.0189)	64.36	0.1773 (0.0213)	0.00	0.1817 (0.0223)	0.00
		$T_0(\psi)$	0.1520 (0.0182)	20.88	0.1526 (0.0192)	0.00	0.1563 (0.0225)	0.00
		$T_{-1/3}$	0.1405 (0.0154)	17.96	0.1404 (0.0161)	0.00	0.1424 (0.0172)	0.00
	6	H_1 (SVM)	0.2354 (0.0223)	63.26	0.2271 (0.0227)	0.00	0.2002 (0.0249)	0.00
		$T_0(\psi)$	0.2271 (0.0291)	29.32	0.2141 (0.0234)	1.02	0.1782 (0.0245)	8.28
		$T_{-1/3}$	0.2186 (0.0210)	32.12	0.2054 (0.0234)	1.02	0.1702 (0.0231)	7.84

Table 2: Linear example with 20% flipping

k	p	Method	L_2		L_1		Adaptive L_1	
			Testing Error	# of SV's	Testing Error	Avg # of zeros	Testing Error	Avg # of zeros
3	2	H_1 (SVM)	0.2588 (0.0312)	75.16	0.2603 (0.0317)	0.00	0.2590 (0.0303)	0.00
		$T_0(\psi)$	0.2381 (0.0179)	21.96	0.2359 (0.0155)	0.06	0.2377 (0.0186)	0.62
		$T_{-0.5}$	0.2272 (0.0142)	23.76	0.2287 (0.0177)	0.04	0.2261 (0.0151)	0.54
	6	H_1 (SVM)	0.3012 (0.0207)	73.32	0.2866 (0.0264)	0.00	0.2695 (0.0249)	0.00
		$T_0(\psi)$	0.2883 (0.0236)	28.12	0.2672 (0.0211)	0.74	0.2527 (0.0212)	5.56
		$T_{-0.5}$	0.2776 (0.0221)	33.18	0.2614 (0.0205)	0.74	0.2434 (0.0137)	4.44
4	2	H_1 (SVM)	0.3039 (0.0404)	83.50	0.3063 (0.0433)	0.00	0.3073 (0.0403)	0.00
		$T_0(\psi)$	0.2611 (0.0312)	25.32	0.2609 (0.0319)	0.04	0.2630 (0.0322)	0.06
		$T_{-1/3}$	0.2480 (0.0288)	22.18	0.2523 (0.0318)	0.04	0.2536 (0.0327)	0.04
	6	H_1 (SVM)	0.3553 (0.0286)	79.74	0.3514 (0.0268)	0.00	0.3357 (0.0359)	0.00
		$T_0(\psi)$	0.3375 (0.0355)	29.40	0.3213 (0.0338)	1.20	0.2980 (0.0438)	6.76
		$T_{-1/3}$	0.3300 (0.0339)	32.02	0.3153 (0.0341)	0.78	0.2858 (0.0331)	6.10

Table 3: Nonlinear example using Gaussian kernel: $k = 3, p = 2, \sigma = 1$

Flipping percentage	Method	Testing Error	# of SV's
10%	H_1 (SVM)	0.1840 (0.0459)	57.15
	T_0 (ψ)	0.1695 (0.0355)	21.00
	$T_{-0.5}$	0.1688 (0.0433)	33.55
20%	H_1 (SVM)	0.2755 (0.0183)	67.60
	T_0 (ψ)	0.2657 (0.0158)	18.45
	$T_{-0.5}$	0.2615 (0.0217)	27.15

Summary

- Propose a new SVM with the truncated hinge loss.
- Explore its consistency and variable selection.
- Use D.C. algorithm to optimize the nonconvex minimization problem via sequential convex subproblems.
- Numerical examples suggest its superiority over the SVM and ψ -learning.

On-going and future research

- More numerical comparisons.
- Further theoretical investigation.
- More efficient algorithms like solution path.
- Better tuning methods: covariance penalty, data perturbation, etc.
- Apply other penalty functions.
- Truncation of other loss functions.