Adaptively Weighted Large Margin Classifiers

Yichao Wu

North Carolina State Univesity

Joint work with Yufeng Liu, UNC

Outline

- Background
- Large-Margin Classifiers in Regularization Framework
- Robust Classifiers via Loss Truncation
- D. C. Algorithm for Nonconvex Optimization
- Robust Adaptive Weighted Learning: One-step and iterative weighting
- Connection between RSVM and WSVM
- Numerical Examples

General Framework

- Supervised learning: Given training data $\{(\boldsymbol{x}_i, y_i)_{i=1}^n\}$, i.i.d. $\sim$ unknown $P(\boldsymbol{x}, y)$.
  - input $\boldsymbol{x}_i \in R^d$ 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 $\boldsymbol{x}$ (Good Generalization).

Classification Methods

- Traditional statistical methods
  Linear/Quadratic Discriminant Analysis, Nearest Neighbor, Logistic Regression, etc.

- Machine learning
  Margins $\rightarrow$ SVM (Boser et al., 1992, Vapnik, 1995),
  Boosting (Freund & Schapire, 1997),
  ψ-Learning (Shen et al., 2003, Liu & Shen, 2006),
  Distance Weighted Discrimination (Marron et al. 2007), etc.

Binary Large-Margin Classifiers & Regularization

- $y \in \{\pm 1\}$;
  Estimate $f(\boldsymbol{x})$ with classification rule sign$[f(\boldsymbol{x})] : R^d \to \{\pm 1\}$,
  $\hat{y} = +1$ if $f(\boldsymbol{x}) > 0$ and $\hat{y} = -1$ if $f(\boldsymbol{x}) < 0$.

- Regularization framework

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

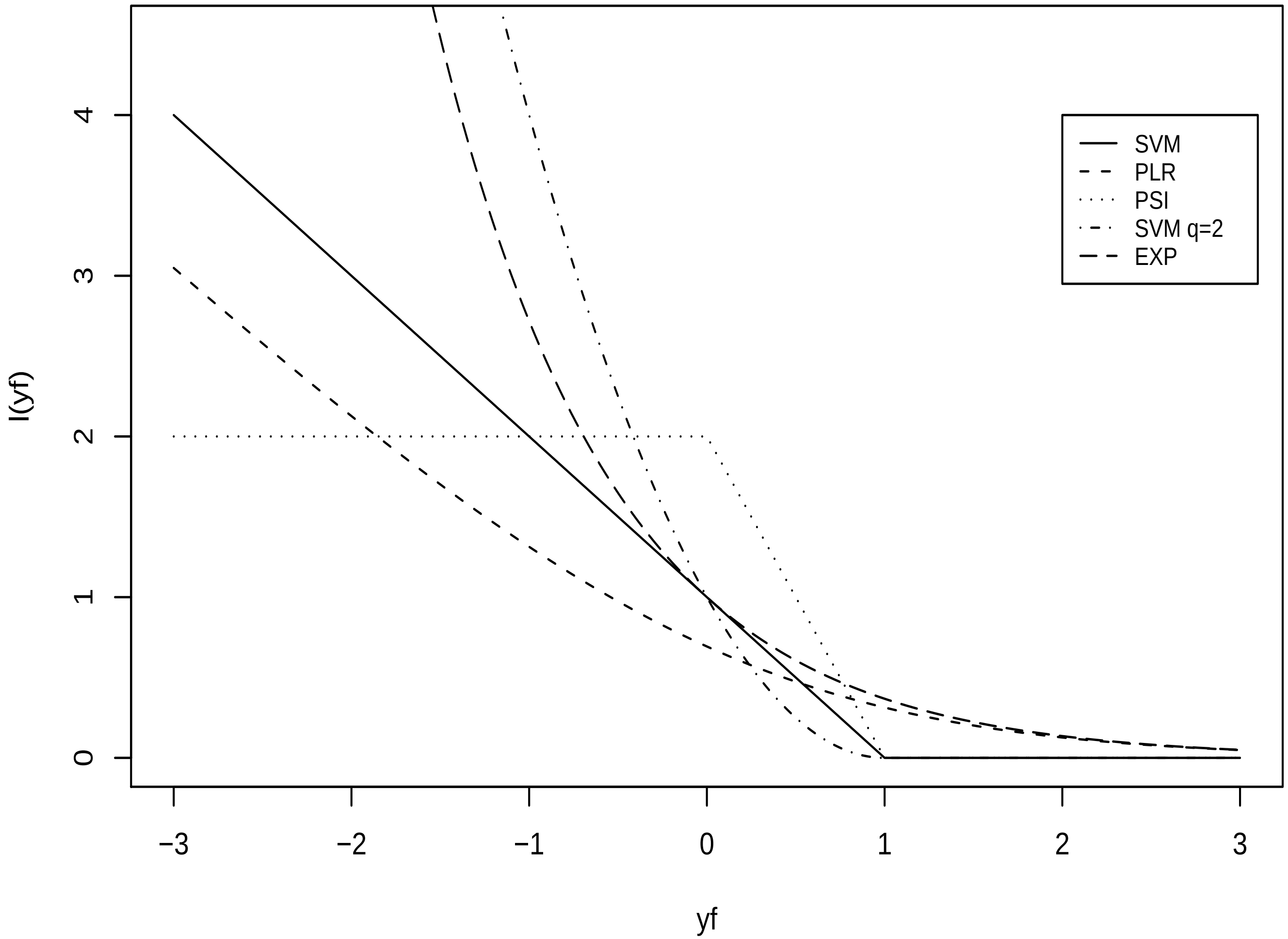
- Regularization term $J(f)$: roughness penalty of f;
  Loss l: data fit measure.

- Consider $f(\boldsymbol{x}) = \boldsymbol{w}^{'}\boldsymbol{x} + b$ (Nonlinear learning via basis expansion or kernel learning).

Large-Margin Loss Functions

- The loss $l(u)$ is typically non-increasing in u.
  - $u = y_i f(\boldsymbol{x}_i)$: functional margin.
  - Correction classification if $y_i f(\boldsymbol{x}_i) > 0$.
- The SVM:
  - The Hinge Loss: $l(f(\boldsymbol{x}_i), y_i) = [1 - y_i f(\boldsymbol{x}_i)]_+$ (Vapnik, 1998; Wahba, 1998).
  - The minimizer is $f^*(\boldsymbol{x}) = \text{sign}(p(\boldsymbol{x}) - 1/2)$; $p(\boldsymbol{x}) = P(Y = 1|\boldsymbol{X} = \boldsymbol{x})$ (Lin, 2002).

Different Losses



Outlier Effects & Robust Learning

- Unbounded loss l (e.g. The hinge loss): large loss for outliers; sensitive to outliers.
- Robust Learning: Reduce the loss for outliers
  - Loss Truncation. (Wu and Liu, JASA, 2007; Park and Liu, CJS, 2011)
  - Adaptive Weighting.

Optimization Problem for SVM

To get $(\boldsymbol{w}, b)$ for the optimal hyperplane, solve:

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

subject to

$$y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geq (1 - \xi_i), \quad \xi_i \geq 0; \quad i = 1, \ldots, n,$$

where $C > 0$ is a tuning parameter.

The dual problem for SVM

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

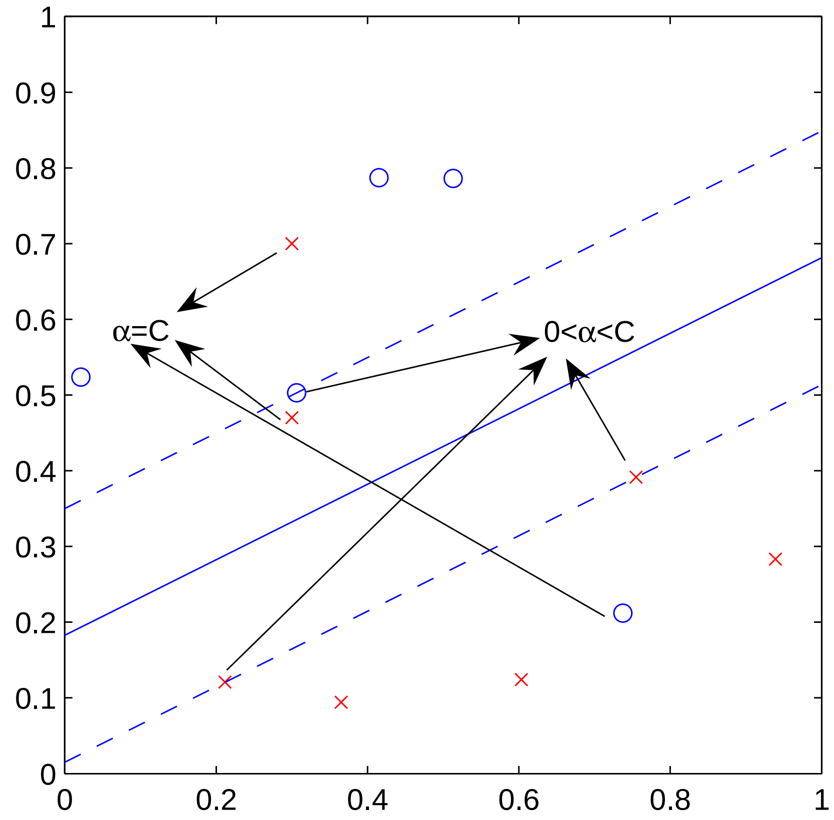
subject to: $$0 \leq \alpha_i \leq C, \quad i = 1, 2, \cdots, n$$

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

- Can be solved by quadratic programming.
- Recover $\boldsymbol{w}$: $\boldsymbol{w} = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i$; For given $\boldsymbol{w}$, b can be solved using KKT conditions or Linear Programming (LP).
- Kernel Trick: Replace $\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$ by $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $f(\boldsymbol{x}) = \sum_{i=1}^{n} y_i \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b$.

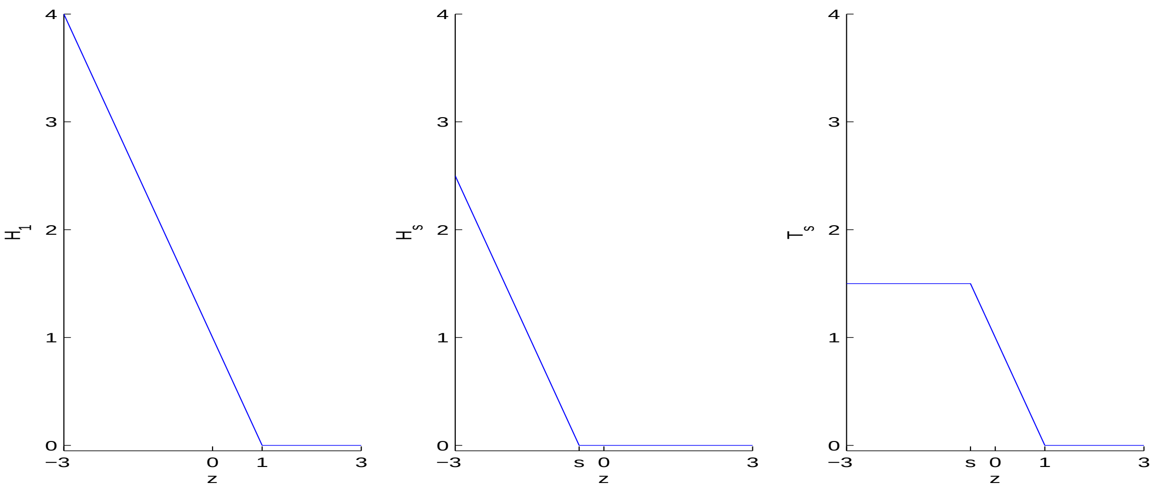
Support Vectors

- $\alpha_i = 0 \rightarrow y_i f(\boldsymbol{x}_i) > 1$; not needed in constructing $f(\boldsymbol{x})$.
  Support vectors:
- $0 < \alpha_i < C \rightarrow y_i f(\boldsymbol{x}_i) = 1$ (Solve b).
- $\alpha_i = C \rightarrow y_i f(\boldsymbol{x}_i) < 1$.
- Outliers will be SVs!

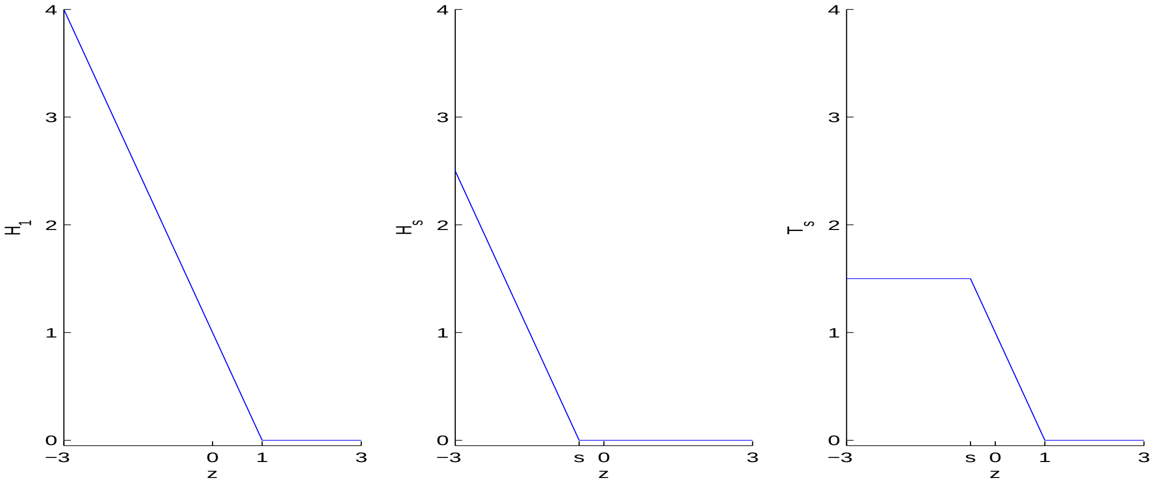


Truncation of unbounded losses

- Work for general unbounded losses.
- Truncated hinge loss: $T_s(u) = H_1(u) - H_s(u)$; $H_s(u) = [s - u]_+$; similar for the logistic loss.
- Choice of s is important (especially for multicategory classification).



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

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., JCGS, 2005).
- Choice of initial values: Use the original classifiers without truncation.
- The set of SVs is a only a SUBSET of the original one!

Weighted Learning

- Assign weights $v_1, \ldots, v_n$ for the n training points.
- Bigger (smaller) weights for points close to (far from) the boundary.
- Outliers far from the boundary receive smaller weights.
- Robustness can be achieved with properly chosen weights.

Optimization Problem for Weighted SVM

To get $(\boldsymbol{w}, b)$ for the optimal hyperplane, solve:

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

subject to

$$y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geq (1 - \xi_i), \quad \xi_i \geq 0; \quad i = 1, \ldots, n,$$

where $C > 0$ is a tuning parameter and $v_i > 0$ is the weight for the i-th point.

The dual problem for Weighted SVM

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

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

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

Choice of Weights

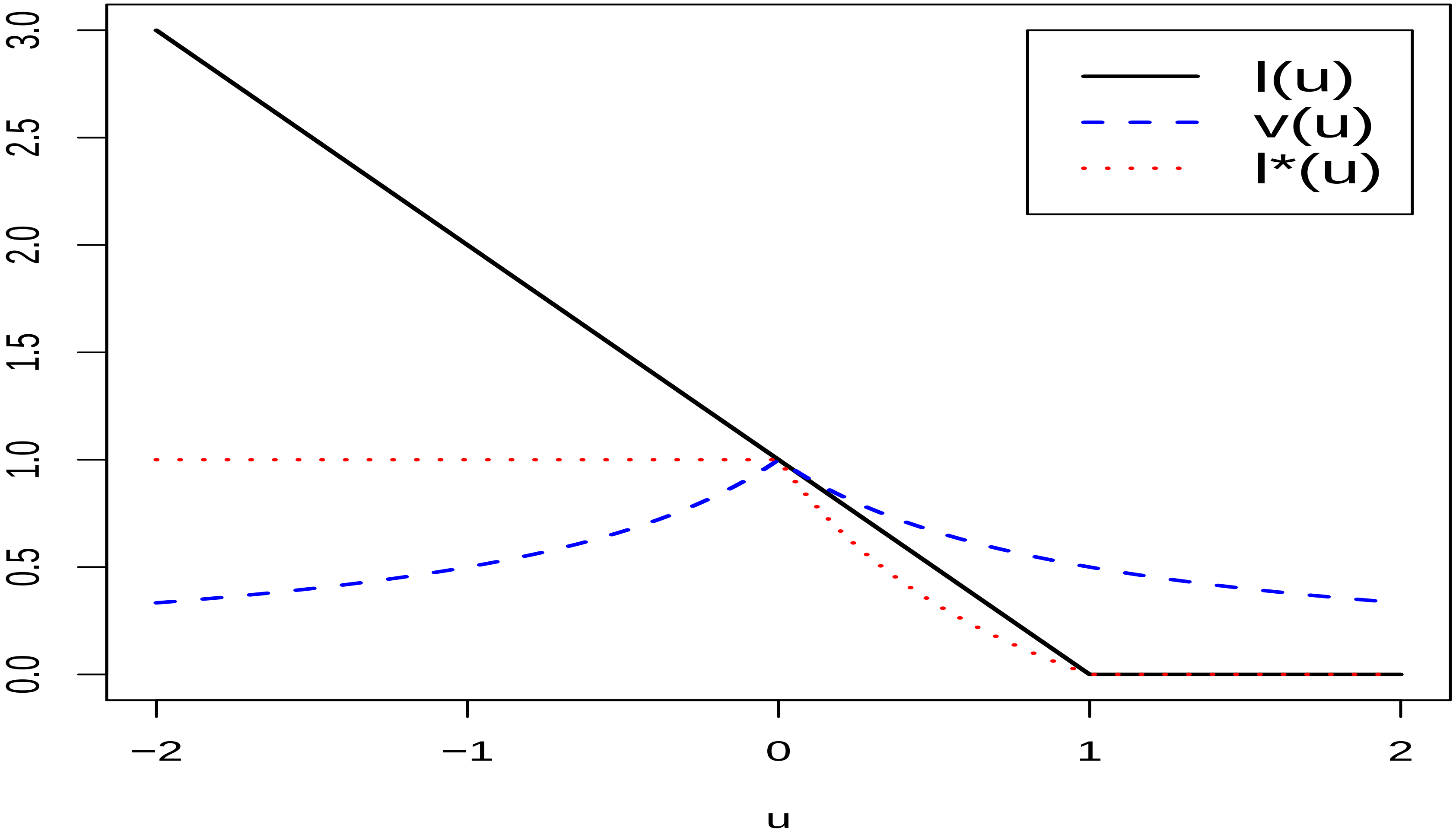
- Loss $l(u)$ is non-increasing.
- Define the weight function $v(\cdot)$

$$v(u) = \begin{cases} v_-(u) = 1/l(u) & \text{if } u \leq 0 \\ v_+(u) = v_-(-u) & \text{if } u > 0 \end{cases}$$

- New loss: $l^*(u) = v(u)l(u)$

$$l^*(u) = \begin{cases} 1 & \text{if } u \leq 0 \\ l(u)/l(-u) & \text{if } u > 0 \end{cases}$$

- $l^*(u)$ is close to the 0-1 loss and enjoys Fisher consistency.


l(u)
v(u)
l*(u)
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−2
−1
0
1
2
u

Adaptive Weighting: One-step weighting (OWSVM)

- Do not solve the nonconvex loss $l^*(u)$ directly.
- Apply the idea of adaptive weighting

Steps:

- Implement standard learning with equal weights and obtain $f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n)$.
- Apply one-step weighted learning with weights $v_i = v(y_i f(\boldsymbol{x}_i))$.

Adaptive Weighting: Iterative weighting (IWSVM)

Step 1 (Initialization): Solve standard learning with equal weights.

Step 2 (Iteration): At tth iteration, set $v_i^{(t)} = 1$ if $s \le y_i \hat{f}^{(t-1)}(\boldsymbol{x}_i) \le 1$ and 0 otherwise for $i = 1, 2, \cdots, n$. Solve WSVM with weights $v_i^{(t)}$'s to get $\hat{f}^{(t)}(\cdot)$.

Step 3 (Convergence): Iterate until convergence.

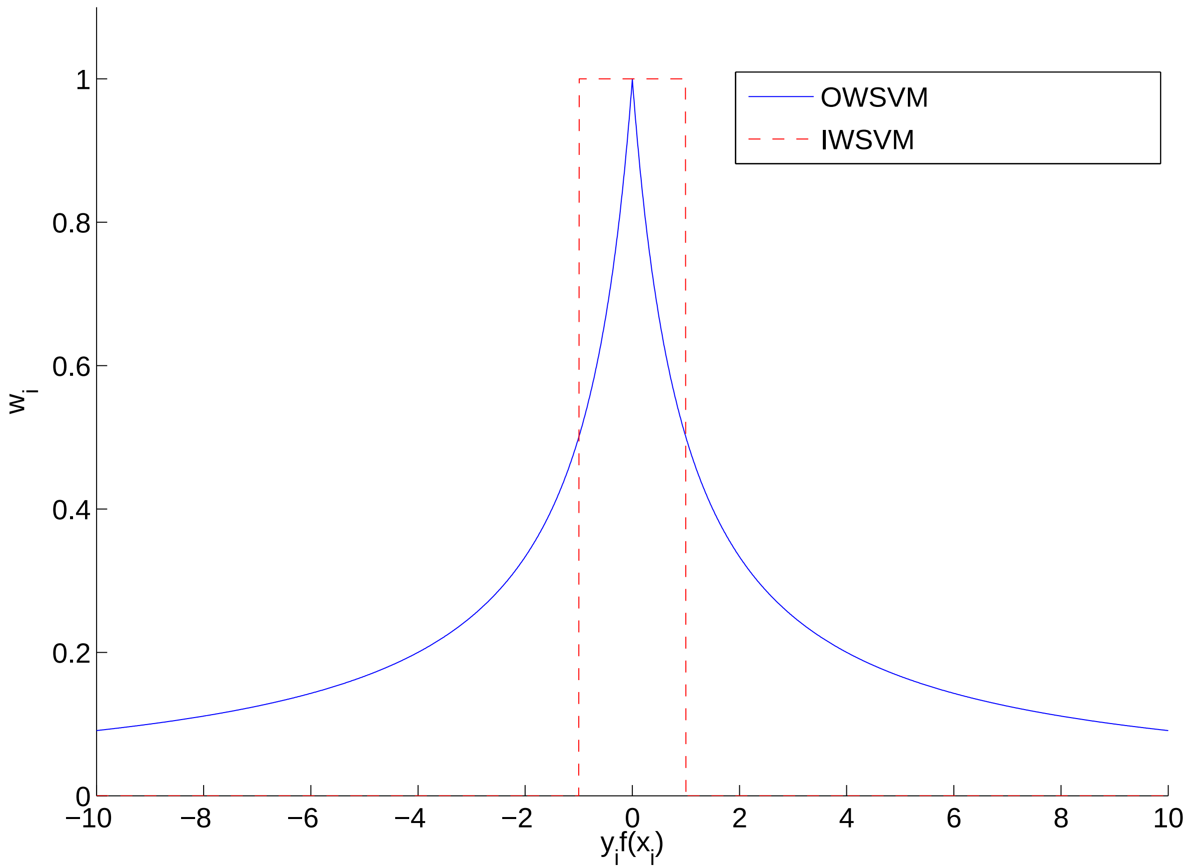


Figure 1: Plot of the weight functions for the OWSVM and IWSVM

Connection between DCA and IWSVM

- IWSVM can be shown to yield a local minimizer of RSVM.
- Both DCA and IWSVM solve the RSVM.
- **Theorem:** The DCA and IWSVM algorithms are equivalent in terms of fixed points. Namely, the local solution of the DCA is a fixed point of the proposed IWSVM algorithm, and vice versa.

From Binary to Multicategory

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

Multicategory Framework

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

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.

- Sizes of training, tuning and testing data are 100, 100 and 10000.

- Choose C based on the tuning set.

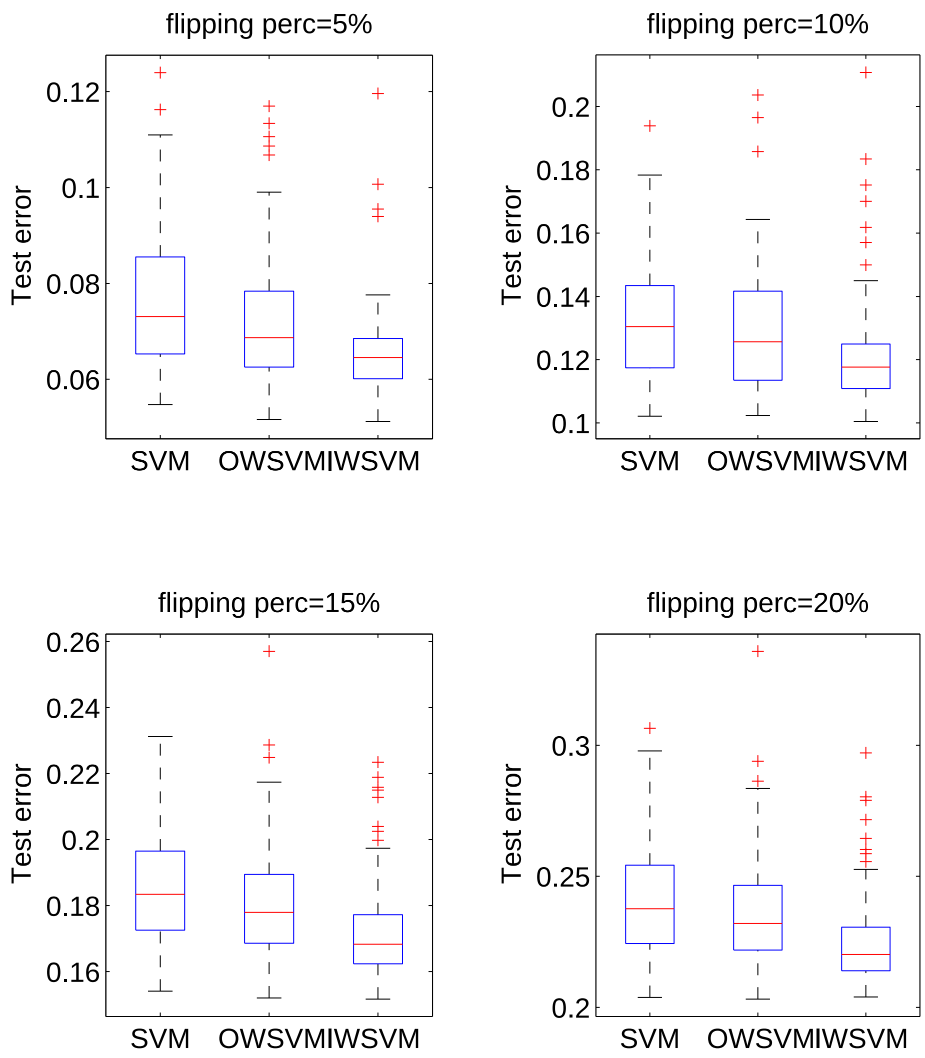


Figure 2: Classification accuracy comparison for $K = 2$.

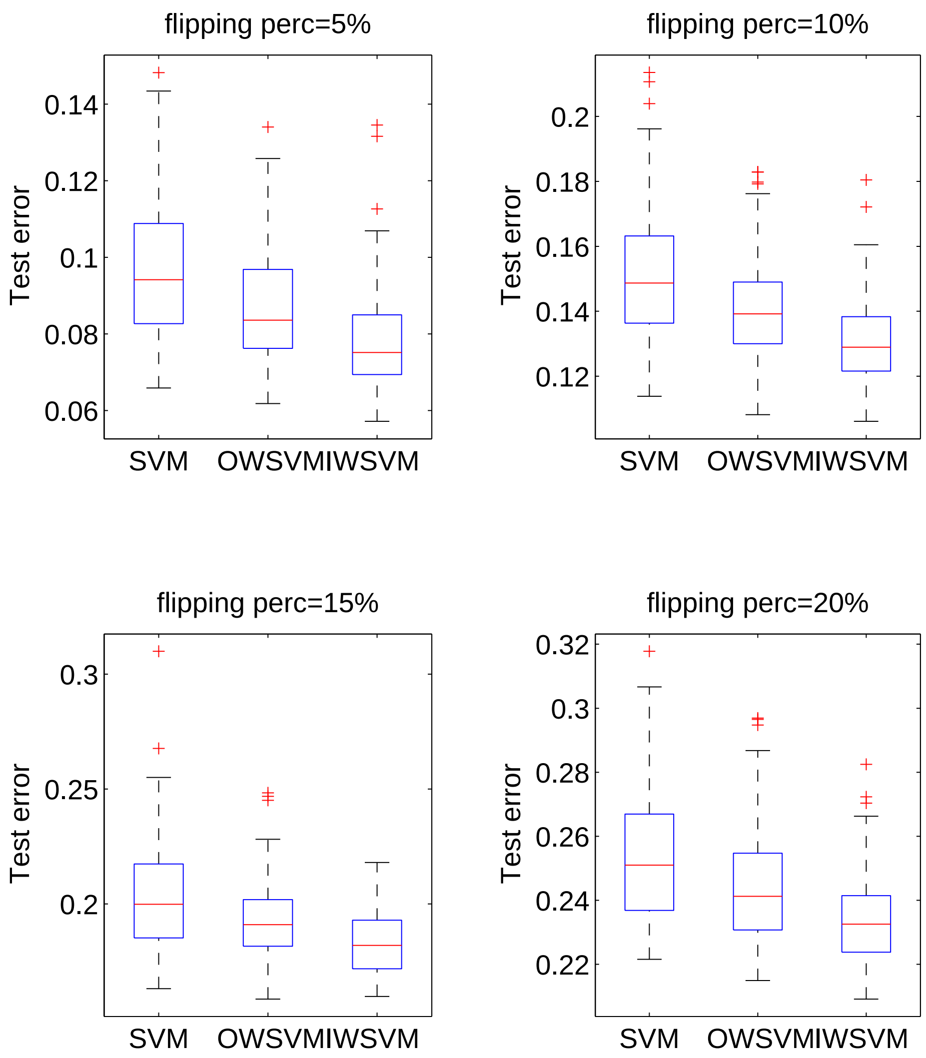


Figure 3: Classification accuracy comparison for $K = 3$.

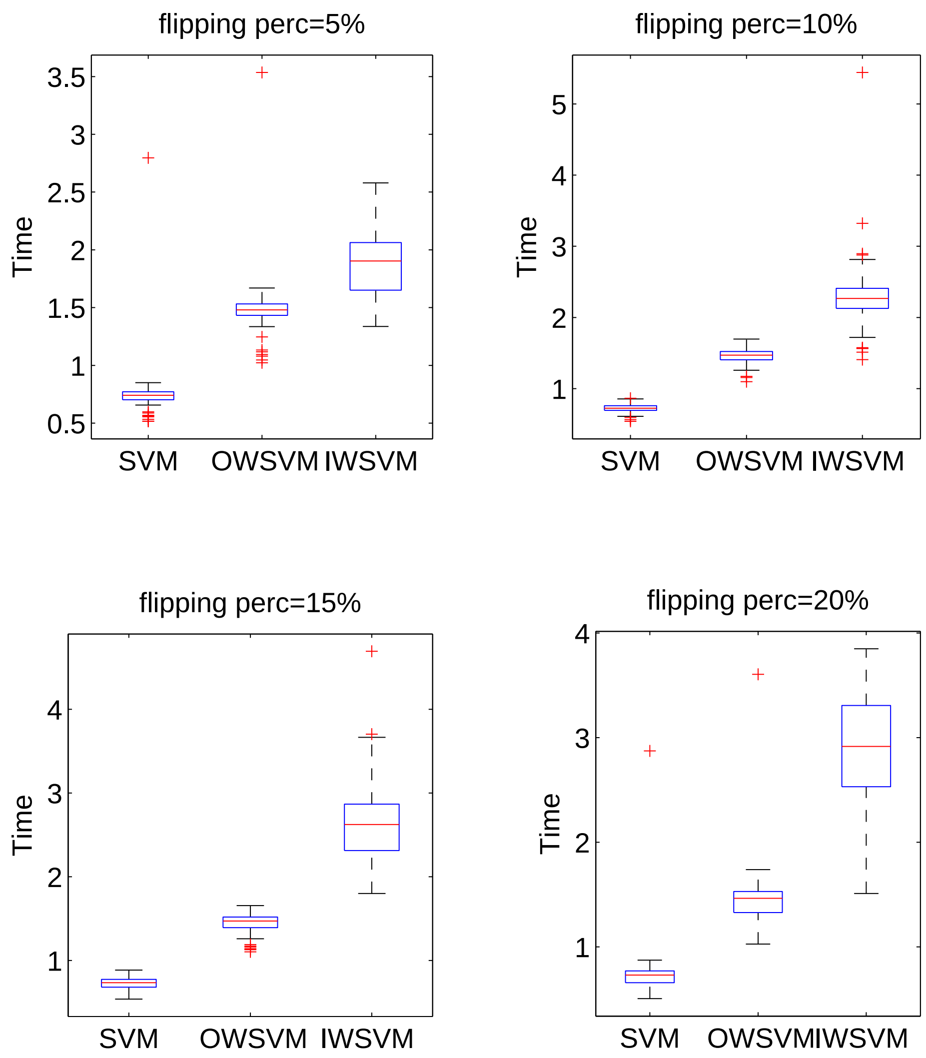


Figure 4: Computational time comparison for $K = 2$.

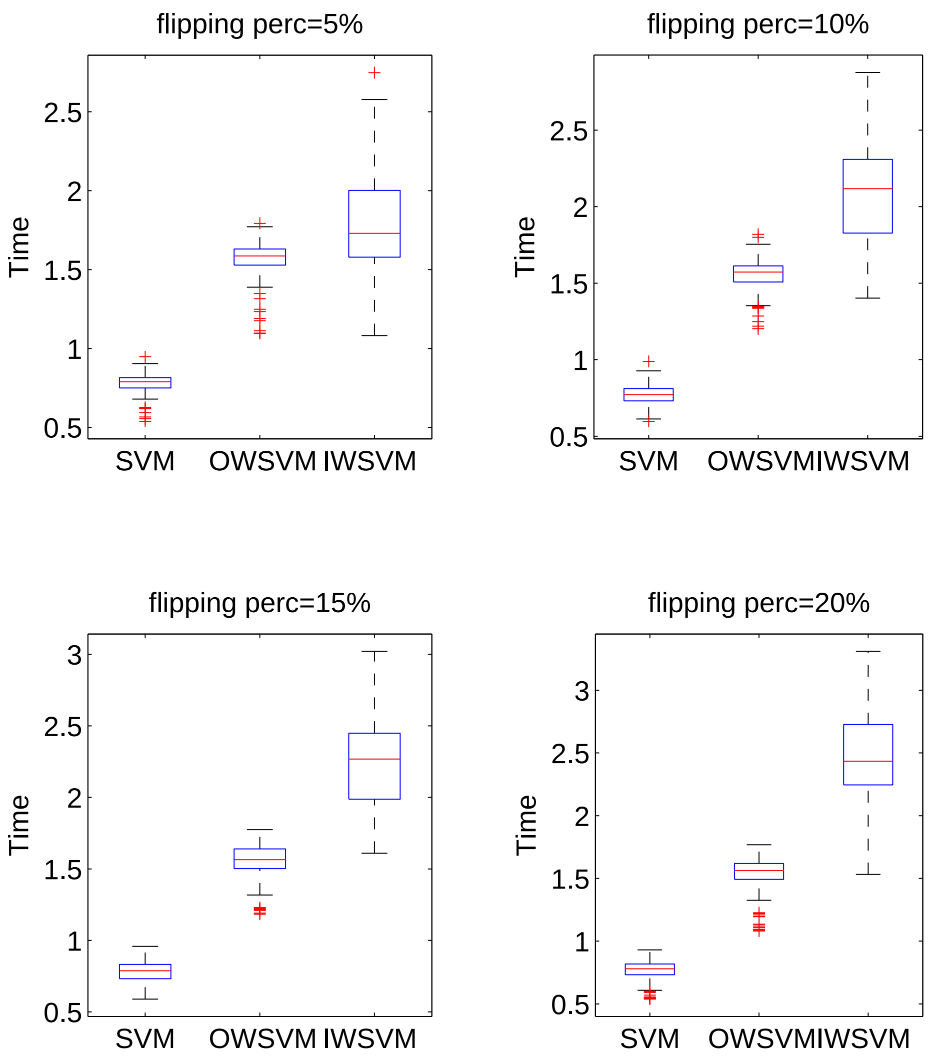


Figure 5: Computational time comparison for $K = 3$.

Wisconsin Breast Cancer Data

- Goal: use a digitized image of a fine needle aspirate of a breast mass to diagnose the corresponding breast cancer status.
- Binary classification: response of diagnosis as either malignant or benign.
- $d = 30$ and $n = 569$; training, tuning, and test sets of sizes 150, 150, and 269.
- Three different levels of flipping 0%, 5%, and 10%.
- Report the average testing error over the test set across 100 random repetitions.

	0%	5%	10%
SVM	0.0382(0.0013)	0.0438(0.0014)	0.0521(0.0017)
OWSVM	0.0377(0.0013)	0.0401(0.0016)	0.0444(0.0018)
IWSVM	0.0381(0.0014)	0.0377(0.0015)	0.0413(0.0016)

Table 1: Classification accuracy of the SVM, OWSVM, and IWSVM on the WDBC data.

Summary

- Propose robust large-margin classifiers: truncated loss functions & Adaptive weighting.
- Weighted learning: one-step weighting and iterative weighting.
- Equivalence between DCA and IWSVM.
- Numerical examples.