Adaptively Weighted Large Margin Classifiers

Yichao Wu
North Carolina State University

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 \( \{(x_i, y_i)_{i=1}^{n}\} \), i.i.d. \( \sim \) unknown \( P(x, y) \).
  - input \( x_i \in \mathbb{R}^d \) 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 \( x \) (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),
  $\psi$-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 $f(x)$ with classification rule $\text{sign}[f(x)] : \mathbb{R}^d \rightarrow \{\pm 1\}$,
  $\hat{y} = +1$ if $f(x) > 0$ and $\hat{y} = -1$ if $f(x) < 0$.

- Regularization framework

$$
\min_{f} \ J(f) + C \sum_{i=1}^{n} l(f(x_i), y_i).
$$

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

- Consider $f(x) = w'x + b$ (Nonlinear learning via basis expansion or kernel learning).
Large-Margin Loss Functions

- The loss $l(u)$ is typically non-increasing in $u$.
  - $u = y_if(x_i)$: functional margin.
  - Correction classification if $y_if(x_i) > 0$.

- The SVM:
  - The Hinge Loss: $l(f(x_i), y_i) = [1 - y_if(x_i)]_+$ (Vapnik, 1998; Wahba, 1998).
  - The minimizer is $f^*(x) = \text{sign}(p(x) - 1/2)$;
    \[ p(x) = P(Y = 1|X = x) \] (Lin, 2002).
Different Losses

![Graph showing different loss functions: SVM, PLR, PSI, SVM q=2, and EXP. The x-axis represents yf, and the y-axis represents l(yf).]
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 \((\mathbf{w}, b)\) for the optimal hyperplane, solve:

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

subject to

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

where \(C > 0\) is a tuning parameter.
The dual problem for SVM

\[ \min L_D(\alpha) = \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle - \sum_{i=1}^{n} \alpha_i \]

subject to:  

\[ 0 \leq \alpha_i \leq C, \quad i = 1, 2, \ldots, n \]

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

- Can be solved by quadratic programming.

- Recover \( \mathbf{w} \): \( \mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \); For given \( \mathbf{w} \), \( b \) can be solved using KKT conditions or Linear Programming (LP).

- Kernel Trick: Replace \( \langle x_i, x_j \rangle \) by \( K(x_i, x_j) \) and \( f(x) = \sum_{i=1}^{n} y_i \alpha_i K(x_i, x) + b \).
Support Vectors

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

Support vectors:

- $0 < \alpha_i < C \rightarrow y_i f(x_i) = 1$ (Solve $b$).
- $\alpha_i = C \rightarrow y_i f(x_i) < 1$.
- 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 multiclass classification).
D.C. Algorithm

- $T_s(u) = H_1(u) - H_s(u)$. 

![Graphs of D.C. functions](image-url)
D.C. Algorithm: The Difference Convex Algorithm for minimizing
\[ J(\Theta) = J_{vex}(\Theta) + J_{cav}(\Theta) \]
1. Initialize \( \Theta_0 \).
2. Repeat \( \Theta_{t+1} = \text{argmin}_\Theta (J_{vex}(\Theta) + \langle J'_{cav}(\Theta_t), \Theta - \Theta_t \rangle) \)
   until convergence of \( \Theta_t \).

- 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 \((w, b)\) for the optimal hyperplane, solve:

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

subject to

\[
y_i(\langle w, x_i \rangle + b) \geq (1 - \xi_i), \quad \xi_i \geq 0; \quad i = 1, \ldots, n,
\]

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

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

subject to:

\[
0 \leq \alpha_i \leq C v_i, \quad i = 1, 2, \ldots, n
\]

\[
\sum_{i=1}^{n} \alpha_i y_i = 0
\]
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.
The graph shows three functions $l(u)$, $v(u)$, and $l^*(u)$ plotted against $u$. The functions are defined over the range $-2$ to $2$ on the $u$-axis and $0$ to $3$ on the $y$-axis. The line $l(u)$ is a straight line decreasing from left to right, $v(u)$ is a smooth curve starting from a higher value and decreasing, and $l^*(u)$ is a dotted line that starts high and decreases to zero at $u = 0$. The functions $l(u)$ and $v(u)$ reach their highest values near $u = -1$, while $l^*(u)$ starts high but decreases sharply near $u = 0$. The legend on the right identifies the functions with their respective lines and styles.
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 
  \( f(x_1), \ldots, f(x_n) \).

• Apply one-step weighted learning with weights \( v_i = v(y_i f(x_i)) \).
Adaptive Weighting: Iterative weighting (IWSVM)

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

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

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 \( \mathbf{f} = (f_1, \ldots, f_k) \).
    
    \( (k = 2 \text{ only one } f) \)
  
  - Classifier: \( \text{argmax}_{j=1,\ldots,k} f_j(\mathbf{x}) \). (\( k = 2 : \text{sign}(f) \))

- Accuracy
  
  Generalization Error (GE): \( \text{Err}(\mathbf{f}) = P(Y \neq \text{argmax}_j f_j(\mathbf{X})) \).
Multicategory Framework

• Multiple comparison: \( g(x, y) = \{ f_y(x) - f_j(x), \forall j \neq y \} \). (Liu and Shen, JASA, 2006)
  
  - Compare class \( y \) with rest \( k - 1 \) classes.
  - \( g(x, y) = f_y(x) - f_{3-y}(x) \) when \( k = 2 \).

• \( f \) yields correct classification for \( (x, y) \) if \( g(x, y) > 0_{k-1} \), i.e., \( \min(g(x, y)) > 0 \).

• Generalized functional margin: \( \min(g(x, y)) \); reduces to \( yf(x) \) 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 \(\{(x_1, x_2) : x_1^2 + x_2^2 \leq 1\}\).

• \(y = \left\lfloor \frac{k \vartheta}{2\pi} \right\rfloor + 1\), where \(\vartheta\) 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 $K = 2$. 
<table>
<thead>
<tr>
<th>Flipping Perc</th>
<th>Test Error SVM</th>
<th>Test Error OWSVM</th>
<th>Test Error IWSVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>5%</td>
<td>0.06</td>
<td>0.08</td>
<td>0.10</td>
</tr>
<tr>
<td>10%</td>
<td>0.12</td>
<td>0.14</td>
<td>0.16</td>
</tr>
<tr>
<td>15%</td>
<td>0.22</td>
<td>0.24</td>
<td>0.26</td>
</tr>
<tr>
<td>20%</td>
<td>0.28</td>
<td>0.30</td>
<td>0.32</td>
</tr>
</tbody>
</table>

Figure 3: Classification accuracy comparison for $K = 3$. 
Figure 4: Computational time comparison for $K = 2$. 
Figure 5: Computational time comparison for $K = 3$. 

![Box plots comparing computational time for different flipping percentages with $K = 3$.]
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.
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.