Multicategory Vertex Discriminant Analysis for High-Dimensional Data

Tong Tong Wu

Department of Epidemiology and Biostatistics
University of Maryland, College Park

October 28, 2010

Joint work with Prof. Kenneth Lange at UCLA, to appear in AOAS
Outline

1. Introduction

2. Vertex Discriminant Analysis (VDA)
   - Category Indicator: Equidistant Points in $\mathbb{R}^{k-1}$
   - $\epsilon$-insensitive Loss Function for VDA
   - Cyclic Coordinate Descent for VDA$_L$
   - Euclidean Penalty for Grouped Effects (VDA$_E$, VDA$_{LE}$)

3. Fisher Consistency of $\epsilon$-Insensitive Loss

4. Numerical Examples

5. Discussion
Introduction
Overview of Vertex Discriminant Analysis (VDA)

- A new method of supervised learning
- Linear discrimination among the vertices
- Each vertex of a regular simplex in Euclidean space representing a different category
- Minimization of $\epsilon$-insensitive residuals and penalties on the coefficients of the linear predictors
- Classification and variable selection performed simultaneously
- Minimization by a primal MM algorithm or a coordinate descent algorithm
- Fisher consistency of $\epsilon$-insensitive
- Statistical accuracy and computational speed
Motivation

Cancer subtype classification

- sheer scale of cancer data sets
- prevalence of multicategory problems
- excess of predictors over cases
- exceptional speed and memory capacity of modern computers
Review of Discriminant Analysis

- **Purpose**: categorize objects based on a fixed number of observed features $x \in \mathbb{R}^p$

- **Observations**: category membership indicator $y$ and feature vector $x \in \mathbb{R}^p$

- **Discriminant rule**: divide $\mathbb{R}^p$ into disjoint regions corresponding to different categories

- **Supervised learning**
  - Begin with a set of fully categorized cases (training data)
  - Build discriminant rules using training data

- **Given a loss function** $L(y, x)$, minimize
  - Expected loss $E \left[ L(Y, X) \right] = E \{ E \left[ L(Y, X)|X \right] \}$
  - Average conditional loss $n^{-1} \sum_{i=1}^{n} L(y_i, x_i)$ with a penalty term
Multicategory Problems in SVM

- Solving a series of binary problems
  - One-versus-rest (OVR): \( k \) binary classifications, but poor performance when no dominating class exists (Lee at al. 2004)
  - Pairwise comparisons: \( \binom{k}{2} \) comparisons, a violation of the criterion of parsimony (Kressel 1999)

- Considering all classes simultaneously (Bredensteiner and Bennett 1999; Crammer and Singer 2001; Guermeur 2002; Lee et al. 2004; Liu et al. 2006, 2005, 2006; Liu 2007; Vapnik 1998; Weston and Watkins 1999; Zhang 2004b; Zou et al. 2006)

- Closely related work: L1MSVM (Wang and Shen 2007) and L2MSVM (Lee et al. 2004)
Vertex Discriminant Analysis (VDA)
Notation

- \( n \): number of observations
- \( p \): dimension of feature space
- \( k \): number of categories
Questions for Multicategory Discriminant Analysis

- How to choose category indicators?
- How to choose a loss function?
- How to minimize the loss function?
Equidistant Points in $R^{k-1}$

**Question**

How to choose class indicators?
Equidistant Points in $\mathbb{R}^{k-1}$

**Question**

How to choose class indicators?

**Proposition 1**

It is possible to choose $k$ equidistant points in $\mathbb{R}^{k-1}$ but not $k + 1$ equidistant points under the Euclidean norm.
Equidistant Points in $\mathbb{R}^{k-1}$

**Question**

How to choose class indicators?

**Proposition 1**

It is possible to choose $k$ equidistant points in $\mathbb{R}^{k-1}$ but not $k + 1$ equidistant points under the Euclidean norm.

The points occur at the vertices of a regular simplex

- 2 classes: -1, 1 (line)
- 3 classes: 3 vertices of an equilateral triangle circumscribed by the unit circle (plane)
- $k$ classes: $v_1, \ldots, v_k$ of a regular simplex in $\mathbb{R}^{k-1}$
Plot of Indicator Vertices for 3 Classes
Ridge Penalized $\epsilon$-insensitive Loss Function for VDA$_R$

- $\epsilon$-insensitive Euclidean Loss

$$f(z) = \|z\|_{2,\epsilon} = \max\{\|z\|_2 - \epsilon, 0\}$$

- Linear classifier $y = Ax + b$ to maintain parsimony
- Penalties on the slopes $a_{jl}$ imposed to avoid overfitting
- Minimizing the objective function to proceed classification

$$R(A, b) = \frac{1}{n} \sum_{i=1}^{n} f(y_i - Ax_i - b) + \lambda_R \sum_{j=1}^{k-1} \sum_{l=1}^{p} a_{jl}^2$$

$$= \frac{1}{n} \sum_{i=1}^{n} f(y_i - Ax_i - b) + \lambda_R \sum_{l=1}^{p} ||a_l||_2^2$$

where

- $y_i$ is the vertex assignment for case $i$
- $a^l_j$ is the $j$th row of a $k \times p$ matrix $A$ of regression coefficients
- $b$ is a $k \times 1$ column vector of intercepts
Multivariate Regression

• Prediction function \( y_i = A x_i + b \) for the \( i \)th observation:

\[
y_i = \begin{pmatrix} v_{y_i,1} \\ \vdots \\ v_{y_i,k-1} \end{pmatrix} = \begin{pmatrix} a_{11} & \cdots & a_{1p} \\ \vdots & \ddots & \vdots \\ a_{k-1,1} & \cdots & a_{k-1,p} \end{pmatrix} \begin{pmatrix} x_{i1} \\ \vdots \\ x_{ip} \end{pmatrix} + \begin{pmatrix} b_1 \\ \vdots \\ b_{k-1} \end{pmatrix}
\]

\uparrow \quad \uparrow

a_1 \quad a_p

• Linear system for \( n \) observations:

\[
\begin{pmatrix} y_1^t \\ \vdots \\ y_n^t \end{pmatrix} = \begin{pmatrix} x_1^t \\ \vdots \\ x_n^t \end{pmatrix} \begin{pmatrix} a_{11} & \cdots & a_{1p} \\ \vdots & \ddots & \vdots \\ a_{k-1,1} & \cdots & a_{k-1,p} \end{pmatrix}^t + \begin{pmatrix} b_1 \\ \vdots \\ b_{k-1} \end{pmatrix}^t
\]
A Toy Example for VDA

\( n = 300 \) training observations over \( k = 3 \) classes, each attached a normally distributed predictor with variance 1 and mean

\[
\mu = \begin{cases} 
-4, & \text{class } = 1 \\
0, & \text{class } = 2 \\
4, & \text{class } = 3
\end{cases}
\]

Compare:

1. least squares with class indicators \( v_j \) equated to \( e_j \) in \( \mathbb{R}^3 \) (indicator regression)
2. least squares with class indicators \( v_j \) equated to the vertices of an equilateral triangle
3. \( \epsilon \)-insensitive loss with the triangular vertices and \( \epsilon = 0.6 \)
4. \( \epsilon \)-insensitive loss with the triangular vertices and \( \epsilon = \frac{1}{2} \sqrt{2k/(k-1)} = 0.866 \)
Modified $\epsilon$-insensitive Loss

$$g(v) = \begin{cases} 
\|v\|_2^2 - \epsilon \\
\frac{(\|v\|_2^2 - \epsilon + \delta)^3(3\delta - \|v\|_2^2 + \epsilon)}{16\delta^3} \\
0
\end{cases}
$$

if $\|v\|_2 \geq \epsilon + \delta$

if $\|v\|_2 \in (\epsilon - \delta, \epsilon + \delta)$

if $\|v\|_2 \leq \epsilon - \delta$
Minimizing the objective function

\[ R(A, b) = \frac{1}{n} \sum_{i=1}^{n} g(y_i - Ax_i - b) + \lambda L \sum_{j=1}^{k-1} \sum_{l=1}^{p} |a_{jl}| \]

Although \( R(A, b) \) is non-differentiable, it possesses forward and backward directional derivatives along each coordinate direction.

Related work: L1MSVM (Wang and Shen 2007)
Cyclic Coordinate Descent

Forward and backward directional derivatives along $e_{jl}$ are

$$d_{ejl} R(A, b) = \lim_{\tau \downarrow 0} \frac{R(\theta + \tau e_{jl}) - R(\theta)}{\tau}$$

$$= \frac{1}{n} \sum_{i=1}^{n} \frac{\partial}{\partial a_{jl}} g(r_i) + \begin{cases} 
\lambda & \text{if } a_{jl} \geq 0 \\
-\lambda & \text{if } a_{jl} < 0 
\end{cases}$$

and

$$d_{-ejl} R(A, b) = \lim_{\tau \downarrow 0} \frac{R(\theta - \tau e_{jl}) - R(\theta)}{\tau}$$

$$= -\frac{1}{n} \sum_{i=1}^{n} \frac{\partial}{\partial a_{jl}} g(r_i) + \begin{cases} 
-\lambda & \text{if } a_{jl} > 0 \\
\lambda & \text{if } a_{jl} \leq 0 
\end{cases}$$

- If both $d_{ejl} R(A, b)$ and $d_{-ejl} R(A, b)$ are nonnegative $\Rightarrow$ skip
- If either directional derivative is negative $\Rightarrow$ solve for the minimum in the corresponding direction
Newton’s Updates

If \( r_i^m \) is the value of the \( i \)th residual at iteration \( m \)

\[
a_{jl}^{m+1} = a_{jl}^m - \frac{1}{n} \sum_{i=1}^n \frac{\partial}{\partial a_{jl}} g(r_i^m) + \begin{cases} 
\lambda & \text{if } a_{jl}^m \geq 0 \\
-\lambda & \text{if } a_{jl}^m < 0 
\end{cases} \\
\frac{1}{n} \sum_{i=1}^n \frac{\partial^2}{\partial a_{jl}^2} g(r_i^m)
\]

and

\[
b_j^{m+1} = b_j^m - \frac{1}{n} \sum_{i=1}^n \frac{\partial}{\partial b_j} g(r_i^m) \\
\frac{1}{n} \sum_{i=1}^n \frac{\partial^2}{\partial b_j^2} g(r_i^m)
\]
Euclidean Penalty for Grouped Effects

- Selection of groups of variables rather than individual variables ("all-in-or-all-out")
- Euclidean norm $\lambda_E \| a_l \|_2$ is an ideal group penalty since it couples parameters and preserves convexity (Wu and Lange 2008)
- In multicategory classification, the slopes of a single predictor for different dimensions of $R^{k-1}$ form a natural group
- In $\text{VDA}_E$, we minimize the objective function

$$R(A, b) = \frac{1}{n} \sum_{i=1}^{n} g(y_i - Ax_i - b) + \lambda_E \sum_{l=1}^{p} \| a_l \|_2$$
In VDA\textsubscript{LE}, we minimize the objective function

\[ R(A, b) = \frac{1}{n} \sum_{i=1}^{n} g(y_i - Ax_i - b) + \lambda_L \sum_{j=1}^{k-1} \sum_{l=1}^{p} |a_{jl}| + \lambda_E \sum_{l=1}^{p} \|a_l\|_2 \]

- \( \lambda_E = 0 \Rightarrow \text{VDA}_L \)
- \( \lambda_L = 0 \Rightarrow \text{VDA}_E \)
Fisher Consistency of $\epsilon$-Insensitive Loss
Fisher Consistency of \( \epsilon \)-Insensitive Loss

Proposition 2

If a minimizer \( f^*(x) \) of \( \mathbb{E}[\|Y - f(X)\|_\epsilon \mid X = x] \) with \( \epsilon = \frac{1}{2} \sqrt{2k/(k - 1)} \) lies closest to vertex \( v_l \), then \( p_l(x) = \max_j p_j(x) \). Either \( f^*(x) \) occurs exterior to all of the \( \epsilon \)-insensitive balls or on the boundary of the ball surrounding \( v_l \). The assigned vertex \( v_l \) is unique if the \( p_j(x) \) are distinct.
Introduction  Vertex Discriminant Analysis (VDA)  Fisher Consistency of $\epsilon$-Insensitive Loss  Numerical Examples  Discussion

$p = (1/3, 1/3, 1/3)$

$p = (0.37, 0.37, 0.26)$

$p = (0.6, 0.3, 0.1)$

$p = \left( \frac{1}{3} + t, \frac{1}{3} - \frac{t}{4}, \frac{1}{3} - \frac{3t}{3} \right)$, where $t = 0.025$
Numerical Examples
Simulation Example

An example in Wang and Shen (2007)

- $k = 3$, $n = 60$, and $p = 10, 20, 40$ (overdetermined), $80, 160$ (underdetermined)
- $x_{ij}$ are i.i.d. $N(0, 1)$ for $j > 2$ and have mean $a_j$ for $j \leq 2$

$$ (a_1, a_2) = \begin{cases} (\sqrt{2}, \sqrt{2}) & \text{for class 1} \\ (\sqrt{2}, -\sqrt{2}) & \text{for class 2} \\ (-\sqrt{2}, -\sqrt{2}) & \text{for class 3} \end{cases} $$

- 60 training cases are spread evenly across the 3 classes and 30,000 testing cases
- To compare the three modified VDA methods with L1MSVM (Wang and Shen 2007) and L2MSVM (Lee et al. 2004)
<table>
<thead>
<tr>
<th>$p$</th>
<th>Bayes Error</th>
<th>$\text{VDA}<em>{\text{LE}}$, $\text{VDA}</em>{\text{L}}$, and $\text{VDA}_{\text{E}}$ Error</th>
<th># Var</th>
<th>Time</th>
<th>Error</th>
<th>L1MSVM Error</th>
<th>L2MSVM Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>10.81%</td>
<td>12.38% (0.10%)</td>
<td>2.93 (0.11)</td>
<td>0.0071 (0.0008)</td>
<td>13.61%</td>
<td>15.44%</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>14.42% (0.14%)</td>
<td>4.82 (0.34)</td>
<td>0.0050 (0.0008)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>12.70% (0.12%)</td>
<td>3.08 (0.10)</td>
<td>0.0074 (0.0008)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>10.81%</td>
<td>12.65% (0.11%)</td>
<td>3.87 (0.14)</td>
<td>0.0104 (0.0007)</td>
<td>14.06%</td>
<td>17.81%</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>15.38% (0.19%)</td>
<td>6.89 (0.64)</td>
<td>0.0043 (0.0007)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>13.08% (0.13%)</td>
<td>4.68 (0.16)</td>
<td>0.0130 (0.0007)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>10.81%</td>
<td>13.01% (0.13%)</td>
<td>5.51 (0.21)</td>
<td>0.0178 (0.0010)</td>
<td>14.94%</td>
<td>20.01%</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>15.66% (0.20%)</td>
<td>8.76 (0.93)</td>
<td>0.0056 (0.0007)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>13.50% (0.13%)</td>
<td>7.15 (0.23)</td>
<td>0.0247 (0.0008)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>10.81%</td>
<td>13.33% (0.14%)</td>
<td>8.81 (0.33)</td>
<td>0.0345 (0.0015)</td>
<td>15.68%</td>
<td>21.81%</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>16.15% (0.22%)</td>
<td>12.81 (1.26)</td>
<td>0.0089 (0.0008)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>13.99% (0.15%)</td>
<td>12.12 (0.35)</td>
<td>0.0440 (0.0014)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>160</td>
<td>10.81%</td>
<td>14.02% (0.14%)</td>
<td>13.00 (0.55)</td>
<td>0.0647 (0.0030)</td>
<td>16.58%</td>
<td>27.54%</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>17.12% (0.23%)</td>
<td>20.59 (1.78)</td>
<td>0.0180 (0.0008)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>15.08% (0.19%)</td>
<td>19.82 (0.50)</td>
<td>0.0830 (0.0022)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Stability Selection (Meinshausen and Buehlmann 2009) for $p = 160$
Real Data Examples: Underdetermined Problems

Average 3-fold cross-validated testing errors (%) over 50 random partitions for six benchmark cancer data sets

<table>
<thead>
<tr>
<th>Method</th>
<th>Leukemia</th>
<th>Colon</th>
<th>Prostate</th>
<th>Lymphoma</th>
<th>SRBCT</th>
<th>Brain</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(2, 72, 3571)</td>
<td>(2, 62, 2000)</td>
<td>(2, 102, 6033)</td>
<td>(3, 62, 4026)</td>
<td>(4, 63, 2308)</td>
<td>(5, 42, 5597)</td>
</tr>
<tr>
<td>VDA&lt;sub&gt;LE&lt;/sub&gt;</td>
<td>1.56 (45.5)</td>
<td>9.68 (37.1)</td>
<td>5.48 (48.3)</td>
<td>1.66 (71.2)</td>
<td>1.58 (65.2)</td>
<td>23.80 (76.2)</td>
</tr>
<tr>
<td>VDA&lt;sub&gt;L&lt;/sub&gt;</td>
<td>7.14 (40.7)</td>
<td>14.26 (49.3)</td>
<td>9.83 (68.7)</td>
<td>14.36 (56.6)</td>
<td>9.52 (53.5)</td>
<td>48.86 (56.1)</td>
</tr>
<tr>
<td>VDA&lt;sub&gt;E&lt;/sub&gt;</td>
<td>3.02 (89.4)</td>
<td>11.08 (76.7)</td>
<td>6.76 (140.4)</td>
<td>3.25 (92.3)</td>
<td>1.58 (79.9)</td>
<td>30.44 (84.9)</td>
</tr>
<tr>
<td>BagBoost</td>
<td>4.08</td>
<td>16.10</td>
<td>7.53</td>
<td>1.62</td>
<td>1.24</td>
<td>23.86</td>
</tr>
<tr>
<td>Boosting</td>
<td>5.67</td>
<td>19.14</td>
<td>8.71</td>
<td>6.29</td>
<td>6.19</td>
<td>27.57</td>
</tr>
<tr>
<td>RanFor</td>
<td>1.92</td>
<td>14.86</td>
<td>9.00</td>
<td>1.24</td>
<td>3.71</td>
<td>33.71</td>
</tr>
<tr>
<td>SVM</td>
<td>1.83</td>
<td>15.05</td>
<td>7.88</td>
<td>1.62</td>
<td>2.00</td>
<td>28.29</td>
</tr>
<tr>
<td>PAM</td>
<td>3.75</td>
<td>11.90</td>
<td>16.53</td>
<td>5.33</td>
<td>2.10</td>
<td>25.29</td>
</tr>
<tr>
<td>DLDA</td>
<td>2.92</td>
<td>12.86</td>
<td>14.18</td>
<td>2.19</td>
<td>2.19</td>
<td>28.57</td>
</tr>
<tr>
<td>KNN</td>
<td>3.83</td>
<td>16.38</td>
<td>10.59</td>
<td>1.52</td>
<td>1.43</td>
<td>29.71</td>
</tr>
</tbody>
</table>
Discussion
### Summary

- VDA and its various modifications are competitive among the competing methods
- Virtues of VDA: parsimony, robustness, speed, and symmetry
- Four VDA methods
  - $\text{VDA}_R$: robustness and symmetry but falling behind in parsimony and speed, highly recommended for problems with a handful of predictors
  - $\text{VDA}_{LE}$: best performance on high-dimensional problems, though sacrificing a little symmetry for extra parsimony
  - $\text{VDA}_E$: robustness, speed, and symmetry
  - $\text{VDA}_L$: putting too high a premium on parsimony at the expense of symmetry
Future Work

- Euclidean penalties for grouped effects (Wu and Lange 2008)
- Redesigning the class vertices if they are not symmetrically distributed
- Nonlinear classifier
- Extension to multi-task learning
- More theoretical studies
- Further increasing computing speed in parallel computing