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

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 ϵ-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 ϵ-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 $x \in R^p$
- **Observations**: category membership indicator y and feature vector $x \in R^p$
- **Discriminant rule**: divide 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 $\mathrm{E}\left[L(Y, X)\right] = \mathrm{E}\left\{\mathrm{E}\left[L(Y, X)|X\right]\right\}$
  - 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 R^{k-1}

Question

How to choose class indicators?

Proposition 1

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

Equidistant Points in R^{k-1}

Question

How to choose class indicators?

Proposition 1

It is possible to choose k equidistant points in 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 R^{k-1}

Plot of Indicator Vertices for 3 Classes

Ridge Penalized ϵ-insensitive Loss Function for $\mathrm{VDA_R}$

- ϵ-insensitive Euclidean Loss

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

- 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

$$\begin{aligned} R(A,b) &= \frac{1}{n}\sum_{i=1}^{n} f(y_i - Ax_i - b) + \lambda_{\mathrm{R}} \sum_{j=1}^{k-1}\sum_{l=1}^{p} a_{jl}^2 \\ &= \frac{1}{n}\sum_{i=1}^{n} f(y_i - Ax_i - b) + \lambda_{\mathrm{R}} \sum_{l=1}^{p} \|a_l\|_2^2 \end{aligned}$$

where

- y_i is the vertex assignment for case i
- a_j^t is the jth 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 $y_i = Ax_i + b$ for the ith observation:

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

$$\begin{matrix} \uparrow & & \uparrow \\ a_1 & & a_p \end{matrix}$$

- Linear system for n observations:

$$\begin{pmatrix} y_1^t \\ \vdots \\ y_n^t \end{pmatrix} = \begin{pmatrix} x_i^t \\ \vdots \\ x_n^t \end{pmatrix} \begin{pmatrix} a_{11} & \dots & a_{1p} \\ & \ddots & \\ a_{k-1,1} & \dots & a_{k-1,p} \end{pmatrix}^t + \begin{pmatrix} b_1 \\ \vdots \\ b_{k-1} \end{pmatrix}^t$$

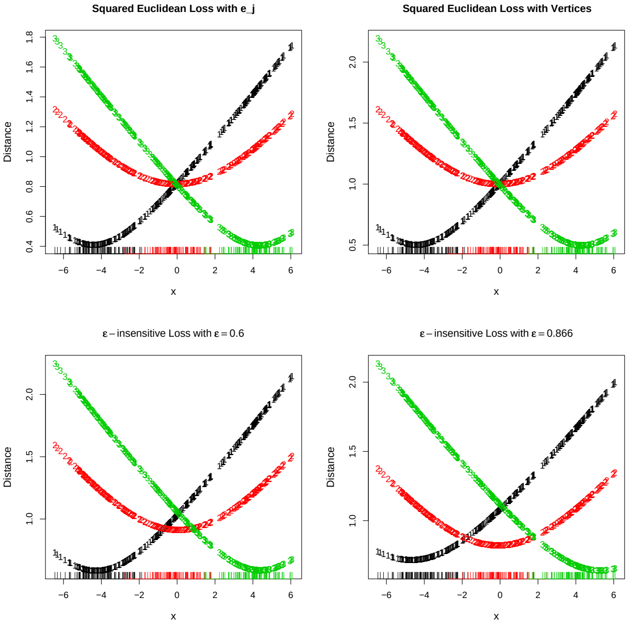
A Toy Example for VDA

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

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

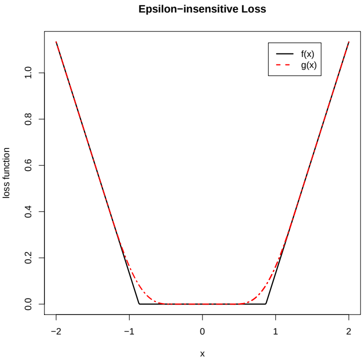
Compare:

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


Squared Euclidean Loss with e_j
Squared Euclidean Loss with Vertices
ε – insensitive Loss with ε = 0.6
ε – insensitive Loss with ε = 0.866
Distance
x
−6
−4
−2
0
2
4
6
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
0.5
1.0
1.5
2.0

Modified ϵ-insensitive Loss

$$g(v) = \begin{cases} \|v\|_2 - \epsilon & \text{if } \|v\|_2 \geq \epsilon + \delta \\ \frac{(\|v\|_2 - \epsilon + \delta)^3 (3\delta - \|v\|_2 + \epsilon)}{16\delta^3} & \text{if } \|v\|_2 \in (\epsilon - \delta, \epsilon + \delta) \\ 0 & \text{if } \|v\|_2 \leq \epsilon - \delta \end{cases}$$



VDA_L

- Minimizing the objective function

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

- 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

$$
\begin{aligned}
d_{e_{jl}} R(A, b) &= \lim_{\tau \downarrow 0} \frac{R(\theta + \tau e_{jl}) - R(\theta)}{\tau} \\
&= \frac{1}{n} \sum_{i=1}^{n} \frac{\partial}{\partial a_{jl}} g(r_i) + \begin{cases} \lambda & \text{if } a_{jl} \geq 0 \\ -\lambda & \text{if } a_{jl} < 0 \end{cases}
\end{aligned}
$$

and

$$
\begin{aligned}
d_{-e_{jl}} R(A, b) &= \lim_{\tau \downarrow 0} \frac{R(\theta - \tau e_{jl}) - R(\theta)}{\tau} \\
&= -\frac{1}{n} \sum_{i=1}^{n} \frac{\partial}{\partial a_{jl}} g(r_i) + \begin{cases} -\lambda & \text{if } a_{jl} > 0 \\ \lambda & \text{if } a_{jl} \leq 0 \end{cases}
\end{aligned}
$$

- If both $d_{e_{jl}} R(A, b)$ and $d_{-e_{jl}} R(A, b)$ 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 ith residual at iteration m

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

and

$$b_j^{m+1} = b_j^m - \frac{\frac{1}{n}\sum_{i=1}^n \frac{\partial}{\partial b_j} g(r_i^m)}{\frac{1}{n}\sum_{i=1}^n \frac{\partial^2}{\partial b_j^2} g(r_i^m)}$$

Euclidean Penalty for Grouped Effects

- Selection of groups of variables rather than individual variables ("all-in-or-all-out")
- Euclidean norm $\lambda_{\mathrm{E}}\|a_l\|_2$ 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 $\mathrm{VDA_E}$, we minimize the objective function

$$R(A, b) = \frac{1}{n}\sum_{i=1}^{n} g(y_i - Ax_i - b) + \lambda_{\mathrm{E}}\sum_{l=1}^{p}\|a_l\|_2$$

VDA_{LE}

- In VDA_{LE}, we minimize the objective function

$$R(A, b) = \frac{1}{n}\sum_{i=1}^{n} g(y_i - Ax_i - b) + \lambda_{\text{L}} \sum_{j=1}^{k-1}\sum_{l=1}^{p} |a_{jl}| + \lambda_{\text{E}} \sum_{l=1}^{p} \|a_l\|_2$$

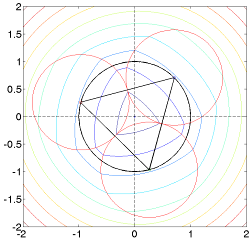
- $\lambda_{\text{E}} = 0 \Rightarrow \text{VDA}_{\text{L}}$
- $\lambda_{\text{L}} = 0 \Rightarrow \text{VDA}_{\text{E}}$

Fisher Consistency of ϵ-Insensitive Loss

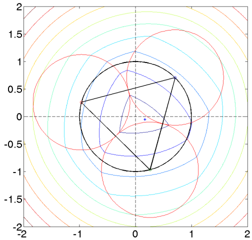
Fisher Consistency of ϵ-Insensitive Loss

Proposition 2

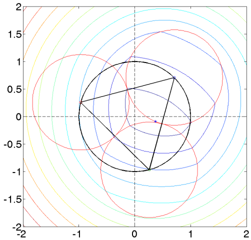
If a minimizer $f^*(x)$ of $\mathrm{E}\left[\|Y - f(X)\|_\epsilon \mid X = x\right]$ with $\epsilon = \frac{1}{2}\sqrt{2k/(k-1)}$ lies closest to vertex v_l, then $p_l(x) = \max_j p_j(x)$. Either $f^*(x)$ occurs exterior to all of the ϵ-insensitive balls or on the boundary of the ball surrounding v_l. The assigned vertex v_l is unique if the $p_j(x)$ are distinct.



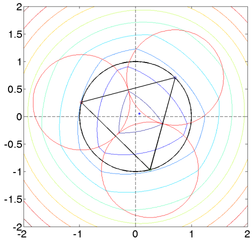
$p = (1/3, 1/3, 1/3)$



$p = (0.37, 0.37, 0.26)$



$p = (0.6, 0.3, 0.1)$



$p = (\frac{1}{3} + t, \frac{1}{3} - \frac{t}{4}, \frac{1}{3} - \frac{3t}{3})$, where $t = 0.025$

Numerical Examples

Simulation Example

An example in Wang and Shen (2007)

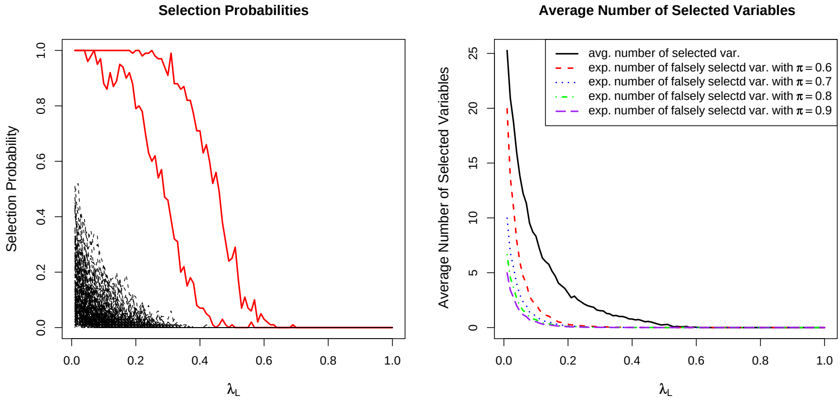
- $k = 3$, $n = 60$, and $p = 10, 20, 40$ (overdetermind), $80, 160$ (underdetermined)
- x_{ij} are i.i.d. $N(0, 1)$ for $j > 2$ and have mean a_j for $j \leq 2$

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

- 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)

p	Bayes	VDA_{LE}, VDA_{L}, and VDA_{E}			L1MSVM	L2MSVM
		Error	# Var	Time	Error	Error
10	10.81%	12.38% (0.10%)	2.93 (0.11)	0.0071 (0.0008)	13.61%	15.44%
		14.42% (0.14%)	4.82 (0.34)	0.0050 (0.0008)		
		12.70% (0.12%)	3.08 (0.10)	0.0074 (0.0008)		
20	10.81%	12.65% (0.11%)	3.87 (0.14)	0.0104 (0.0007)	14.06%	17.81%
		15.38% (0.19%)	6.89 (0.64)	0.0043 (0.0007)		
		13.08% (0.13%)	4.68 (0.16)	0.0130 (0.0007)		
40	10.81%	13.01% (0.13%)	5.51 (0.21)	0.0178 (0.0010)	14.94%	20.01%
		15.66% (0.20%)	8.76 (0.93)	0.0056 (0.0007)		
		13.50% (0.13%)	7.15 (0.23)	0.0247 (0.0008)		
80	10.81%	13.33% (0.14%)	8.81 (0.33)	0.0345 (0.0015)	15.68%	21.81%
		16.15% (0.22%)	12.81 (1.26)	0.0089 (0.0008)		
		13.99% (0.15%)	12.12 (0.35)	0.0440 (0.0014)		
160	10.81%	14.02% (0.14%)	13.00 (0.55)	0.0647 (0.0030)	16.58%	27.54%
		17.12% (0.23%)	20.59 (1.78)	0.0180 (0.0008)		
		15.08% (0.19%)	19.82 (0.50)	0.0830 (0.0022)		

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

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

Discussion

Summary

- VDA and its various modifications are competitive among the competing methods
- Virtues of VDA: parsimony, robustness, speed, and symmetry
- Four VDA methods
    - VDA_R: robustness and symmetry but falling behind in parsimony and speed, highly recommended for problems with a handful of predictors
    - VDA_LE: best performance on high-dimensional problems, though sacrificing a little symmetry for extra parsimony
    - VDA_E: robustness, speed, and symmetry
    - 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