

An MM Algorithm for Multicategory Vertex Discriminant Analysis

Tong Tong Wu

Department of Epidemiology and Biostatistics
University of Maryland, College Park

May 22, 2008

Joint work with Professor Kenneth Lange, to appear in *JCGS*

Outline

- 1 Introduction
- 2 Multicategory VDA
 - Category Indicator: Equidistant Points in R^k
 - Regularized ϵ -insensitive Loss Function
 - Optimization by An MM Algorithm
- 3 Real Data Examples
- 4 Simulation Studies
- 5 Discussion

oooooooooooooooooooooooooooo

Introduction

Overview of Vertex Discriminant Analysis (VDA)

- A new method of supervised learning
- Based on linear discrimination among the vertices of a regular simplex in Euclidean space
- Each vertex represents a different category
- A regression problem involving ϵ -insensitive residuals and a quadratic penalty on the coefficients of the linear predictors
- Minimizing the objective function by a primal MM algorithm
 - Relying on quadratic majorization and iteratively reweighted least squares
 - Simpler to program than dual optimization approaches
 - Accelerated by step doubling
- Competitive in statistical accuracy and computational speed with the best algorithms for discriminant analysis

Review of Discriminant Analysis

- Category membership indicator y and feature vector $x \in R^p$
- Purpose: categorize objects based on a fixed number of observed features $x \in R^p$
- Discriminant rule: divide R^p into disjoint regions corresponding to the 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 $E[L(Y, X)] = E\{E[L(Y, X)|X]\}$
 - Average conditional loss $n^{-1} \sum_{i=1}^n L(y_i, x_i)$ with a penalty term

Trend of Discriminant Analysis

- Expand the number of features by adding simple functions and operate by linear methods on the resulting high-dimensional feature space (Cristianini and Shawe-Taylor 2000; Wahba 1999)
 - Danger: over-parametrization
 - Remedy: regularizing estimation of regression coefficients by adding penalty terms that shrink estimates toward the origin
- Increase the number of categories

Existing Methods

- Linear discriminant analysis (LDA)
- Quadratic discriminant analysis (QDA)
- K -nearest neighbor
- Support vector machines (SVM)
- Classification and regression trees (CART)



Multicategory Vertex Discriminant Analysis



Questions for Multicategory Discriminant Analysis

- How to choose category indicators?
- How to choose a loss function?
- How to minimize the loss function?



Notation

- n : number of observations
- p : dimension of feature space
- $k + 1$: number of categories

Equidistant Points in R^k

Question

How to choose class indicators?

Equidistant Points in R^k

Question

How to choose class indicators?

Answer: Equidistant points in R^k

Equidistant Points in R^k

Question

How to choose class indicators?

Answer: Equidistant points in R^k

Proposition 1

It is possible to choose $k + 1$ equidistant points in R^k but not $k + 2$ equidistant points under the Euclidean norm.



Equidistant Points in R^k

- The points occur at the vertices of a regular simplex
- One possible way of constructing the simplex

$$y_j = \begin{cases} k^{-1/2}\mathbf{1} & \text{if } j = 0 \\ c\mathbf{1} + de_j & \text{if } 1 \leq j \leq k \end{cases}$$

where

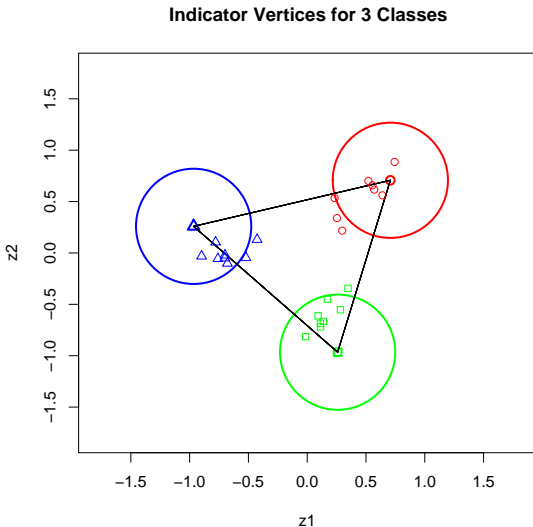
$$c = -\frac{1 + \sqrt{k+1}}{k^{3/2}}, \quad d = \sqrt{\frac{k+1}{k}},$$

and e_j is the standard unit vector with 1 in position j and 0 in all other positions.

- Any rotated, reflected, dilated, or translated version of these $k + 1$ vectors retains the equidistant property

○○○○●○○○○○○○○○○○○○○○○○○○○

Plot of Indicator Vertices for 3 Classes



ϵ -insensitive Loss vs. Hinge Loss

- Hard to quantify the costs of various errors
 - Generic loss functions and linear predictors that have good empirical error rates
 - Hinge loss for SVM (Wahba 1999)

$$L(y, x) = [1 - yf(x)]_+$$

where $(a)_+ = \max(a, 0)$

- ϵ -insensitive loss for regression (Vapnik 1995; Hastie et al. 2001; Scholkopf and Smola 2002)

$$L(y, x) = |y - a^t x - b|_\epsilon$$

where $|v|_\epsilon = \max\{|v| - \epsilon, 0\}$



ϵ -insensitive Loss vs. Hinge Loss

- Similarities

- Both are more resistant to outliers than squared error loss (Liu et al. 2005; Shen et al. 2003)
- Both assume it does not make much difference how close a linear predictor is to its class indicator
- Choose the closest of the possible class indicators
- Training observations on the boundary of the ϵ -insensitive sphere act as support vectors, and observations within the sphere do not contribute to the estimation of regression coefficients

- Dissimilarities

- Hinge loss imposes no penalty for over-prediction when a linear predictor falls on the correct side of its class indicator (Wahba 1999)
- Hinge loss does not generalize as well to higher dimensions



Regularized Loss Function

Given our identification of class indicators with vertices, the regularized loss function is defined as

ϵ -insensitive Loss

$$R(A, b) = \frac{1}{n} \sum_{i=1}^n \|y_i - Ax_i - b\|_{\epsilon} + \lambda \sum_{j=1}^k \|a_j\|^2$$

where

- y_i is the vertex assignment for case i
- a_j^t 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
- $\|v\|_{\epsilon} = \max\{\|v\| - \epsilon, 0\}$ is ϵ -insensitive Euclidean distance



Regularized Loss Function

ϵ -insensitive Loss

$$R(A, b) = \frac{1}{n} \sum_{i=1}^n \|y_i - Ax_i - b\|_{\epsilon} + \lambda \sum_{j=1}^k \|a_j\|^2$$

Convexity

- The first term is convex in A and b
- The second term is strictly convex in A
- In most regions of parameter space it is strictly convex and therefore we can expect a unique minimum point
- Once A and b are estimated, we can assign a new case to the closest vertex, and hence category

How to Minimize Loss Function

- ϵ -insensitive loss is nondifferentiable
- Dual optimization algorithms are hard to understand and harder yet to implement
- A primal MM algorithm is proposed and tested (Groenen et al. 2006, binary classification using a different loss function)

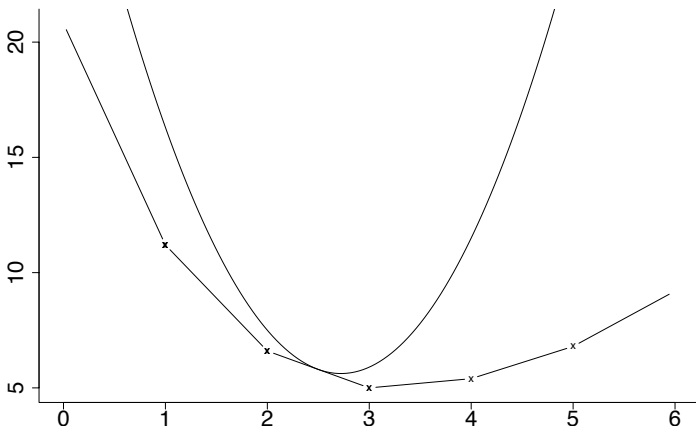
Overview of the MM Algorithm

An MM algorithm substitutes a simple optimization problem for a difficult optimization problem

- In minimization MM stands for majorize-minimize, and in maximization MM stands for minorize-maximize
- An MM algorithm operates by creating a surrogate function that majorizes or minorizes the objective function. When the surrogate function is optimized, the objective function is driven uphill or downhill as needed
- The MM algorithm is not an algorithm, but a prescription for constructing optimization algorithms
- The EM algorithm from statistics is a special case



An Example: A Sum of Quadratic Majorizers for Finding a Sample Median



Rationale for the MM Principle

- Generate an algorithm that avoids matrix inversion
- Separate the parameters of a problem
- Linearize an optimization problem
- Deal gracefully with equality and inequality constraints
- Turn a nondifferentiable problem into a smooth problem



Majorization and Definition of the Algorithm

- A function $g(\theta | \theta^n)$ is said to **majorize** the function $f(\theta)$ at θ^n provided

$$\begin{aligned} f(\theta^n) &= g(\theta^n | \theta^n) \\ f(\theta) &\leq g(\theta | \theta^n) \quad \text{for all } \theta \end{aligned}$$

- A function $g(\theta | \theta^n)$ is said to **minorize** the function $f(\theta)$ at θ^n provided $-g(\theta | \theta^n)$ majorizes $-f(\theta)$
- In minimization, we choose a majorizing function $g(\theta | \theta^n)$ and minimize it, which produces the next point θ^{n+1} in the algorithm



Descent Property

- The descent property follows from the definitions and

$$f(\theta^{n+1}) \leq g(\theta^{n+1} | \theta^n) \leq g(\theta^n | \theta^n) = f(\theta^n)$$

- An MM minimization algorithm satisfies the descent property $f(\theta^{n+1}) \leq f(\theta^n)$ with strict inequality unless both

$$\begin{aligned}g(\theta^{n+1} | \theta^n) &= g(\theta^n | \theta^n) \\ f(\theta^{n+1}) &= g(\theta^{n+1} | \theta^n)\end{aligned}$$

- The descent property makes the MM algorithm very stable

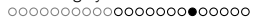
Majorization of ϵ -insensitive Distance

Repeated application of the Cauchy-Schwarz inequality produces the majorizer of $\|x\|_\epsilon$

Majorizing function

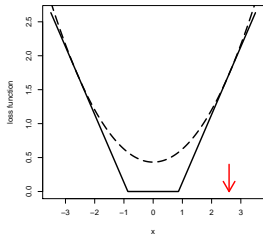
$$q(x \mid x^{(m)}) = \begin{cases} \frac{1}{2\|x^{(m)}\|} \|x\|^2 + \frac{1}{2} \|x^{(m)}\| - \epsilon & \text{if } \|x^{(m)}\| \geq 2\epsilon \\ \frac{1}{4(\epsilon - \|x^{(m)}\|)} \|x - x^{(m)}\|^2 & \text{if } \|x^{(m)}\| < \epsilon \\ \frac{1}{4(\epsilon - \|z\|)} \|x - z\|^2 & \text{if } \epsilon < \|x^{(m)}\| < 2\epsilon \end{cases}$$

where in the last case $z = cx^{(m)}$ and $c = 2\epsilon/\|x^{(m)}\| - 1$. There is no majorization in the anomalous situation $\|x^{(m)}\| = \epsilon$.

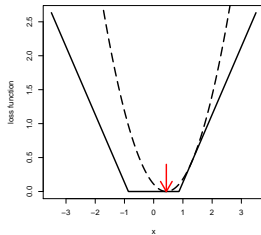


Majorizer of $\|x\|_\epsilon$

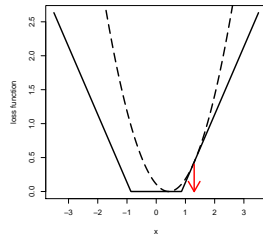
Majorization of the Loss Function



Majorization of the Loss Function



Majorization of the Loss Function





Majorization of the Objective Function

Minimization of the surrogate function reduces to weighted least squares estimation

$$\begin{aligned}
 R(A, b) &= \frac{1}{n} \sum_{i=1}^n \|y_i - Ax_i - b\|_{\epsilon} + \lambda \sum_{j=1}^k \|a_j\|^2 \\
 &\leq \frac{1}{n} \sum_{i=1}^n w_i \|y_i - Ax_i - b - v_i\|^2 + \lambda \sum_{j=1}^k \|a_j\|^2 + d \\
 &= \frac{1}{n} \sum_{j=1}^k \left[\sum_{i=1}^n w_i (y_{ij} - a_j^t x_i - b_j - v_{ij})^2 + \lambda \|a_j\|^2 \right] + d
 \end{aligned}$$



Majorization of the Objective Function

- Case weights

$$w_i = \begin{cases} \frac{1}{2\|r_i\|} & \text{if } \|r_i\| \geq 2\epsilon \\ \frac{1}{4(\epsilon - \|r_i\|)} & \text{if } \|r_i\| < \epsilon \\ \frac{1}{4(\|r_i\| - \epsilon)} & \text{if } \epsilon < \|r_i\| < 2\epsilon \end{cases}$$

- Argument shifts

$$v_i = \begin{cases} \mathbf{0} & \text{if } \|r_i\| \geq 2\epsilon \\ r_i & \text{if } \|r_i\| < \epsilon \\ \left(\frac{2\epsilon}{\|r_i\|} - 1\right) r_i & \text{if } \epsilon < \|r_i\| < 2\epsilon \end{cases}$$

- Constant d depends on the residual r_i at iteration m



Cases with $\|r_i\| = \epsilon$

- As $\|r_i\| \rightarrow \epsilon$, the case weight $w_i \rightarrow \infty$
- Truncate the case weights by taking $w_i = \frac{1}{4\delta}$ for $\epsilon - \delta \leq \|r_i\| \leq \epsilon + \delta$ for a small $\delta \in [10^{-7}, 10^{-3}]$
- As the tangent point r_i of the surrogate function moves across its 3 domains, the center and truncated weight of the surrogate function vary continuously
- Although truncation negates the guarantee of the algorithm's descent property, it does no practical harm

Computational Issues

- Fast, accurate code is widely available for weighted least squares estimation
- Standard methods based on Cholesky decomposition or sweeping involve inversion of the matrix X^tWX and take on the order of $O(p^3)$ arithmetic operations
- All k problems share the same matrix X^tWX , so a single matrix inversion suffices per MM iteration rather than k separate matrix inversions
- Step doubling is a standard tactic that usually halves the number of iterations until convergence (de Leeuw and Heiser 1980; Lange 1995)
- The MM algorithm can be improved by alternating it with a modified version of Newton's method



Pseudocode for VDA

- 1 Set the iteration counter $m = 0$, and initialize $A^{(0)} = \mathbf{0}$ and $b^{(0)} = \mathbf{0}$;
- 2 Define $y_i = v_j$ if the i th subject belongs to category j , where

$$v_j = \begin{cases} k^{-1/2}\mathbf{1}, & j = 1 \\ c\mathbf{1} + de_{j-1}, & 2 \leq j \leq k+1 \end{cases}$$

$$c = -\frac{1+\sqrt{k+1}}{k^{3/2}}, \text{ and } d = \sqrt{\frac{k+1}{k}};$$

- 3 Majorize the regularized loss function with i th current residual $r_i^{(m)} = y_i - A^{(m)}x_i - b^{(m)}$;
- 4 Minimize the surrogate function and determine $A^{(m+1)}$ and $b^{(m+1)}$ by solving k sets of linear equations;
- 5 If $\|A^{(m+1)} - A^{(m)}\| < \gamma$ and $|R(A^{(m+1)}, b^{(m+1)}) - R(A^{(m)}, b^{(m)})| < \gamma$ both hold for $\gamma = 10^{-4}$, then stop;
- 6 Otherwise repeat steps 3 through 5

Real Data Examples

Real Data Examples

- Four standard data sets (wine, glass, zoo, and lymphography) from the UCI machine learning repository (<http://www.ics.uci.edu/~mllearn/>)
- The error rates are average misclassification rates based on 10-fold cross validation
- To choose the tuning parameters ϵ and λ , we evaluated the cross-validated error over a two-dimensional grid
 - The optimal value of ϵ is close to our suggested value of $\frac{1}{n}\sqrt{(2k+2)/k}$
 - The error rate for VDA is insensitive to λ over the broad range $[10^{-5}, 10^3]$

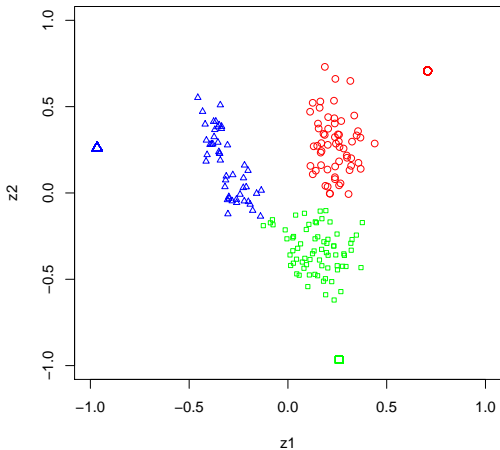
Empirical Examples from the UCI Data Repository

| Method ($n, p, k + 1$) | Wine (178,13,3) | Glass (214,9,7) | Zoo (101,16,7) | Lympho (148,18,4) |
|-----------------------------|--------------------|--------------------|-------------------|----------------------|
| VDA | 0 | 0.2970 | 0 | 0.0541 |
| LDA | 0.0112 | 0.4065 | NA | 0.0878 |
| QDA | 0.0169 | NA | NA | NA |
| KNN | 0.0506 | 0.2991 | 0.0792 | 0.1351 |
| OVR | 0.0169 | 0.3458 | 0.0099 | 0.0541 |
| MSVM | 0.0169 | 0.3645 | NA | NA |
| AltMSVM* | 0.0169 | 0.3170 | NA | NA |
| CART | 0.1404 | 0.1449 | 0.1683 | 0.1351 |
| Random Forest | 0.0674 | 0.1589 | 0.0297 | 0.0135 |

*Guermeur (2002)



VDA Prediction for Wine Data



Simulation Studies

Simulation 1

An example in Lee et al. (2004) designed to test multiclass discrimination with a single predictor

- $X \sim$ i.i.d. $U_{[0,1]}$
- Given $X = x$, the probabilities of assigning an observation to the three classes are

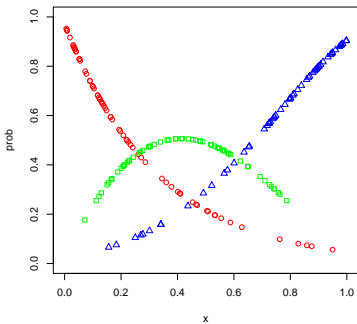
$$p_1(x) = 0.97e^{-3x}, p_3(x) = e^{-2.5(x-1.2)^2}, p_2(x) = 1 - p_1(x) - p_3(x)$$

- training datasets of size 200
- testing datasets of size 10,000
- 100 random samples

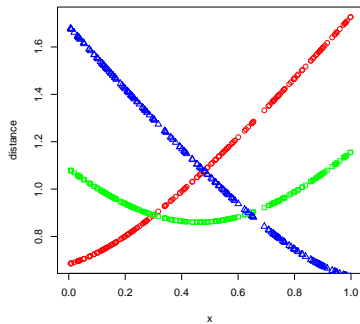


Plots of Simulation 1

Conditional Probabilities of Class Assignment



Distance to Artificial Vertices



Comparison of Error Rates for Simulation Example 1

| Method | Testing Error Rate | Standard Error | Time |
|---------------|--------------------|----------------|-------------|
| Bayes Rule | 0.3841 | NA | NA |
| VDA | 0.3940 | 0.0006 | 0.16 |
| LDA | 0.4084 | 0.0011 | 0.63 |
| QDA | 0.4068 | 0.0011 | 0.54 |
| KNN | 0.4288 | 0.0016 | 0.17 |
| OVR | 0.4164 | 0.0017 | 2.11 |
| MSVM | 0.4077 | 0.0015 | 13.86 |
| CART | 0.4440 | 0.0035 | 0.17 |
| Random Forest | 0.4760 | 0.0019 | 2.26 |

Simulation 2

A standard bench-mark waveform data of Breiman et al. (1984) featured as Example 12.7 of Hastie et al. (2001)

- $p = 21$
- j th predictor X_{ij} of observation i is generated via

$$X_{ij} = \begin{cases} U_i h_1(j) + (1 - U_i) h_2(j) + Z_{ij} & \text{for category 1} \\ U_i h_1(j) + (1 - U_i) h_3(j) + Z_{ij} & \text{for category 2} \\ U_i h_2(j) + (1 - U_i) h_3(j) + Z_{ij} & \text{for category 3} \end{cases}$$

where $U_i \sim U_{[0,1]}$, $Z_{ij} \sim \mathcal{N}(0, 1)$, $h_1(j) = \max\{6 - |j - 11|, 0\}$,
 $h_2(j) = h_1(j - 4)$, and $h_3(j) = h_1(j + 4)$

- Training set of size 300 and validation set of size 500
- $\epsilon = \frac{1}{2} \sqrt{\frac{2k+2}{k}} = 0.866$
- $\delta = 10^{-5}$
- 10 random samples



Comparisons of Error Rates on Simulation Example 2

| Method | Error Rates | |
|------------------------------------|---------------|---------------|
| | Training | Testing |
| LDA | 0.121 (0.006) | 0.191 (0.006) |
| QDA | 0.039 (0.004) | 0.205 (0.006) |
| CART | 0.072 (0.003) | 0.289 (0.004) |
| FDA/MARS (1 df) | 0.100 (0.006) | 0.191 (0.006) |
| FDA/MARS (2 df) | 0.068 (0.004) | 0.215 (0.002) |
| MDA (3 subclasses) | 0.087 (0.005) | 0.169 (0.006) |
| MDA (3 subclasses, penalized 4 df) | 0.137 (0.006) | 0.157 (0.005) |
| PDA (penalized 4 df) | 0.150 (0.005) | 0.171 (0.005) |
| KNN (5) | 0.130 (0.004) | 0.193 (0.007) |
| OVR | 0.055 (0.005) | 0.172 (0.007) |
| Random Forest | 0.037 (0.003) | 0.330 (0.011) |
| VDA ($\lambda = 0.001$) | 0.099 (0.005) | 0.155 (0.008) |
| VDA ($\lambda = 0.01$) | 0.105 (0.006) | 0.162 (0.006) |
| VDA ($\lambda = 0.1$) | 0.097 (0.005) | 0.163 (0.005) |
| VDA ($\lambda = 1$) | 0.098 (0.003) | 0.163 (0.003) |
| VDA ($\lambda = 10$) | 0.095 (0.005) | 0.159 (0.006) |
| VDA ($\lambda = 100$) | 0.096 (0.004) | 0.149 (0.008) |
| VDA ($\lambda = 1000$) | 0.125 (0.007) | 0.138 (0.006) |

Speed of the MM Algorithm

The waveform data from UCI data repository, with the training set of size 300 and the validation set of size 4700

Numerical and Statistical Performance of Different Discriminant Methods

| Method | Testing Error | Iterations | Time |
|-----------------------|---------------|------------|-------|
| VDA w/o step doubling | 0.1396 | 236 | 1.300 |
| VDA w/ step doubling | 0.1340 | 79 | 0.440 |
| LDA | 0.1800 | NA | 0.750 |
| QDA | 0.1928 | NA | 0.800 |
| KNN (5) | 0.1934 | NA | 0.660 |
| OVR | 0.1606 | NA | 4.610 |
| MSVM | NA | NA | NA |
| CART | 0.2864 | NA | 0.490 |
| Random Forest | 0.3149 | NA | 3.040 |

Discussion

Discussion

- Multicategory discriminant analysis is attractive because it avoids laborious pairwise comparisons of the different categories and creates a clear decision rule in a single computation
- Parameterizing the different categories by the vertices of a regular simplex is a good way of enforcing parsimony
- ϵ -insensitive loss function captures several of the criteria for reliable discrimination
- The low testing error of VDA across a variety of data sets is impressive
- The MM algorithm implementing VDA is relatively easy to program, numerically stable, and responds well to step doubling
- Although its rate of convergence is hard to evaluate theoretically, the overall speed of the MM algorithm is certainly acceptable on practical problems of moderate size

Discussion

Good performance of VDA is culmination of several factors

- Insensitive to outliers
- Driven by support vectors
- Not prone to overfitting
- Regularized estimation on problems with fewer cases than parameters
- Reasonably scaled to problems with large numbers of cases or categories
- Parsimonious in its number of parameters

Recommended Tactics

- Standardize all predictors to have mean 0 and variance 1
- Expand the number of features in discriminant problems to encourage nonlinear decision boundaries
- Truncate case weights. Choosing δ in the range $[10^{-7}, 10^{-3}]$ leads to good results
- Choose λ by cross validation
- Define ϵ by

$$\epsilon = \frac{1}{2} \sqrt{\frac{2k+2}{k}}$$

the largest possible value avoiding overlap of the ϵ -insensitive spheres around each vertex of the regular simplex

Future Work

- ϵ -insensitive loss and penalty based on L_1 norm
- Application to underdetermined cases where $p \gg n$, e.g. cancer subtype classification based on microarray data
- Statistical behavior and consistency of VDA

Thank you very much!