

Distance Weighted Discrimination &
Geometrical Representation of HDLSS data

by

J. S. Marron¹, Peter Hall² and Michael Todd³

¹Department of Statistics
University of North Carolina

²Centre for Mathematics and its Applications
Australian National University

³School of Operations Research and Industrial Engineering
& Department of Computer Science - Cornell University

A Brief Advertisement



[Go to SAMS I Ad](#)

Quick Web Access: type SAMS I in google.com

Main Points of Talk

1. An “Improvement” of SVM, in a *really new direction*
2. An “alternate” approach to kernel methods:
“Implicit” vs. “Explicit”
3. Large dimension asymptotics
“Geometric Representation”

Some Personal Viewpoints

- “Interpretation” is higher priority than “Error Rate”
- “Direction Vectors” are useful
(so often prefer linear methods)
- Prefer “original” optimization problems for insights
(not penalized RKHS form, not dual forms)
- Prefer *wide range* of asymptotic analyses

Important Data Analytic Context

High Dimension Low Sample Size $d \gg n$

(Personal) driving problems:

1. Medical imaging
 d high 10s – 100s, n 20s – 100s
2. Micro-arrays measuring gene expression
 d 100s – 10,000s, n 10s – 100s
3. Chemometric spectra
 d 1,000s, n 10s

A Suggestion on Terminology

Let's replace "small n , large p " with

HDLSS

Because only statisticians understand "p"

- Leaves others wondering
- Stands for "parameters" (in linear regression)?
- And don't really have so many parameters anyway...

A real data example

Genetic Micro-Arrays (thanks to C. M. Perou, et. al.):

Measures “expression” (activity) of many genes at once

Current Problem: “Batch effects” ($n = 49, d = 2,452$)

(caused by different production runs, **g**, **h**, **j**)

Visualization of Problem: [PCA and 2-d scatterplots of projns](#)

- Serious problem, likely to affect subsequent analysis
- How to correct?

Batch Effect Adjustment

“Standard Approach”: PCA (i.e. SVD), based on PC1

- Works well when PC1 is “in that direction” ([Toy e.g.](#))
(recall PC1 is in “direction of greatest variation”)
- Otherwise (e.g. here) quite doubtful

Linear Model (+ Random Effects) Approaches

- “Interpretability”? (followed by exploratory data analysis??)

Proposed “New” Approach: Use discrimination methods

Basics of Discrimination (Classification)

Two Class (Binary) Version:

- Using “Training Data” from **Class +1**, and from **Class –1**
- Develop a “Rule”, for assigning new data to a Class

Canonical Example: Disease Diagnosis

- New patients are either “healthy” or “ill”
- Determine on basis of measurements
- Based on preceding experience (training data)

Quick Overview of Discrimination

Toy Graphic i.i.d. $N(\mu, I)$, $\mu_{1,\pm} = \pm 2.2$, $n = 40$, $d = 50$

Classical Attempt: Fisher Linear Discrimination

Modern Approaches:

Support Vector Machine ([toy graphic illustration](#))

Distance Weighted Discrimination

- Idea: “feel all of the data”, not just “support vectors”
- Requires more complicated optimization

Distance Weighted Discrimination

Based on Optimization Problem:

$$\max_{w, \beta} \sum_{i=1}^n \frac{1}{r_i}$$

More precisely: Work in appropriate penalty for violations

Same effect achievable by more conventional SVM methods???

- NOT by “squaring hinge loss” (or any “margin” method???)
- since *no effect* for [separable data](#) (common for **HDLSS**)

Distance Weighted Discrimination (cont.)

Optimization Method: Second Order Cone Programming

- “Still convex” generalization of quadratic programming
- Allows fast greedy solution
- Can use available fast software
- Mike Todd & Students package: SDPT3
- Type into Google, to obtain paper

Distance Weighted Discrimination (cont.)

Performance in [Toy Example](#):

- Clearly superior to FLD and SVM
- Very competitive with Mean Difference
- Solves above problem?
- Looks “better” to statisticians
- Expect “better generalization”
- Since “less driven by noise”

Application to Batch Effect Data

SVM Adjustment

- Looks reminiscent of above problem
- 2nd application to residuals still has gap?
- Must, since **HDLSS**, but “perhaps very small”?

DWD Adjustment

- Again reminiscent of above example
- 2nd application to residuals looks great!

Application to Batch Effect Data (cont.)

Careful: used different criteria for assessment

[SVM adjustment, DWD assessment](#)

- Now looks like similar results
- Reason for this? “Geometrical Representation”

Final result: [Adjusted 2-d Scatterplots](#)

- Applied Stepwise: 1. **g** vs. **h & j**, 2. **h** vs. **j**
- Great “mixing” of batches, i.e. successful adjustment

Application to “Outcomes”

Breast Cancer Study (C. M. Perou):

Outcome of interest = death or survival

Approach:

- Treat death vs. survival during study as “classes”
- Find “direction that best separates the classes”

[DWD Projection](#) & [SVM Projection](#)

- SVM is “better separated”?
- DWD gives “more spread between sub-populations”???

Application to “Outcomes” (cont.)

Which is “better”?

Approach: Find “genes” (i.e. coordinates) “of interest”?

Show “intensity plot” of “top 20 in each direction”:

[SVM top-bottom 20](#) & [DWD top-bottom 20](#)

- DWD finds genes showing better separation
- SVM genes are less informative

Application to “Outcomes” (cont.)

How about Centroid (Mean Difference) Method?

MD Projection & MD top-bottom 20

- Best yet, in terms of red – green plot?
- Projections unacceptably mixed?
- These are two *different* goals...
- Try for trade-off? Scale space approach???

Interesting philosophical point: very simple things often “best”

DWD vs. SVM Simulations

3 simulations: [Dist'n 1](#) [Dist'n 2](#) [Dist'n 3](#)

- Shows each method is sometimes best
- DWD is “usually near best” (i.e. “good overall”)
- Note: all are closer together for higher $d = 1600$
- Explanation: **Geometrical Representation**

Fisher Consistency of DWD?

As formulated in:

Lin, Wahba, Zhang, and Lee (2002) Statistical Properties and Adaptive Tuning of Support Vector Machines, *Machine Learning*, 48, 115-136.

Work in progress by Helen Zhang

Aside on SVM Computation

Important Note: available algorithms are *not* created equal

Toy Example:

- [Gunn's Matlab code](#)
- [Todd's Matlab code](#)

Other solid implementations:

- Clint Scovel & LANL group
- Others???

Aside on Embedding

Motivation for Support Vector Machine idea???

Key Reference:

Aizerman, Braverman and Rozoner (1964) *Automation and Remote Control*, 15, 821-837.

Toy Example: [{Donut data}](#)

Separate with a linear (separating plane) method?

Polynomial Embedding

Key Idea: embed data in *higher dimensional space*,

then apply linear methods for *better separation*

E.g. Replace data

$$\begin{pmatrix} X_{1,1} \\ \vdots \\ X_{1,d} \end{pmatrix}, \dots, \begin{pmatrix} X_{n,1} \\ \vdots \\ X_{n,d} \end{pmatrix}$$

by

$$\begin{pmatrix} X_{1,1} \\ \vdots \\ X_{1,d} \\ X_{1,1}^2 \\ \vdots \\ X_{1,d}^2 \end{pmatrix}, \dots, \begin{pmatrix} X_{n,1} \\ \vdots \\ X_{n,d} \\ X_{n,1}^2 \\ \vdots \\ X_{n,d}^2 \end{pmatrix}$$

Polynomial Embedding (cont.)

Practical Effect:

- Maps data to high dimensional manifold
- Which can be “better sliced” by linear discriminators

[Donut Data Example:](#) Major success,

since $X_1^2 + X_2^2$ found by linear method in embedded space

Kernel Embedding (cont.)

Other types of embedding:

- Sigmoid functions (ala neural networks)
- Radial Basis Functions (a.k.a. Gaussian Windows)

Toy Data: [Checkerboard](#)

- (low degree) [polynomials](#) fail
- [Gaussian Windows](#) are excellent

SVMs, Computation & Embedding

For an “Embedding Map”, $\Phi(x)$ e.g. $\Phi(x) = \begin{pmatrix} x \\ x^2 \end{pmatrix}$

Explicit Embedding (following original idea):

Maximize: $L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \Phi(x_i) \cdot \Phi(x_j)$

Get classification function: $f(x) = \sum_{i=1}^n \alpha_i y_i \Phi(x) \cdot \Phi(x_i) + b$

- Straightforward application of embedding idea
- But loses inner product advantage

SVMs, Computation & Embedding (cont.)

Implicit Embedding:

$$\text{Maximize: } L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \Phi(x_i \cdot x_j)$$

$$\text{Get classification function: } f(x) = \sum_{i=1}^n \alpha_i y_i \Phi(x \cdot x_i) + b$$

- Still defined only in terms of “inner products”
- Retains optimization advantage
- Thus used very commonly
- Comparison to explicit embedding? Which is “better”???

Tuning Parameter Choice

On “weight for violations”. Serious issue [{Toy Example}](#)

Machine Learning Approach:

Complexity Theory Bounds

(Interesting theory, but questionable practicality)

Wahba School:

Generalized Cross-Validation

Personal suggestion:

Scale Space Approach: “try them all” [{Toy Example}](#)

Gaussian Kernel Window Width

Example: [Target Toy Data](#)

Explicit Gaussian Kernel Embedding:

sd = 0.1

sd = 1

sd = 10

sd = 100

- too small → poor generalizability
- too big → miss important regions
- surprisingly broad “reasonable region”???

Gaussian Kernel Window Width (cont.)

Example: [Target Toy Data](#) (cont.)

Implicit Gaussian Kernel Embedding:

[sd = 0.1](#)

[sd = 0.5](#)

[sd = 1](#)

[sd = 10](#)

- Similar “large – small” lessons
- Seems to require smaller range for “reasonable results”
- Much different “edge behavior”
- Interesting questions for future investigation...

Some Simple “Paradoxes” of HDLSS data

For d dimensional “Standard Normal” distribution:

$$\underline{Z} = \begin{pmatrix} Z_1 \\ \vdots \\ Z_d \end{pmatrix} \sim N(\underline{0}, I)$$

Euclidean Distance to Origin (as $d \rightarrow \infty$):

$$\|\underline{Z}\| = \sqrt{d} + O_p(1)$$

- Data lie roughly on surface of sphere of radius \sqrt{d}
- Yet origin is point of “highest density”???
- Paradox resolved by “density w. r. t. Lebesgue Measure”

Some Simple “Paradoxes” of HDLSS data (cont.)

For d dim’al “Standard Normal” dist’n:

$$\underline{Z}_1 \text{ indep. of } \underline{Z}_2 \sim N(\underline{0}, I)$$

Euclidean Distance between \underline{Z}_1 and \underline{Z}_2 (as $d \rightarrow \infty$):

$$\|\underline{Z}_1 - \underline{Z}_2\| = \sqrt{2d} + O_p(1)$$

- Distance tends to *non-random* constant
- Can extend to $\underline{Z}_1, \dots, \underline{Z}_n$
- Where do they all go?? (we can only perceive 3 dim’ns)

Some Simple “Paradoxes” of HDLSS data (cont.)

For d dim’al “Standard Normal” dist’n:

$$\underline{Z}_1 \text{ indep. of } \underline{Z}_2 \sim N(\underline{0}, I)$$

High dim’al Angles (as $d \rightarrow \infty$):

$$\text{Angle}(\underline{Z}_1, \underline{Z}_2) = 90^\circ + O_p\left(\frac{1}{\sqrt{d}}\right)$$

- “Everything is orthogonal”???
- Where do they all go??? (again our perceptual limitations)
- Again 1st order structure is *non-random*

Geometrical Representation of HDLSS data

Assume $Z_1, \dots, Z_n \sim N(0, I)$, $d \gg n$, asymptotics as $d \rightarrow \infty$

1. Study Subspace Generated by Data
 - a. Hyperplane through 0, of dimension n
 - b. Points are “nearly equidistant to 0”, & $\text{dist} \sim \sqrt{d}$
 - c. Within plane, can “rotate towards $\sqrt{d} \times$ Unit Simplex”
 - d. *All Gaussian data sets are “near U. Simplex vertices”!!!*
 - e. “Randomness” *appears only in rotation of simplex*

[Two Point Toy Example](#)

Geometrical Representation of HDLSS data (cont.)

Assume $Z_1, \dots, Z_n \sim N(0, I)$, $d \gg n$, asymptotics as $d \rightarrow \infty$

2. Study Hyperplane Generated by Data

- a. $n - 1$ dimensional hyperplane
- b. Points are pairwise equidistant, $\text{dist} \sim \sqrt{2d}$
- c. Points lie at vertices of $\sqrt{2d} \times$ “regular $n - \text{hedron}$ ”
- d. Again “randomness in data” is *only in rotation*
- e. Surprisingly rigid structure in data?

[Three Point Toy Example](#)

Geometrical Representation of HDLSS data (cont.)

Simulation View: shows “rigidity after rotation”

Straightforward Generalizations:

- non-Gaussian data: only need moments
- non-independent: use “mixing conditions”
- ⋮

All based on simple “Laws of Large Numbers”

Geometrical Representation of HDLSS data (cont.)

Explanation of Observed Behavior (Batch Effect & Simulations):

Recall “everything similar for very high d ”

- 2 popn's are 2 simplices (i.e. regular n -hedrons)
- everything is the same distance from the other class
- i.e. everything is a support vector
- i.e. all sensible directions show “data piling”
- so “sensible methods are all nearly the same”

Further Consequences of Geometric Representation

1. Inefficiency of DWD for uneven sample size
(motivates “weighted version”, work in progress)
2. DWD more “stable” than SVM
(based on “deeper limiting distributions”)
(reflects above intuition about “feeling sampling variation”)
(something like “mean vs. median”)

The Future of Geometric Representations?

- HDLSS versions of nice optimality results?
- “Contiguity” approach? (parameters depending on d)
- Rates of convergence?
- Suggest approaches to improving DWD?
(other functions of distance besides inverse?)

It is still early days ...

Interesting asymptotic domains for future work:

- a. Classical: d fixed, n grows
- b. d and n grow (wide range of different relationships)
- c. HDLSS: n fixed, d grows

All are of interest, and anticipate different lessons

Some Carry Away Lessons

- HDLSS contexts are worth more study
- DWD better than SVM for HDLSS data
- Let's pay more attention to optimization
- “Randomness” in HDLSS data is *only rotations*
- Modulo random rotation, have “constant simplex shape”