An overview of Support Vector Machines and Kernel Methods

by J. S. Marron

SAMSI &
Department of Statistics
University of North Carolina

With a lot of help from:

Jeongyoun Ahn, UNC

Helen Hao Zhang, NCSU

A Brief Advertisement

samsi

NSF • Duke • NCSU • UNC • NISS

[Go to SAMSI Ad]

Quick Web Access: type SAMSI in google.com

Kernel Methods & Support Vector Machines

Several Viewpoints:

- Historical
- Statistical
- Optimization
- Machine Learning
- Big Picture: Classification, i. e. Discrimination

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)

Discrimination (Classification) (cont.)

Important Methods:

- Fisher Linear Discrimination

  (nonparametric method! Gaussian "requirement" is a common misconception)

- Nearest Neighbor Methods

- Neural Networks

- …

Discrimination (Classification) (cont.)

Interesting Reference:

Duda, Hart & Stork (2001) *Pattern Classification*, Wiley.

- 2^{nd} Edition of classic book Duda & Hart (1973)
- Uses neural networks as "the language"
- Elegant mathematical framework
- Intuitive content???
- Fisher Linear Discrimination as a neural net?

Discrimination (Classification) (cont.)

A Dichotomy of Methods:

I. “Direction” Based

- Fisher Linear Discrimination [toy example]
- Support Vector Machines

II. Black Box

- Nearest Neighbor Methods
- Neural Networks

Direction Oriented Methods

Useful for more than “misclassification error rate”

E.g. Micro-arrays:

- Bias Adjustment {before} {after}

- Gene Insights {outcome data}

Polynomial 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 (cont.)

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

then apply linear methods for *better separation*

E.g. Replace data $\begin{pmatrix} X_{1,1} \\ \vdots \\ X_{1,d} \end{pmatrix}, \cdots, \begin{pmatrix} X_{n,1} \\ \vdots \\ X_{n,d} \end{pmatrix}$ by $\begin{pmatrix} X_{1,1} \\ \vdots \\ X_{1,d} \\ X_{1,1}^2 \\ \vdots \\ X_{1,d}^2 \end{pmatrix}, \cdots, \begin{pmatrix} X_{n,1} \\ \vdots \\ X_{n,d} \\ X_{n,1}^2 \\ \vdots \\ X_{n,d}^2 \end{pmatrix}$

Polynomial Embedding (cont.)

Practical Effect:

- Maps data to high dim'al manifold
- Which can be "better sliced" by linear discriminators

Toy Examples in 1-d: 1 break, 2 breaks, 3 breaks

Embedding creates richer discrimination regions

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

Support Vector Machines

Early References:

Vapnik (1982) *Estimation of dependences based on empirical data*, Springer (Russian version, 1979).

Vapnik (1995) *The nature of statistical learning theory*, Springer.

Motivation???:

- Find a linear method that "works well" for embedded data
- Note: embedded data are *very* non-Gaussian
- Suggests value of "really new approach"

SVMs (cont.)

Graphical View {Toy Example}:

- Find “separating plane”
- To maximize “distance from data to plane”
- In particular “smallest distance”
- Data points closest are called “support vectors”,
- Gap between is called “margin”

SVMs, Optimization View

Setup Optimization problem, based on:

- Data (feature) vectors $x_1, \ldots, x_n$
- Class Labels $y_i = \pm 1$
- Normal Vector w
- Location (determines intercept) b
- Residuals (right side) $r_i = y_i \left(x_i{}^t w + b \right)$
- Residuals (wrong side) $\xi_i = -r_i$
- Solve (convex problem) by quadratic programming

SVMs, Optimization View (cont.)

Lagrange Multipliers "primal" formulation (separable case):

Minimize: $$L_P(w,b,\alpha)=\tfrac{1}{2}\|w\|^2-\sum_{i=1}^{n}\alpha_i\left(y_i\left(x_i\cdot w+b\right)-1\right)$$

Where $\alpha_1,\ldots,\alpha_n>0$ are Lagrange multipliers

Dual Lagrangian version:

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

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

SVMs, Computation

Major Computational Point:

- Only depends on data through inner products!
- Thus enough to “only store inner products”
- Creates savings in optimization
- But creates variations in “kernel embedding”

SVMs, Computation & Embedding

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

Explicit Embedding:

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

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:

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

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

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

SVMs, Computation (cont.)

Caution: available algorithms are *not* created equal

Toy Example:

- Gunn's Matlab code
- Todd's Matlab code

Distance Weighted Discrimination

Variation of SVM for High Dimension, Low Sample Size Data

Toy Example $d = 50$, $N(0,1)$, but $\mu_1 = \pm 2.2$, $n_+ = n_- = 20$.

1. Fisher Linear Discrimination

   - Gives "perfect separation"
   - But grossly overfits
   - Results in poor generalizability

Distance Weighted Discrimination (cont.)

2. SVM, better results

   - Much more stable than FLD

   - But still have“piling at margin”, somewhat like FLD

   - “feels support vectors” too strongly?

   - Possible to improve?

DWD idea: Replace “minimum distance” by “average”

I.e. optimization “feels all of the data”

Distance Weighted Discrimination (cont.)

Based on Optimization Problem:

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

More precisely: Work in appropriate penalty for violations

Optimization Method: Second Order Cone Programming

- "Still convex" generalization of quadratic programming
- Allows fast greedy solution
- Can use available fast software

Distance Weighted Discrimination (cont.)

Performance in {Toy Example}:

- Clearly superior to FLD and SVM
- Smallest "angle to optimal"
- Gives best generalizability performance
- Projected dist'ns have "reasonable Gaussian shapes"

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}

Tuning Parameter Choice

Key GCV Type References:

Wahba, Lin and Zhang (2000) Generalized Approximate Cross Validation for Support Vector Machines, or, Another Way to Look at Margin-Like Quantities, *Advances in Large Margin Classifiers*, Smola, Bartlett, Scholkopf and Schurmans, eds., MIT Press (2000), 297-309.

Wahba, Lin, Lee, and Zhang (2002) Optimal Properties and Adaptive Tuning of Standard and Nonstandard Support Vector Machines, *Nonlinear Estimation and Classification,* Denison, Hansen, Holmes, Mallick and u, eds, Springer, 125-143.

Joachims (2000) Estimating the generalization performance of a SVM efficiently.*Proceedings of the International Conference on Machine Learning*, San Francisco, 2000. Morgan Kaufman.

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…

Robustness

Toy Example

- Single point generates huge changes in SVM direction
- Clearly not "robust" in classical sense
- But all are "pretty good" for classification
- I.e. will give good "generalizability" over many directions

Multi-Class SVMs

Lee, Y., Lin, Y. and Wahba, G. (2002) "Multicategory Support Vector Machines, Theory, and Application to the Classification of Microarray Data and Satellite Radiance Data", U. Wisc. TR 1064.

- So far only have “implicit” version

- “Direction based” variation is unknown

“Feature Selection” for SVMs

Idea: find a few “important” components of data vector”

e.g. “finding important genes” in micro-array analysis.

Key Reference:

Bradley and Mangasarian (1998) Feature selection via concave minimization and support vector machines, *Machine Learning Proceedings of the Fifteenth International Conference(ICML '98)*, J. Shavlik, ed., pages 82-90. Morgan Kaufmann.

Additional Information

Recommended Introductions ("Tutorials")

Burges (1998) A Tutorial on Support Vector Machines for Pattern Recognition, *Knowledge Discovery and Data Mining*, 2.

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

Favorite Web Pages:

Kernel Machines Web Page: http://www.kernel-machines.org/

Wahba Web Page: http://www.stat.wisc.edu/~wahba/trindex.html

Additional Information (cont.)

Books:

Good (?) Starting point:

Cristianini and Shawe-Taylor (2002) *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, UK.

Good Complete Treatment:

Schölkopf and Smola (2002) *Learning with Kernels*. MIT Press, Cambridge, MA.

Disclaimer

This was a *personal* overview

Other approaches to SVMs: *completely* different

Machine Learners:

Complexity Theory & Optimization

Wahba & Co:

Optimization in Reproducing Kernel Hilbert Spaces

Simulation Comparisons

Geometric Representation