Statistical Learning Methods for Emerging Database Applications

Edward Chang
Associate Professor,
Electrical Engineering, UC Santa Barbara
CTO, VIMA Technologies

Useful Links

- Related Publications
  - http://www-db.stanford.edu/~echang/
- Software Free Trial
  - http://www.imagebeagle.com
  - Locate objectionable images on your hard drives
  - Before your boss finds it!!!

Outline

- Statistical Learning
- Emerging Applications Data Characteristics
- Classical Models
- Kernel Methods
  - Linear Model View
  - Nearest Neighbor View
  - Geometric View
- Dimension Reduction Methods

Statistical Learning

- Program the computers to learn!
- Computers improve performance with experience at some task
- Example:
  - Task: playing checkers
  - Performance: % games it wins
  - Experience: expert players

Statistical Learning

- Task $\hat{Y} = f(U)$
  - Represented by some model(s)
  - Implies hypothesis
- Performance
  - Measured by error functions
- Experience (L)
  - Characterized by training data
- Algorithm (Φ)

Supervised Learning

- X: Data
  - U: Unlabeled pool
  - L: Labeled pool
- G: Labels
  - Regression
  - Classification
- Φ: Learning algorithm
- $f = \Phi(L)$
- $\hat{Y} = f(U)$

Learning Algorithms

- Linear Model
- K-NN
- Neural Networks
- Decision Trees
- Kernel Methods
- Etc.

Classical Model

- N:Number of training instances
- N^+, N^-
- D:Dimensionality
- $N \gg D$ $N \rightarrow \infty$
  - E.g., PAC learnability
- $N^- \approx N^+$

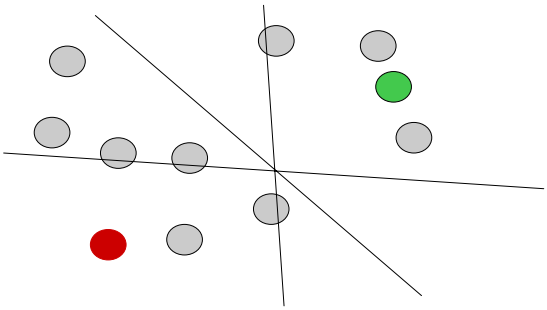
Emerging DB Applications

- $N < D$
- $N^+ << N^-$
- Examples
  - Information Retrieval with relevance feedback
  - Gene Profiling

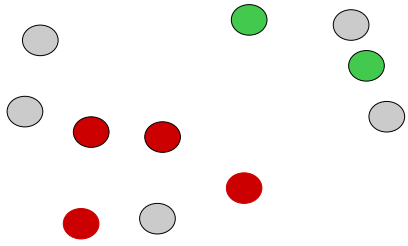
Image Retrieval Demo

- $N < D$
  - $N < 50$
  - $D = 150$
- $N^+ << N^-$
- ACM SIGMOD 01; ACM MM 01,02; IEEE CVPR 03

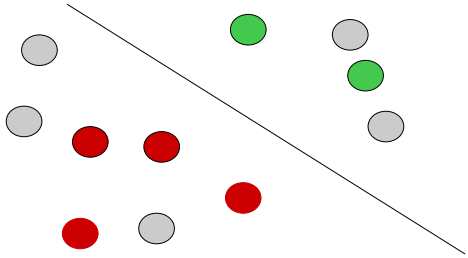
SVMactive



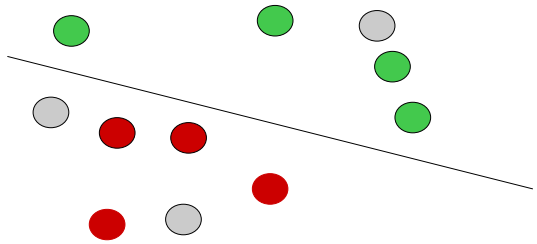
SVMactive



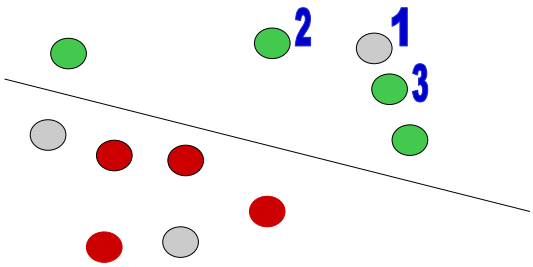
SVMactive



SVMactive

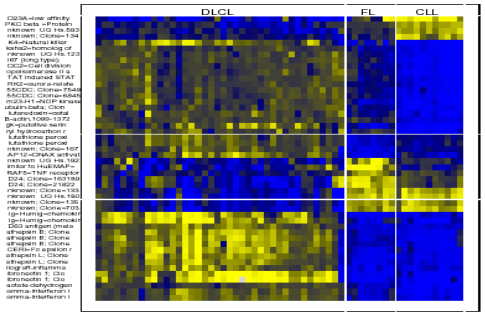


Ranking



Gene Profiling Example

N = 59 cases, D = 4026 genes



Outline

- Statistical Learning
- Emerging Applications Data Characteristics
- Classical Models (Classification)
- Kernel Methods
  - Linear Model View
  - Nearest Neighbor View
  - Geometric View
- Dimension Reduction Methods

Linear Model

- $\underline{Y} = \beta_0 + \Sigma\ \beta_j X_j$ (j = 1 to p)
- $\underline{Y} = X^T\beta$
- $RSS(\beta) = (y - X\beta)^T(y - X\beta)$
  - RSS: Residual Sum of Square
- $\beta = (X^TX)^{-1}X^Ty$

Linear Model

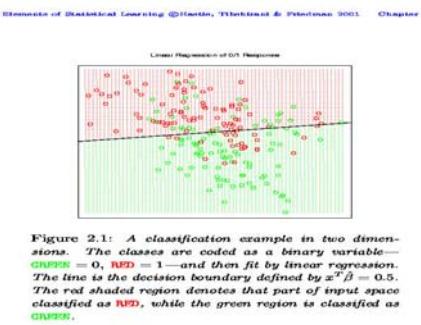


Figure 2.1: *A classification example in two dimensions. The classes are coded as a binary variable— GREEN = 0, RED = 1—and then fit by linear regression. The line is the decision boundary defined by $x^T\beta = 0.5$. The red shaded region denotes that part of input space classified as RED, while the green region is classified as GREEN.*

Maximum Likelihood

- $\underline{Y} = \beta_0 + \Sigma\ \beta_j X_j$ (j = 1 to p)
- $\underline{Y} = X^T\underline{\beta}$
- $\underline{Y} = X^T\underline{\beta} + \varepsilon$
  - ε (noise signals) are independent
  - $\varepsilon \rightarrow N(0, \partial^2)$
- $P(y|\beta x)$ has a normal dist. with
  - Mean at $y = \beta x$
  - Variance ∂^2

Linear Model

- $P(y|\beta x) \rightarrow N(0, \partial^2)$
- Training
  - Given $(x_1,y_1)\ (x_2,y_2)\ \ldots\ (x_n,y_n)$
  - Infer $P(\beta \mid x_1, x_2, \ldots x_n, y_1, y_2, \ldots y_n)$
  - By Bayes rule, or
  - Maximum Likelihood Estimate

Maximum Likelihood

- For what β is
  - $P(y_1, y_2, \ldots y_n \mid x_1, x_2, \ldots x_n, \beta)$ maximized?
  - $\Pi\ P(y_i|\beta x_i)$ maximized?
  - $\Pi\ \exp(-\frac{1}{2}(y_i-\beta x_i/\partial)^2)$ maximized?
  - $\Sigma\ (-\frac{1}{2}(y_i-\beta x_i/\partial)^2$ maximized?
  - $\Sigma\ (y_i-\beta x_i)^2$ minimized?

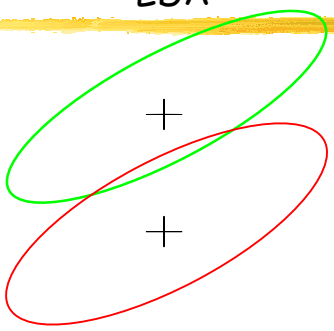
Least Square Linear Model

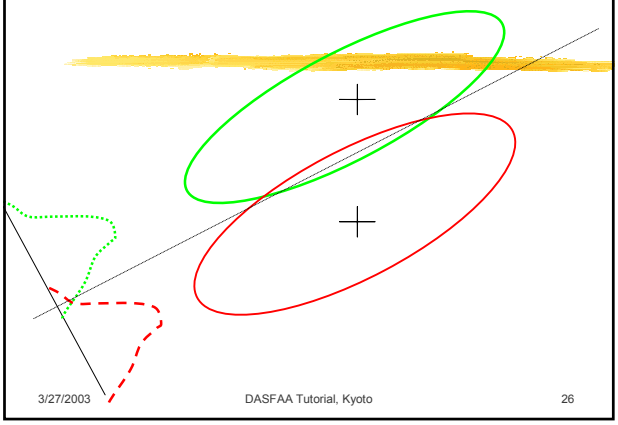
- Solution Method #1
  - $RSS(\beta) = (y - X\beta)^T(y - X\beta)$
  - $\underline{\beta} = (X^TX)^{-1}X^Ty$
- Solution Method #2 (for D > N)
  - Gradient decent
  - Perceptron

Other Linear Models

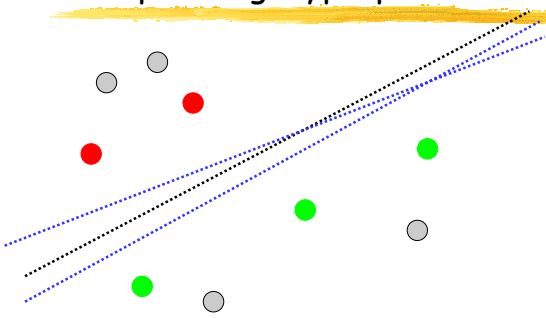
- LDA
  - Find the projection direction which minimizes the overlap for two Gaussian distributions
- Separating Hyperplane

LDA

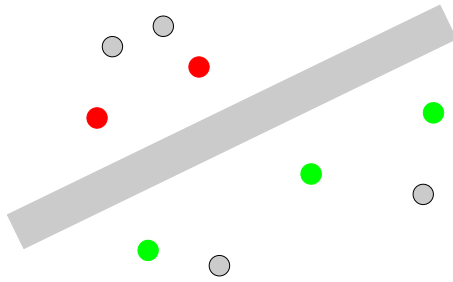




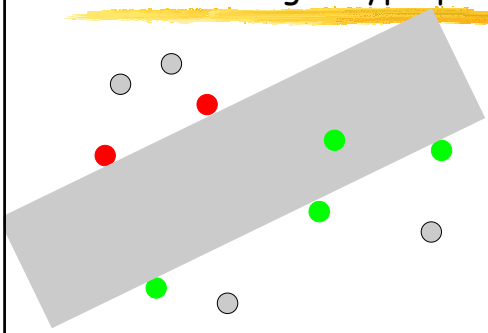
Separating Hyperplane



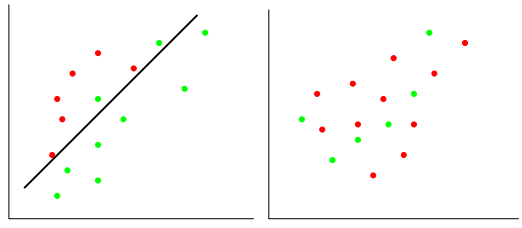
Separating Hyperplane



Maximum Margin Hyperplane



Linear Model Fits All Data?



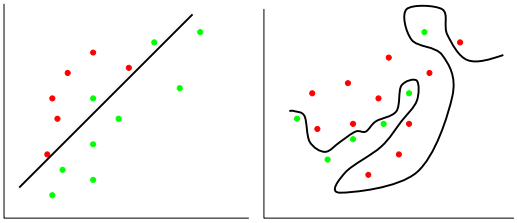
How about Joining the Dots?

- $\underline{Y}(x) = 1/k \sum y_i$,
  - $x_i \in N_k(x)$
- $K = 1$

Linear Models

- $N \geq D$
  - Least Square
  - LDA
- $D > N$
  - Perceptron
  - Maximum Hyperplane

Linear Model Fits All?



NN with k = 1

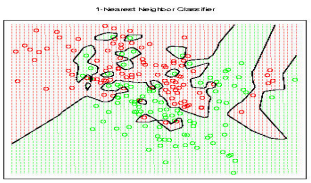


Figure 2.3: *The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable (GREEN = 0, RED = 1), and then predicted by 1-nearest-neighbor classification.* [T. Hastie, etc. 2001]

Nearest Neighbor

- Four Things Make a Memory Based Learner
  - A distance function
  - K: number of neighbors to consider?
  - A weighted function (optional)
  - How to fit with the local points?

Problems

- Fitting Noise
- Jagged Boundaries

Solutions

- Fitting Noise
  - Pick a Larger K?
- Jagged Boundaries
  - Introducing Kernel as a weighting function

NN with k = 15

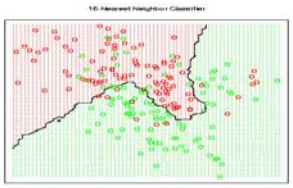
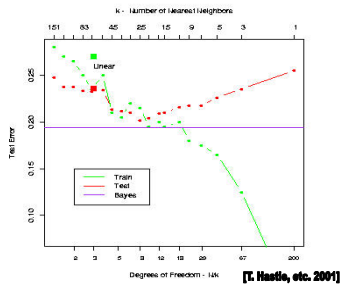


Figure 2.2: *The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable (GREEN = 0, RED = 1) and then fit by 15-nearest-neighbor averaging as in (2.8). The predicted class is hence chosen by majority vote amongst the 15-nearest neighbors.* [T. Hastie, etc. 2001]

NN



[T. Hastie, etc. 2001]

Solutions

- Fitting Noise
  - Pick a larger K?
- Jagged Boundaries
  - Introducing Kernel as a weighting function

Nearest Neighbor -> Kernel Method

- Four Things Make a Memory Based Learner
  - A distance function
  - K: number of neighbors to consider? All
  - A weighted function: RBF kernels
  - How to fit with the local points? Predict weights

Kernel Method

- RBF Weighted Function
  - Kernel width holds the key
  - Use cross validation to find the "optimal" width
- Fitting with the Local Points
  - Where NN meets Linear Model

LM vs. NN

- Linear Model
  - f(x) is approximated by a global linear function
  - More stable, less flexible
- Nearest Neighbor
  - K-NN assumes f(x) is well approximated by a locally constant function
  - Less stable, more flexible
- Between LM and NN
  - The other models…

Decision Theories

- Bias & Variance Tradeoff
- Bayes Prediction
- VC Dimensionality
- PAC Learnability

Variance vs. Bias

- $MSE(x_0) = E_T[f(x_0) - \hat{y}_0]^2$
  $= E_T[\hat{y}_0 - E_T(\hat{y}_0)]^2 + [E_T(\hat{y}_0) - f(x_0)]^2$
- $Error = Var_T(\hat{y}_0) + Bias^2(\hat{y}_0)$

Outline

- Statistical Learning
- Emerging Applications Data Characteristics
- Classical Models (Classification)
- Kernel Methods
- Dimension Reduction Methods

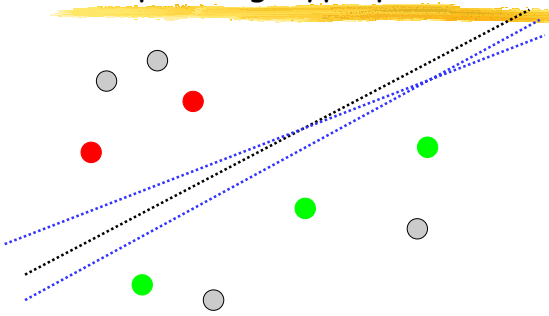
Where Are We and Where Am I Heading To ?

- LM and NN
- Kernel Method of Three Views
  - LM view
  - NN view
  - Geometric view

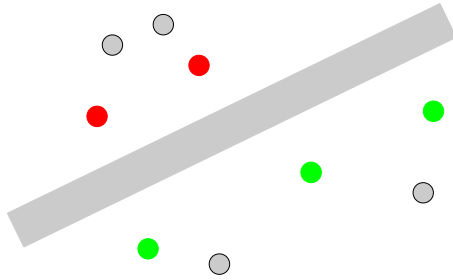
Linear Model View

- $Y = \beta_0 + \Sigma\ \beta X$
- Separating Hyperplane
  - $Max_{||\beta||=1}\ C$
  - Subject to $y_i f(x_i) \geqslant C$, or
  - $y_i (\beta_0 + \beta x_i) \geqslant C$

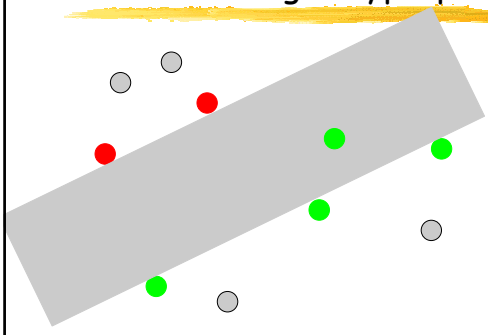
Separating Hyperplane



Separating Hyperplane



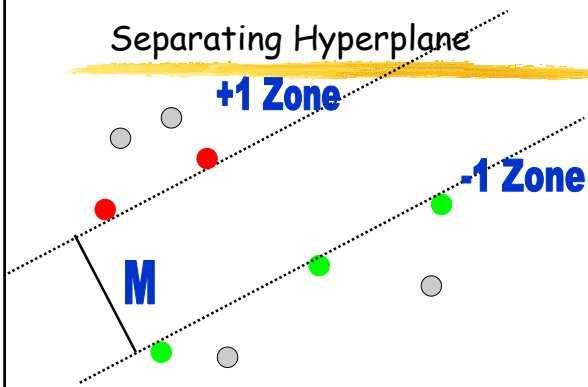
Maximum Margin Hyperplane



Classifier Margin

- Margin
  - Defined as with of the boundary before hitting a data object
- Maximum Margin
  - Tends to minimize classification variance
  - No formal theory for this yet

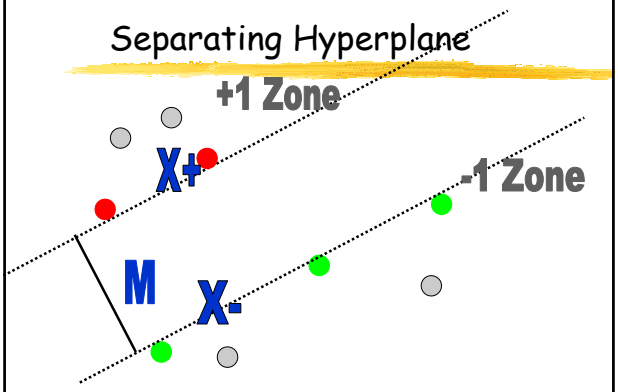
Separating Hyperplane



M's Mathematical Representation

- Plus-plane
  - $\{x: wx+b = +1\}$
- Minus-plane
  - $\{x: wx+b = -1\}$
- $w \perp$ Plus-plane
  - $w(u - v) = 0$, if u and v on plus-plane
- $w \perp$ Minus-plane

Separating Hyperplane



M

- Let x^- be any point on minus-plane
- Let x^+ be the closest plus-plane-point to x^-
- $x^+ = x^- + \lambda w$, why
  - The line $(x^+x^-) \perp$ minus-plane
- $M = |x^+ - x^-|$

M

1. $wx^- + b = -1$
2. $wx^+ + b = 1$
3. $x^+ = x^- + \lambda w$
4. $M = |x^+ - x^-|$
5. $w(x^- + \lambda w) + b = 1$ (from 2 & 3)
6. $wx^- + b + \lambda ww = 1$
7. $\lambda ww = 2$

M

1. $\lambda ww = 2$
2. $\lambda = 2/ww$
3. $M = |x^+ - x^-| = |\lambda w| = \lambda|w| = 2/|w|$

4. Max M
   - Gradient decent, simulated annealing, EM, Newton's method?

Max M

- Max $M = 2/|w|$
- Min $|w|/2$
- Min $|w|^2/2$
  - subject to $y_i(x_iw+b) \geqslant 1$
  - $i = 1,\ldots,N$
- Quadratic criterion with linear inequality constraints

Max M

- Min $|w|^2/2$
  - subject to $y_i(x_iw+b) \geqslant 1$
  - $i = 1,\ldots,N$
- $Lp = \min_{w,b} |w|^2/2 + \sum_{i=1..N} \alpha_i[y_i(x_iw+b)-1]$
- $w = \sum_{i=1..N} \alpha_i y_i x_i$
- $0 = \sum_{i=1..N} \alpha_i y_i$

Wolfe Dual

- $Ld = \sum_{i=1..N} \alpha - 1/2 \sum\sum_{i,j=1..N} \alpha_i \alpha_j y_i y_j x_i x_j$
- Subject to
  - $\alpha_i \geq 0$
  - $\alpha_i [y_i(x_i w+b)-1] = 0$
  - KKT conditions
    - $\alpha_i > 0$, $y_i(x_i w+b) = 1$ (Support Vectors)
    - $\alpha_i = 0$, $y_i(x_i w+b) > 1$

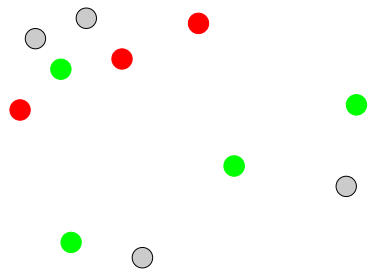
Class Prediction

- $y_q = w\, x_q + b$
- $w = \sum_{i=1..N} \alpha_i y_i x_i$
- $y_q = \text{sign}(\sum_{i=1..N} \alpha_i y_i (x_i \cdot x_q) + b)$

Non-seperatable Classes

- Soft Margin Hyperplane
- Basis Expansion

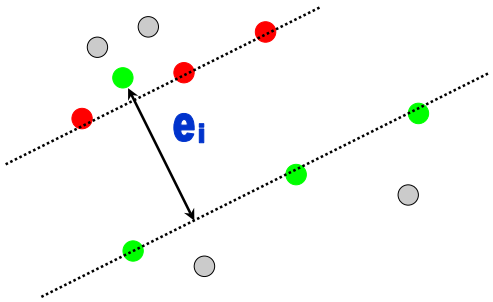
Non-separating Case



Soft Margin SVMs

- Min $|w|^2/2$
  - subject to $y_i(x_i w+b) \geq 1$
  - $i = 1,\ldots,N$
- Min $|w|^2/2 + C \sum \varepsilon_i$
  - $x_i w+b \geq 1 - \varepsilon_i$ if $y_i = 1$
  - $x_i w+b \leq -1 + \varepsilon_i$ if $y_i = -1$
  - $\varepsilon_i \geq 0$

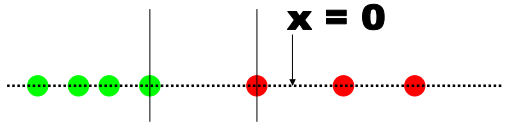
Non-separating Case



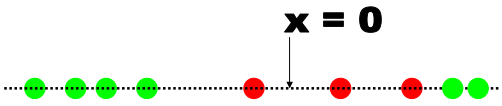
Wolfe Dual

- $Ld = \Sigma_{i=1..N} \alpha - 1/2 \Sigma\Sigma_{i,j=1..N} \alpha_i \alpha_j y_i y_j x_i x_j$
- Subject to
  - $C \geqslant \alpha_i \geqslant 0$
  - $\Sigma \alpha_i y_i = 0$
  - KKT conditions
- $y_q = \text{sign} (\Sigma_{i=1..N} \alpha_i y_i (x_i \cdot x_q) + b)$

Basis Function



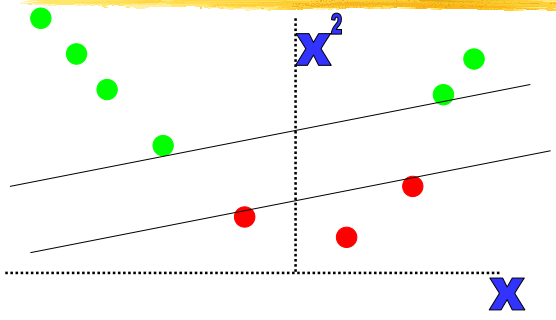
Harder 1D Example



Basis Function

- $\Phi(X) = (x, x^2)$

Harder 1D Example



Some Basis Functions

- $\Phi(X) = \Sigma \gamma_m h_m(X)$
  - $h_m(X)$ $R^p \rightarrow R$
- Common Functions
  - Polynomial
  - Radial basis functions
  - Sigmoid functions

Wolfe Dual

- $Ld = \sum_{i=1..N} \alpha - 1/2 \sum\sum_{i,j=1..N} \alpha_i \alpha_j y_i y_j \Phi(x_i) \Phi(x_j)$
- Subject to
  - $C \geqslant \alpha_i \geqslant 0$
  - $\Sigma \alpha_i y_i = 0$
  - KKT conditions
- $y_q = \text{sign}(\sum_{i=1..N} \alpha_i y_i (\Phi(x_i) \cdot \Phi(x_q)) + b)$
- $K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$
  - Kernel function!

Quadratic Basis Functions

- $\Phi(X) = \{1, x_i, x_i x_j\}$, ij = 1..p
  - (p+1)(p+2) terms
  - P^2 terms
  - $O(P^2)$ computational cost
- It is equivalent to $(x_i x_j + 1)^2$
  - O(p) computational cost
- Total Cost
  - $O(N^2 p)$

Dot Product Saves the Day

- $O(N^2 p)$
- Quadratic
  - $O(N^2 p^2)$
- Cubic
  - $O(N^2 p^3)$
- Quartic
  - $O(N^2 p^4)$

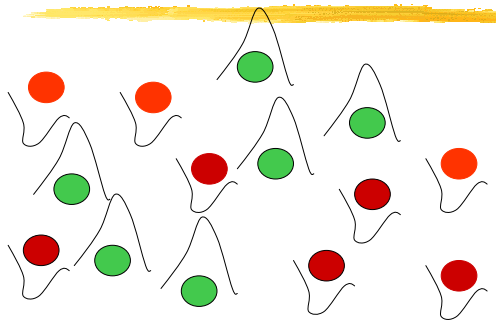
Quiz

- What is a polynomial kernel degree *d* function's signature?
- $(x_i x_j + 1)^d$

Nearest Neighbor View

- Z, a set of zero mean jointly Gaussian random variables,
  - Each Z_i corresponds to one example X_i
  - Cov $(z_i, z_j) = K(x_i, x_j)$
- y_i, the lable of z_i, +1 or -1
  - $P(y_i \mid z_i) = \sigma(y_i, z_i)$

Training Data

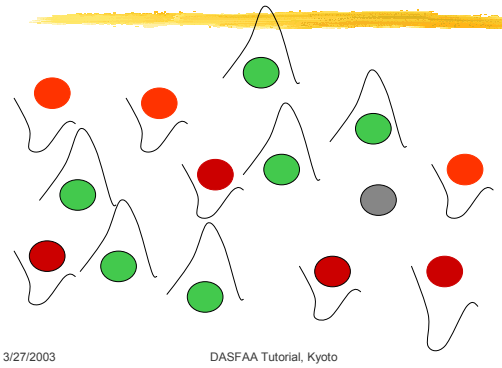


General Kernel Classifier [Jaakkola, etc. 99]

- MAP Classification for x_t
  - $y_t = \text{sign}\,(\Sigma\ \alpha_i\, y_i\, K(x_t, x_i))$
  - $K(x_i, x_j) = \text{Cov}\,(z_i, z_j)$ (some similarity function)
- Supervised Training: Compute α_i
  - Given X and y, and
  - An error function such as

    $J(\alpha) = -\frac{1}{2} \Sigma\ \alpha_i\, \alpha_j\, y_i\, y_j\, K(x_i, x_j) + \Sigma\ F(\alpha_i)$

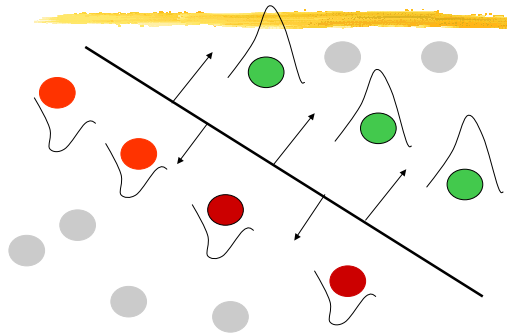
Leave One Out



SVMs

- $y_t = \text{sign}\,(\Sigma\ \alpha_i\, y_i\, K(x_t, x_i))$
- $(y_i\ x_i)$ training data, α_i nonnegative, and kernel K positive definite
- α_i is obtained by maximizing
  - $J(\alpha) = -\frac{1}{2} \Sigma\ \alpha_i\, \alpha_j\, y_i\, y_j\, K(x_i, x_j) + \Sigma\ F(\alpha_i)$
  - $F(\alpha_i) = \alpha_i$
  - $\alpha_i \geqslant 0,\ \Sigma y_i \alpha_i = 0$

SVMs



Important Insight

- $K(x_i, x_j) = \text{Cov}\,(z_i, z_j)$
- To design of a kernel is to design a similarity function that produces a positive definite covariance matrix on the training instances

Basis Function Selection

- Three General Approaches
  - Restriction methods
    - Limit the class of functions
  - Selection methods
    - Scan the dictionary adaptively (Boosting)
  - Regularization methods
    - Use the entire dictionary but restrict coefficients (Ridge Regression)

Overfitting?

- Probably Not
- Because
  - N free parameters (not D)
  - Maximizing margin

Geometrical View

- $S = w X + b$
- $|w| = 1, b = 0$
- $V = \{w \mid S_i f(x_i) > 0; i = 1..n, |w| = 1\}$
- SVM is the center of the largest sphere contained in V

SVMs

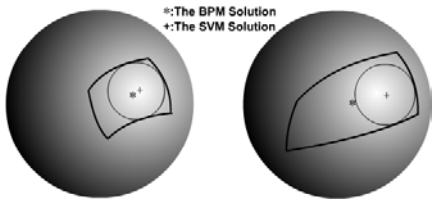
BPMs

- Bayes Objective Function
  - $\hat{S}_t$ = Bayes $_Z$ (X_t)
    = argmin $_{S_i \text{ in } S}$ E $_{H|Z = x}$ [l(H(x), Si)]
- BPMs [Herbrich, etc. 2001]
  - A_{bp}= argmin $_{h \text{ in } H}$ E_x[E $_{H|Z = x}$ [l(H(x), h(x))]]

BPMs

- Linear Classifier
- Input X Posses Spherical Gaussian Density
- BP is the Center of Mass of the Version Space

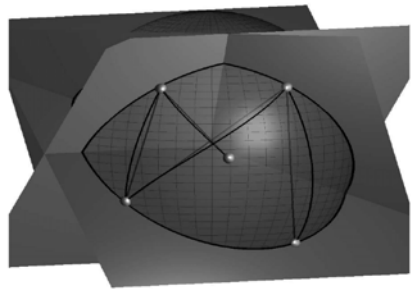
BPMs vs. SVMs



BPMs

- Use SVMs to find a good h in H
- Find the BP
  - Billiard Algorithm [Herbrich, etc. 2001]
  - Perceptron Algorithm [Herbrich, etc. 2001]

Billiard Ball Algorithm (R. Herbrich)



Outline

- Statistical Learning
- Emerging Applications Data Characteristics
- Classical Models (Classification)
- Kernel Methods
- Dimension Reduction Methods

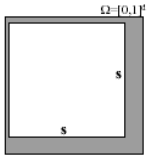
Dimensionality Curse

- D: Data Dimension
- When D increases
  - Nearest neighbors are not local
  - All points are equally distanced

Sparse High-D Space

[C. Aggarwal, etc. ICDT 2001]

- Hyper-cube Range Queries



$$P^d[s] = s^d$$

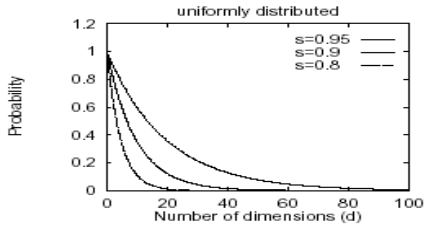
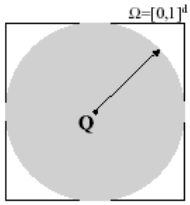


Figure 2: The probability function $P^d[s]$.

Sparse High-D Space

- Spherical Range Queries

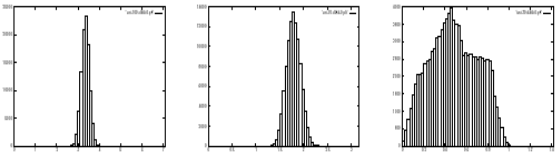


$$P[R \in sp^d(Q,0.5)] = \frac{\sqrt{\pi^d} \bullet (0.5)^d}{\Gamma(\frac{d}{2}+1)}$$

d	$P[R \in sp^d(Q,0.5)]$	$N(d)$
2	0.785	1.273
4	0.308	3.242
10	0.002	401.5
20	$2.461 \cdot 10^{-8}$	40'631'627
40	$3.278 \cdot 10^{-21}$	$3.050 \cdot 10^{20}$
100	$1.868 \cdot 10^{-70}$	$5.353 \cdot 10^{69}$

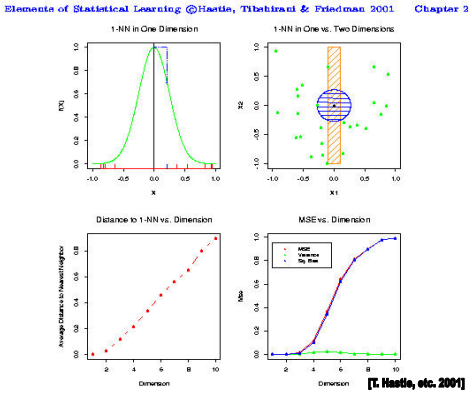
Table 2: Probability that a point lies within the largest range query inside Ω, and the expected database size.

Dimensionality Curse



(a) 50 Dimensions (b) 10 Dimensions (c) 2 Dimensions

Figure 3: Distance Distribution of Uniform Data



[T. Hastie, etc. 2001]

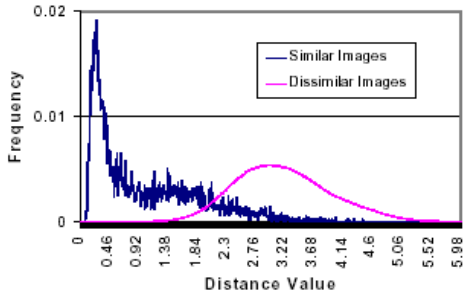
So?

- Is nearest neighbor *estimate* cursed in high-D spaces?
  - Yes!
  - When D is large and N is relatively small, the estimate is off!!

Are We Doomed?

- How does the curse affect classification?
- Similar objects tend to cluster together
- Classification makes binary prediction

Distribution of Distances



(a) $m = 144$

Some Solutions to High-D

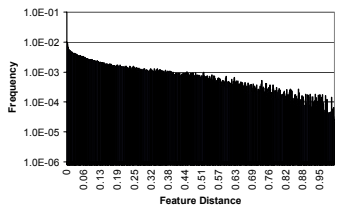
- Restricted Estimators
  - Specifying the nature of local neighborhood
- Adaptive Feature Reduction
  - PCA, LDA
- Dynamic Partial Function

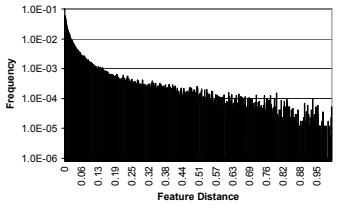
Three Major Paradigms

- Preserve data description in a lower dimensional space
  - PCA
- Maximize discriminability in a lower dimensional space
  - LDA
- Activate only similar channels
  - DPF

Minkowski Distance

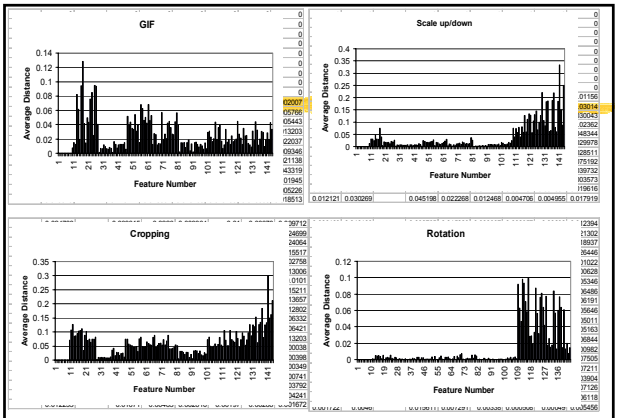
- Objects P and Q
- $D = (\Sigma_M (p_i - q_i)^n)^{1/n}$
- Similar images are similar in all M features





Weighted Minkowski Distance

- $D = (\Sigma_M w_i(p_i - q_i)^n)^{1/n}$
- Similar images are similar in the same subset of the M features



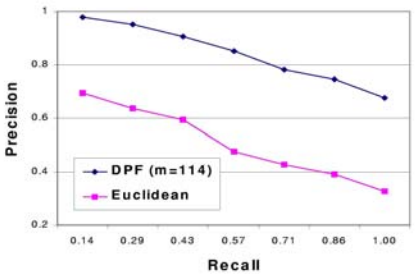
Similarity Theories

- Objects are similar in all respects (Richardson 1928)
- Objects are similar in some respects (Tversky 1977)
- Similarity is a process of determining respects, rather than using predefined respects (Goldstone 94)

DPF

- Which Place is Similar to Kyoto?
- Partial
- Dynamic
- Dynamic Partial Function

Precision/Recall



Summary

- Statistical Learning
- Emerging Applications Data Characteristics
- Classical Models (Classification)
- Kernel Methods
  - Linear Model View
  - Nearest Neighbor View
  - Geometric View
- Dimension Reduction Methods

Emerging DB Applications

- N < D
- $N^+ << N^-$
- Examples
  - Information Retrieval with relevance feedback
  - Gene Profiling
  - Bioinformatics

Useful Links

- Related Publications
  - http://www-db.stanford.edu/~echang/
- Software Free Trial
  - http://www.imagebeagle.com
  - Locate objectionable images on your hard drives
  - Before your boss finds it!!!

References

1. The Elements of Statistical Learning, T. Hastie, R. Tibshirani, and J. Friedman, Springer, N.Y., 2001
2. Machine Learning, T. Mitchell, 1997
3. High-dimensional Data Analysis, D. Donoho, American Math. Society Lecture, 2000
4. Support Vector Machine Active Learning for Image Retrieval, S. Tong and E. Chang, ACM MM, 2001
5. Dynamic Partial Function, B. Li and E. Chang, ACM Journal, 2003
6. Pattern Discovery in Sequences under a Markov Assumption, D. Chudova and P. Smyth, ACM KDD 2002
7. Bayes Point Machines, R. Herbrich, T. Graepel and C. Campbell, Journal of Machine Learning Research, 2001
8. The Nature of Statistical Learning Theory, V. Vapnik, Springer, N.Y., 1995
9. Probabilistic Kernel Regression Models, T. Jaakkola and D. Haussler, Conference of AI and Statistics, 1999
10. Support Vector Machines, Lecture Notes, A. Moore, CMU
11. On the Surprising Behavior of Distance Metrics in High-dimensional Space, C. Aggarwal, A. Hinneburg, and D. Keim, ICDT 2001