WALD LECTURE 1

MACHINE LEARNING

Leo Breiman
UCB Statistics
leo@stat.berkeley.edu

A ROAD MAP

First--a brief overview of what we call machine learning but consists of many diverse interests (not including data mining). How I became a token statistician in this community.

Second--an exploration of ensemble predictors.

Beginning with bagging.

Then onto boosting

ROOTS

Neural Nets--invented circa 1985

Brought together two groups:

A: Brain researchers applying neural nets to model some functions of the brain.

B. Computer scientists working on:

speech recognition
written character recognition
other hard prediction problems.

JOINED BY

C. CS groups with research interests in training robots.

Supervised training

Stimulus was CART- circa 1985

(Machine Learning)

Self-learning robots

(Reinforcement Learning)

D. Other assorted groups

Artificial Intelligence

PAC Learning

etc. etc.

MY INTRODUCTION

1991

Invited talk on CART at a Machine Learning Conference.

Careful and methodical exposition assuming they had never heard of CART.

(as was true in Statistics)

How embarrassing:

> After talk found that they knew all about CART and were busy using its lookalike C4.5

`

NIPS

(neural information processing systems)

Next year I went to NIPS--1992-3 and have gone every year since except for one.

In 1992 NIPS was hotbed of use of neural nets for a variety of purposes.

prediction
control
brain models

A REVELATION!

Neural Nets actually work in prediction:

despite a multitude of
local minima

despite the dangers
of overfitting

Skilled practitioners tailored large architectures of hidden units to accomplish special purpose results in specific problems

i.e. partial rotation and translation invariance character recognition.

NIPS GROWTH

NIPS grew to include many diverse groups:

signal processing
computer vision

etc.

One reason for growth--skiing.
Vancouver --Dec. 9-12th, Whistler 12-15th

In 2001 about 600 attendees
Many foreigners--especially Europeans

Mainly computer scientists, some engineers, physicists, mathematical physiologists, etc.

Average age--30--Energy level--out-of-sight
Approach is strictly algorithmic.

For me, algorithmic oriented, who felt like a voice in the wilderness in statistics, this community was like home. My research was energized.

The papers presented at the NIPS 2000 conference are listed in the following to give a sense of the wide diversity of research interests.

- What Can a Single Neuron Compute?
- Who Does What? A Novel Algorithm to Determine Function Localization
- Programmable Reinforcement Learning Agents
- From Mixtures of Mixtures to Adaptive Transform Coding
- Dendritic Compartmentalization Could Underlie Competition and Attentional Biasing of Simultaneous Visual Stimuli
- Place Cells and Spatial Navigation Based on 2D Visual Feature Extraction, Path Integration, and Reinforcement Learning
- Speech Denoising and Dereverberation Using Probabilistic Models
- Combining ICA and Top-Down Attention for Robust Speech Recognition
- Modelling Spatial Recall, Mental Imagery and Neglect
- Shape Context: A New Descriptor for Shape Matching and Object Recognition
- Efficient Learning of Linear Perceptrons
- A Support Vector Method for Clustering
- A Neural Probabilistic Language Model
- A Variational Mean-Field Theory for Sigmoidal Belief Networks
- Stability and Noise in Biochemical Switches
- Emergence of Movement Sensitive Neurons' Properties by Learning a Sparse Code for Natural Moving Images
- New Approaches Towards Robust and Adaptive Speech Recognition
- Algorithmic Stability and Generalization Performance
- Exact Solutions to Time-Dependent MDPs
- Direct Classification with Indirect Data
- Model Complexity, Goodness of Fit and Diminishing Returns
- A Linear Programming Approach to Novelty Detection
- Decomposition of Reinforcement Learning for Admission Control of Self-Similar Call Arrival Processes
- Overfitting in Neural Nets: Backpropagation, Conjugate Gradient, and Early Stopping
- Incremental and Decremental Support Vector Machine Learning
- Vicinal Risk Minimization

- Temporally Dependent Plasticity: An Information Theoretic Account
- Gaussianization
- The Missing Link - A Probabilistic Model of Document and Hypertext Connectivity
- The Manhattan World Assumption: Regularities in Scene Statistics which Enable Bayesian Inference
- Improved Output Coding for Classification Using Continuous Relaxation
- Koby Crammer, Yoram Singer
- Sparse Representation for Gaussian Process Models
- Competition and Arbors in Ocular Dominance
- Explaining Away in Weight Space
- Feature Correspondence: A Markov Chain Monte Carlo Approach
- A New Model of Spatial Representation in Multimodal Brain Areas
- An Adaptive Metric Machine for Pattern Classification
- High-temperature Expansions for Learning Models of Nonnegative Data
- Incorporating Second-Order Functional Knowledge for Better Option Pricing
- A Productive, Systematic Framework for the Representation of Visual Structure
- Discovering Hidden Variables: A Structure-Based Approach
- Multiple Timescales of Adaptation in a Neural Code
- Learning Joint Statistical Models for Audio-Visual Fusion and Segregation
- Accumulator Networks: Suitors of Local Probability Propagation
- Sequentially Fitting ``Inclusive'' Trees for Inference in Noisy-OR Networks
- Factored Semi-Tied Covariance Matrices
- A New Approximate Maximal Margin Classification Algorithm
- Propagation Algorithms for Variational Bayesian Learning
- Reinforcement Learning with Function Approximation Converges to a Region
- The Kernel Gibbs Sampler

- From Margin to Sparsity
- `N-Body' Problems in Statistical Learning
- A Comparison of Image Processing Techniques for Visual Speech Recognition Applications
- The Interplay of Symbolic and Subsymbolic Processes in Anagram Problem Solving
- Permitted and Forbidden Sets in Symmetric Threshold-Linear Networks
- Support Vector Novelty Detection Applied to Jet Engine Vibration Spectra
- Large Scale Bayes Point Machines
- A PAC-Bayesian Margin Bound for Linear Classifiers: Why SVMs work
- Hierarchical Memory-Based Reinforcement Learning
- Beyond Maximum Likelihood and Density Estimation: A Sample-Based Criterion for Unsupervised Learning of Complex Models
- Ensemble Learning and Linear Response Theory for ICA
- A Silicon Primitive for Competitive Learning
- On Reversing Jensen's Inequality
- Automated State Abstraction for Options using the U-Tree Algorithm
- Dopamine Bonuses
- Hippocampally-Dependent Consolidation in a Hierarchical Model of Neocortex
- Second Order Approximations for Probability Models
- Generalizable Singular Value Decomposition for Ill-posed Datasets
- Some New Bounds on the Generalization Error of Combined Classifiers
- Sparsity of Data Representation of Optimal Kernel Machine and Leave-one-out Estimator
- Keeping Flexible Active Contours on Track using Metropolis Updates
- Smart Vision Chip Fabricated Using Three Dimensional Integration Technology
- Algorithms for Non-negative Matrix Factorization

- Color Opponency Constitutes a Sparse Representation for the Chromatic Structure of Natural Scenes
- Foundations for a Circuit Complexity Theory of Sensory Processing
- A Tighter Bound for Graphical Models
- Position Variance, Recurrence and Perceptual Learning
- Homeostasis in a Silicon Integrate and Fire Neuron
- Text Classification using String Kernels
- Constrained Independent Component Analysis
- Learning Curves for Gaussian Processes Regression: A Framework for Good Approximations
- Active Support Vector Machine Classification
- Weak Learners and Improved Rates of Convergence in Boosting
- Recognizing Hand-written Digits Using Hierarchical Products of Experts
- Learning Segmentation by Random Walks
- The Unscented Particle Filter
- A Mathematical Programming Approach to the Kernel Fisher Algorithm
- Automatic Choice of Dimensionality for PCA
- On Iterative Krylov-Dogleg Trust-Region Steps for Solving Neural Networks Nonlinear Least Squares Problems
- Eiji Mizutani, James W. Demmel
- Sex with Support Vector Machines
- Baback Moghaddam, Ming-Hsuan Yang
- Robust Reinforcement Learning
- Partially Observable SDE Models for Image Sequence Recognition Tasks
- The Use of MDL to Select among Computational Models of Cognition
- Probabilistic Semantic Video Indexing
- Finding the Key to a Synapse
- Processing of Time Series by Neural Circuits with Biologically Realistic Synaptic Dynamics
- Active Inference in Concept Learning

- Learning Continuous Distributions: Simulations With Field Theoretic Priors
- Interactive Parts Model: An Application to Recognition of On-line Cursive Script
- Learning Sparse Image Codes using a Wavelet Pyramid Architecture
- Kernel-Based Reinforcement Learning in Average-Cost Problems: An Application to Optimal Portfolio Choice
- Learning and Tracking Cyclic Human Motion
- Higher-Order Statistical Properties Arising from the Non-Stationarity of Natural Signals
- Learning Switching Linear Models of Human Motion
- Bayes Networks on Ice: Robotic Search for Antarctic Meteorites
- Redundancy and Dimensionality Reduction in Sparse-Distributed Representations of Natural Objects in Terms of Their Local Features
- Fast Training of Support Vector Classifiers
- The Use of Classifiers in Sequential Inference
- Occam's Razor
- One Microphone Source Separation
- Using Free Energies to Represent Q-values in a Multiagent Reinforcement Learning Task
- Minimum Bayes Error Feature Selection for Continuous Speech Recognition
- Periodic Component Analysis: An Eigenvalue Method for Representing Periodic Structure in Speech
- Spike-Timing-Dependent Learning for Oscillatory Networks
- Universality and Individuality in a Neural Code
- Machine Learning for Video-Based Rendering
- The Kernel Trick for Distances
- Natural Sound Statistics and Divisive Normalization in the Auditory System
- Balancing Multiple Sources of Reward in Reinforcement Learning
- An Information Maximization Approach to Overcomplete and Recurrent Representations

- Development of Hybrid Systems: Interfacing a Silicon Neuron to a Leech Heart Interneuron
- FaceSync: A Linear Operator for Measuring Synchronization of Video Facial Images and Audio Tracks
- The Early Word Catches the Weights
- Sparse Greedy Gaussian Process Regression
- Regularization with Dot-Product Kernels
- APRICODD: Approximate Policy Construction Using Decision Diagrams
- Four-legged Walking Gait Control Using a Neuromorphic Chip Interfaced to a Support Vector Learning Algorithm
- Kernel Expansions with Unlabeled Examples
- Analysis of Bit Error Probability of Direct-Sequence CDMA Multiuser Demodulators
- Noise Suppression Based on Neurophysiologically-motivated SNR Estimation for Robust Speech Recognition
- Rate-coded Restricted Boltzmann Machines for Face Recognition
- Structure Learning in Human Causal Induction
- Sparse Kernel Principal Component Analysis
- Data Clustering by Markovian Relaxation and the Information Bottleneck Method
- Adaptive Object Representation with Hierarchically-Distributed Memory Sites
- Active Learning for Parameter Estimation in Bayesian Networks
- Mixtures of Gaussian Processes
- Bayesian Video Shot Segmentation
- Error-correcting Codes on a Bethe-like Lattice
- Whence Sparseness?
- Tree-Based Modeling and Estimation of Gaussian Processes on Graphs with Cycles
- Algebraic Information Geometry for Learning Machines with Singularities
- Feature Selection for SVMs?
- On a Connection between Kernel PCA and Metric Multidimensional Scaling
- Using the Nystr{\"o}m Method to Speed Up Kernel Machines

- Computing with Finite and Infinite Networks
- Stagewise Processing in Error-correcting Codes and Image Restoration
- Learning Winner-take-all Competition Between Groups of Neurons in Lateral Inhibitory Networks
- Generalized Belief Propagation
- A Gradient-Based Boosting Algorithm for Regression Problems
- Divisive and Subtractive Mask Effects: Linking Psychophysics and Biophysics
- Regularized Winnow Methods
- Convergence of Large Margin Separable Linear Classification

PREDICTION REMAINS A MAIN THREAD

Given a training set of data

$$\mathbf{T}= (\mathbf{y}_n, \mathbf{x}_n)\, n = 1, \ldots, N\}$$

where the $\mathbf{y}_n$ are the response vectors and the $\mathbf{x}_n$ are vectors of predictor variables:

Find a function f operating on the space of prediction vectors with values in the response vector space such that:

If the $(\mathbf{y}_n, \mathbf{x}_n)$ are i.i.d from the distribution **(Y,X)** and given a function L(**y,y'**) that measures the loss between **y a**nd the prediction **y**': the prediction error

$$PE(f, \mathbf{T}) = E_{\mathbf{Y},\mathbf{X}} L(\mathbf{Y}, f(\mathbf{X}, \mathbf{T}))$$

is small.

Usually **y** is one dimensional. If numerical, the problem is regression. If unordered labels, it is classification. In regression, the loss is squared error. In classification, if the predicted label does not equal the true label the loss is one, zero other wise

RECENT BREAKTHROUGHS

Two types of classification algorithms originated in 1996 that gave improved accuracy.

A. Support vector Machines (Vapnik)

B. Combining Predictors:

Bagging (Breiman 1996)
Boosting (Freund and Schapire 1996)

Both bagging and boosting use ensembles of predictors defined on the prediction variables in the training set.

Let $\{f_1(\mathbf{x},\mathbf{T}), f_2(\mathbf{x},\mathbf{T}),..., f_K(\mathbf{x},\mathbf{T})\}$ be predictors constructed using the training set **T** such that for every value of **x** in the predictor space they output a value of y in the response space.

In regression, the predicted value of y corresponding to an input **x** is $av_k f_k(\mathbf{x},\mathbf{T})$

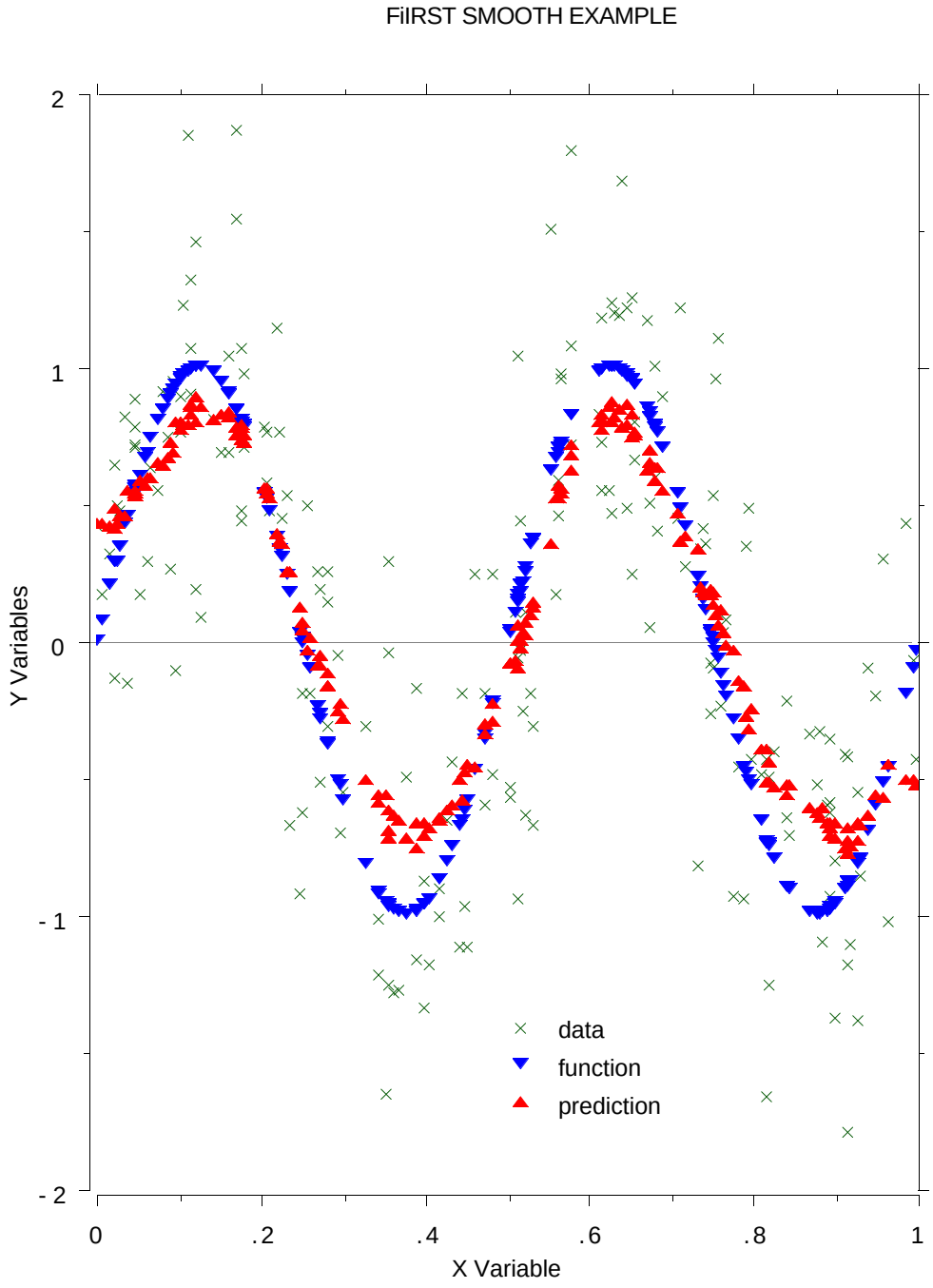
In classification the output takes values in the class labels {1,2,...,J}. The predicted value of y is $plur_k f_k(\mathbf{x},\mathbf{T})$

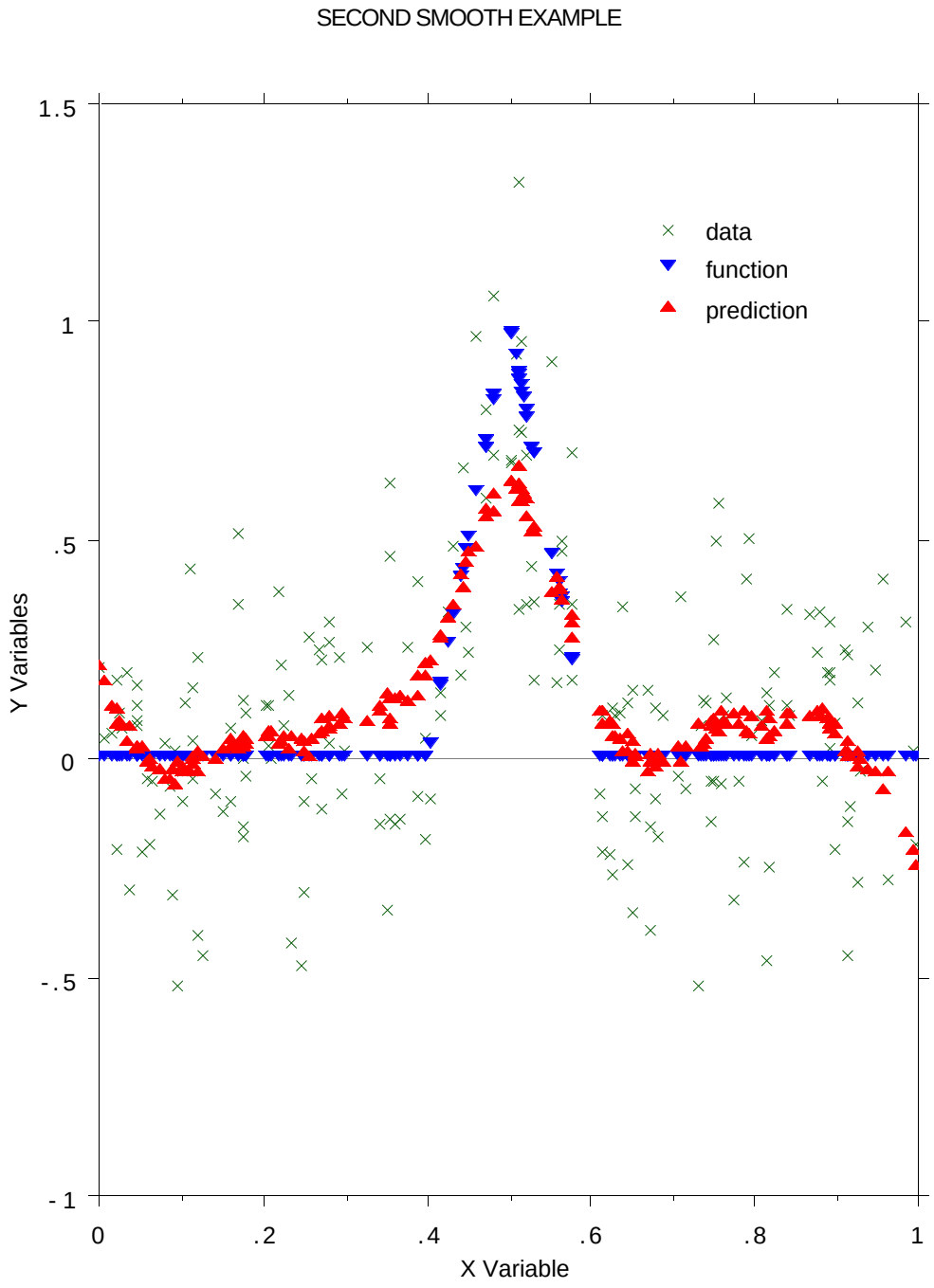
The averaging and voting can be weighted,

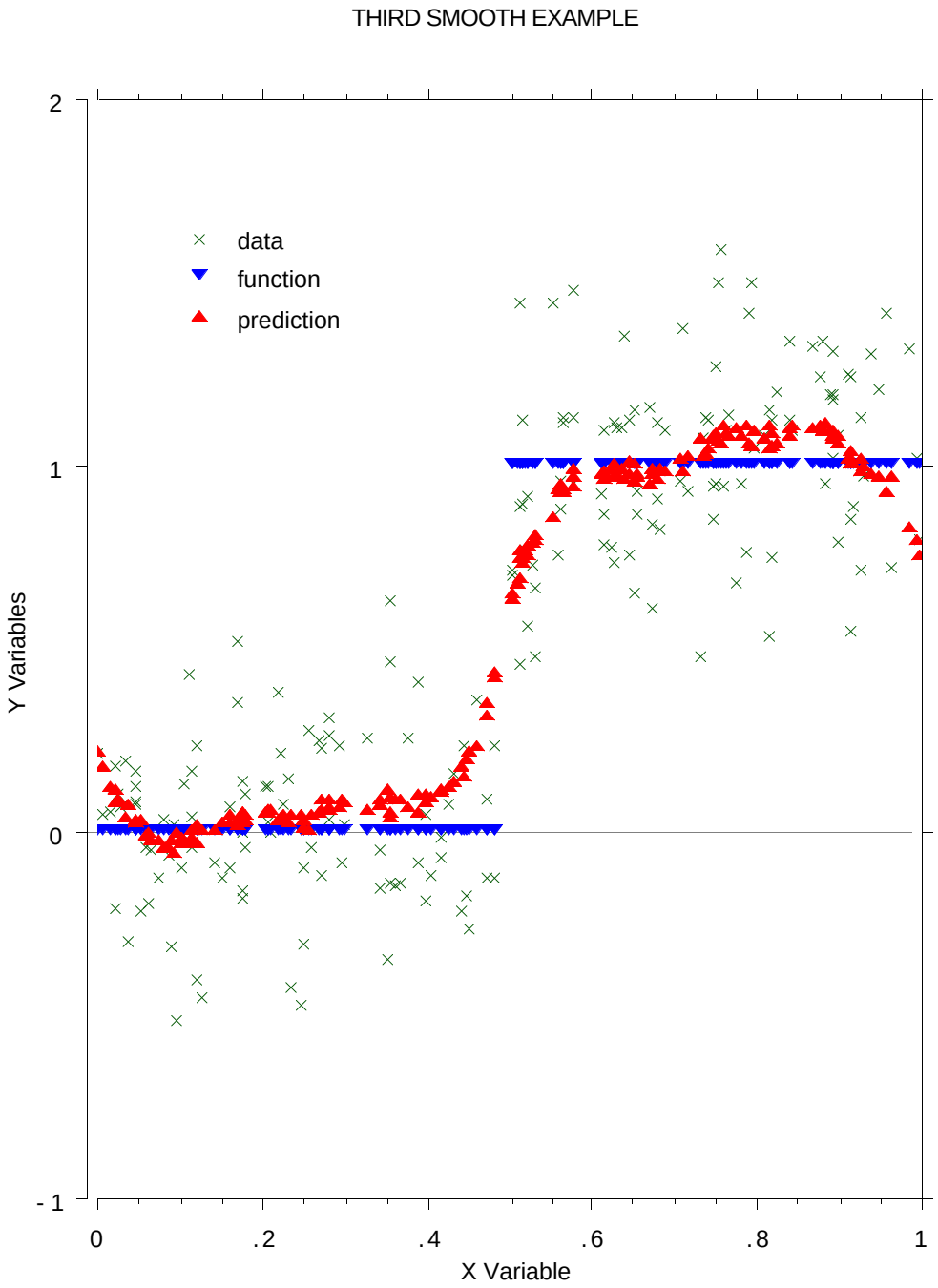
THE STORY OF BAGGING

as illustrated to begin with by pictures of three one dimensional smoothing examples using the same smoother.

They are not really smoothers-but predictors of the underlying function

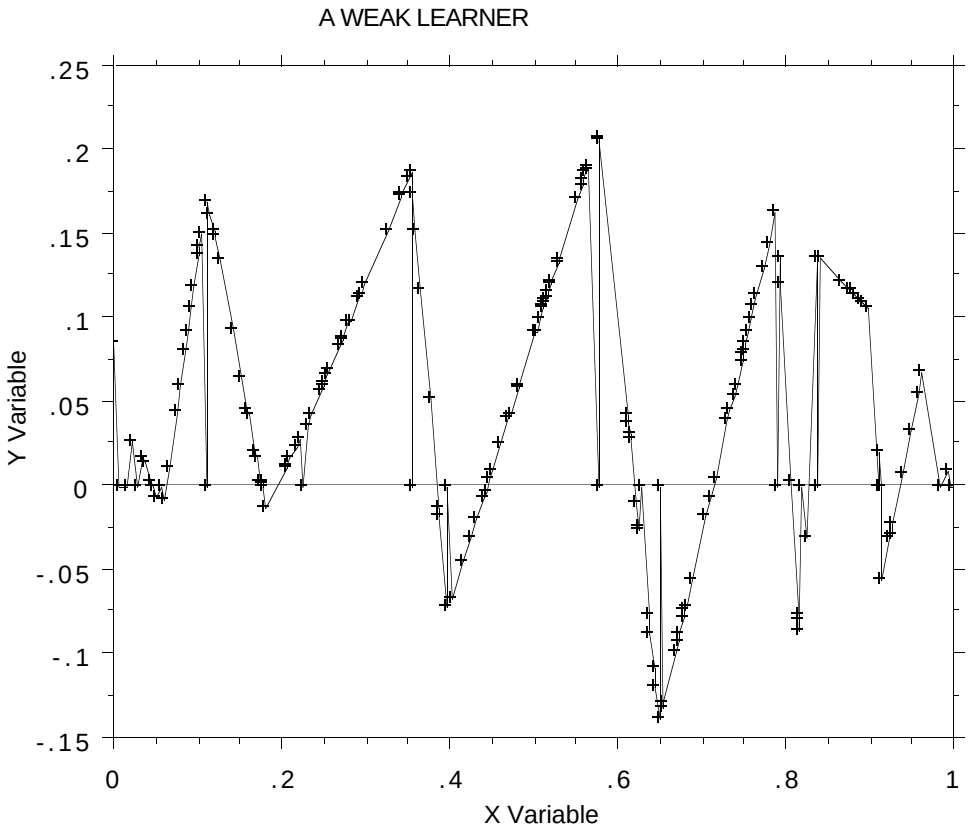

FiIRST SMOOTH EXAMPLE
2
1
0
- 1
- 2
Y Variables
0
.2
.4
.6
.8
1
X Variable
data
function
prediction


SECOND SMOOTH EXAMPLE
1.5
1
.5
0
-.5
-1
Y Variables
0
.2
.4
.6
.8
1
X Variable
data
function
prediction

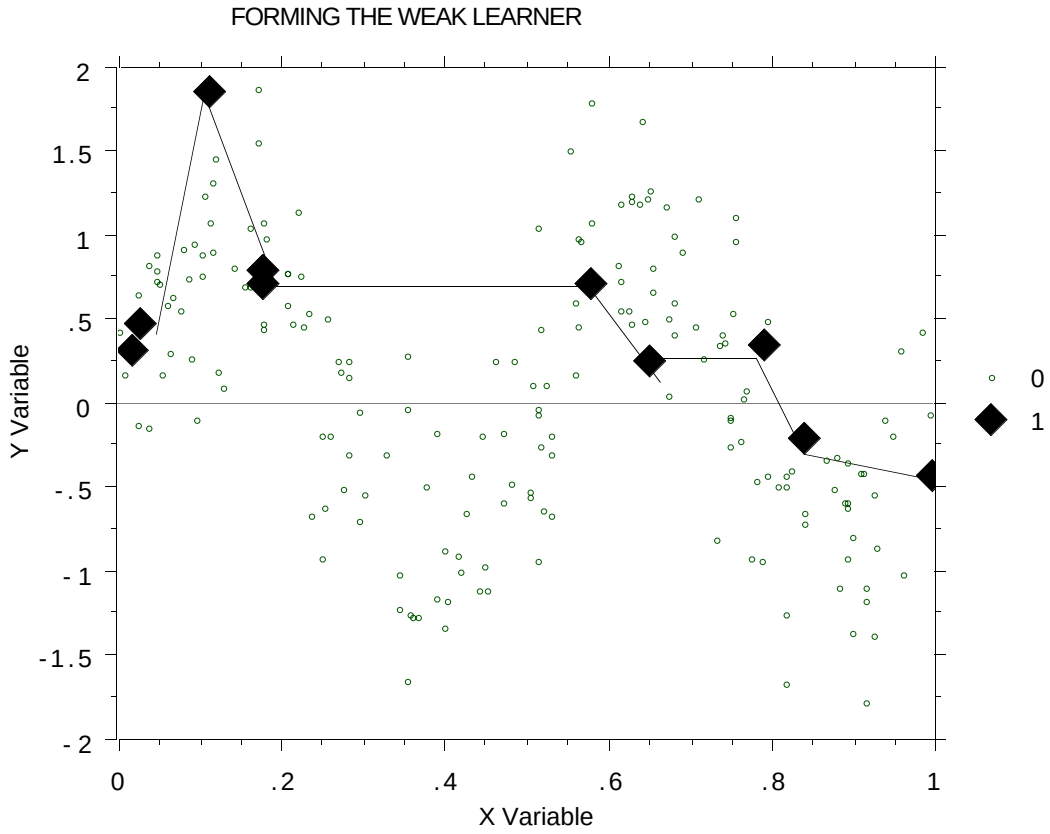

THIRD SMOOTH EXAMPLE
data
function
prediction
Y Variables
X Variable
2
1
0
- 1
0
.2
.4
.6
.8
1

WHAT SMOOTH?

Here is a weak learner--



The smooth is an average of 1000 weak learners. Here is how the weak learners are formed:



Subset of fixed size is selected at random. Then all the (y,x) points in the subset are connected by lines.

Repeated 1000 times and the 1000 weak learners averaged.

THE PRINCIPLE

Its easier to see what is going on in regression:

$$PE(f,\mathbf{T}) = E_{Y,\mathbf{X}}(Y - f(\mathbf{X},\mathbf{T}))^2$$

Want to average over all training sets of same size drawn from the same distribution:

$$PE(f) = E_{Y,\mathbf{X},\mathbf{T}}(Y - f(\mathbf{X},\mathbf{T}))^2$$

This is decomposable into:

$$PE(f) = E_{Y,\mathbf{X}}(Y - E_{\mathbf{T}} f(\mathbf{X},\mathbf{T}))^2 +$$

$$E_{\mathbf{X},\mathbf{T}}(f(\mathbf{X},\mathbf{T}) - E_{\mathbf{T}} f(\mathbf{X},\mathbf{T}))^2$$

Or

$$PE(f) = (bias)^2 + \mathrm{var}\,iance$$

(Pretty Familiar!)

BACK TO EXAMPLE

The kth weak learner is of the form:

$$f_k(\mathbf{x},\mathbf{T}) = f(\mathbf{x},\mathbf{T},\Theta_k)$$

where Θ_k is the random vector that selects the points to be in the weak learner. The Θ_k are i.i.d.

The ensemble predictor is:

$$F(\mathbf{x},\mathbf{T}) = \frac{1}{K}\sum_k f(\mathbf{x},\mathbf{T},\Theta_k)$$

Algebra and the LLN leads to:

$$Var(F) =$$

$$E_{\mathbf{X},\Theta,\Theta'}[\rho_{\mathbf{T}}(f(\mathbf{x},\mathbf{T},\Theta)f(\mathbf{x},\mathbf{T},\Theta'))Var_{\mathbf{T}}(f(\mathbf{x},\mathbf{T},\Theta)$$

where Θ,Θ' are independent. Applying the mean value theorem--

$$Var(F) = \bar{\rho}\,Var(f)$$

and

$$Bias^2(F) = E_{Y,\mathbf{X}}(Y - E_{\mathbf{T},\Theta}f(\mathbf{x},\mathbf{T},\Theta))^2$$

THE MESSAGE

A big win is possible with weak learners as long as their correlation and bias are low.

In sin curve example, base predictor is connect all points in order of x(n).

$bias^2$=.000
variance=.166

For the ensemble

$bias^2$ = .042
variance =.0001

Bagging is of this type--each predictor is grown on a bootstrap sample, requiring a random vector Θ that puts weights 0,1,2,3, on the cases in the training set.

But bagging does not produce as low as possible correlation between the predictors. There are variants that produce lower correlation and better accuracy

This Point Will Turn Up Again Later.

BOOSTING--A STRANGE ALGORITHM

I discuss boosting, in part, to illustrate the difference between the machine learning community and statistics in terms of theory.

But mainly because the story of boosting is fascinating and multifaceted.

Boosting is a classification algorithm that gives consistently lower error rates than bagging.

Bagging works by taking a bootstrap sample from the training set.

Boosting works by changing the weights on the training set.

It assumes that the predictor construction can incorporate weights on the cases.

The procedure for growing the ensemble is--
Use the current weights to grow a predictor.

Depending on the training set errors of this predictor, change the weights and grow the next predictor.

THE ADABOOST ALGORITHM

The weights $w(n)$ on the nth case in the training set are non-negative and sum to one. Originally they are set equal. The process goes like so:

i) let $w^{(k)}(n)$ be the weights for the kth step. f_k the classifier constructed using these weights.

ii) Run the training set down f_k and let $d(n)$=1 if the nth case is classified in error, otherwise zero.

iii) The weighted error is $\varepsilon_k = \sum_n w^{(k)}(n)d(n)$

set $\beta_k = (1-\varepsilon_k)/\varepsilon_k$

iv) The new weights are

$$w^{(k+1)}(n) = w^{(k)}(n)\beta_k^{d(n)} / \sum_n w^{(k)}(n)\beta_k^{d(n)}$$

v) Voting for class is weighted with kth classifier having vote weight β_k

THE MYSTERY THICKENS

Adaboost created a big splash in machine learning and led to hundreds, perhaps thousands of papers. It was the most accurate classification algorithm available at that time.

It differs significantly from bagging. Bagging uses the biggest trees possible as the weak learners to reduce bias.

Adaboost uses small trees as the weak learners, often being effective using trees formed by a single split (stumps).

There is empirical evidence that it reduces bias as well as variance.

It seemed to converges with the test set error gradually decreasing as hundreds or thousands of trees were added.

On simulated data its error rate is close to the Bayes rate.

But why it worked so well was a mystery that bothered me. For the last five years I have characteriized the understanding of Adaboost as the most important open problem in machine learning.

IDEAS ABOUT WHY IT WORKS

A. Adaboost raises the weights on cases previously misclassified, so it focusses on the hard cases ith the easy cases just carried along.

wrong: empirical results showed that Adaboost tried to equalize the misclassification rate over all cases.

B. The margin explanation: An ingenious work by Shapire, et.al. derived an upper bound on the error rate of a convex combination of predictors in terms of the VC dimension of each predictor in the ensemble and the margin distribution.

The margin for the nth case is the vote in the ensemble for the correct class minus the largest vote for any of the other classes.

The authors conjectured that Adaboost was so poweful because it produced high margin distributions.

I devised and published an algorithm that produced uniformly higher margin disrbutions than Adaboost, and yet was less accurate.

So much for margins.

THE DETECTIVE STORY CONTINUES

There was little continuing interest in machine learning about how and why Adaboost worked.

Often the two communities, statistics and machine learning, ask different questions

Machine Learning: Does it work?
Statisticians: Why does it work?

One breakthrough occurred in my work in 1997.

> In the two-class problem, label the classes as -1,+1. Then all the classifiers in the ensemble also take the values -1,+1.
>
> Denote by $F(\mathbf{x}_n)$ any ensemble evaluated at $\mathbf{x}_n$ If $F(\mathbf{x}_n)>0$ the prediction is class 1, else class -1.
>
> On average, want $y_n F(\mathbf{x}_n)$ to be as large as possible. Consider trying to minimize
>
> $$\sum_n \phi(y_n F_m(\mathbf{x}_n))$$
>
> where $\phi(x)$ is decreasing.

GAUSS-SOUTHWELL

The Gauss-Southwell method for minimizing a differentiable function $g(x_1,\ldots,x_m)$ of m real variables goes this way:

i) At a point **x** compute all the partial derivatives $\partial f(x_1,\ldots,x_m)/\partial x_k$.

ii) Let the minimum of these be at x_j. Find step of size α that minimizes $g(x_1,\ldots,x_j+\alpha,\ldots,x_m)$

iii) Let the new **x** be $x_1,\ldots,x_j+\alpha,\ldots,x_m$ for the minimizing α value.

GAUSS SOUTHWELL AND ADABOOST

To minimize $\sum_n \exp(-y_n F(\mathbf{x}_n))$ using Gauss-Southwell. Denote the current ensemble as

$$F_m(\mathbf{x}_n) = \sum_1^m a_k f_k(\mathbf{x}_n)$$

i) Find the k=k* that minimizes

$$\frac{\partial}{\partial f_k} \sum_n \exp(-y_n F_m(\mathbf{x}_n))$$

ii) Find that a=a* that minimizes

$$\sum_n \exp(-y_n [F_m(\mathbf{x}_n) + a^* f_{k*}(\mathbf{x}_n)])$$

iii) Then $F_{m+1}(\mathbf{x}_n) = F_m(\mathbf{x}_n) + a^* f_{k^*}(\mathbf{x}_n)$

The Asdaboost algorithm is identical to Gauss-Southwell as applied above.

This gave a rational basis for the odd form of Adaboost. Following was a plethora of papers in machine learning proposing other functions of $yF(\mathbf{x})$ to minimize using Gauss-Southwell

NAGGING QUESTIONS

The classification by the mth ensemble is defined by

$$sign(F_m(\mathbf{x}))$$

The most important question I chewed on

Is Adaboost consistent?

Does $P(Y \neq sign(F_m(\mathbf{X},\mathbf{T})))$ converge to the Bayes risk as $m \to \infty$ and then $|\mathbf{T}| \to \infty$?

I am not a fan of endless asymptotics, but I believe that we need to know whether predictors are consistent or inconsistent

For five years I have been bugging my theoretical colleagues with these questions.

For a long time I thought the answer was yes.

There was a paper 3 years ago which claimed that Adaboost overfit after 100,000 iterations, but I ascribed that to numerical roundoff error

THE BEGINNING OF THE END

In 2000, I looked at the analog of Adaboost in population space, i.e. using the Gauss-Southwell approach, minimize

$$E_{Y,\mathbf{X}} \exp(-YF(\mathbf{X}))$$

The weak classifiers were the set of all trees with a fixed number (large enough) of terminal nodes.

Under some compactness and continuity conditions I proved that:

$$F_m \to F \quad in \quad L_2(P)$$

$$P(Y \neq sign(F(\mathbf{X})) = Bayes\ Risk$$

But there was a fly in the ointment

THE FLY

Recall the notation

$$F_m(\mathbf{x}) = \sum_{1}^{m} a_k f_k(\mathbf{x})$$

An essential part of the proof in the population case was showing that:

$$\Sigma a_k^2 < \infty$$

But in the N-sample case, one can show that

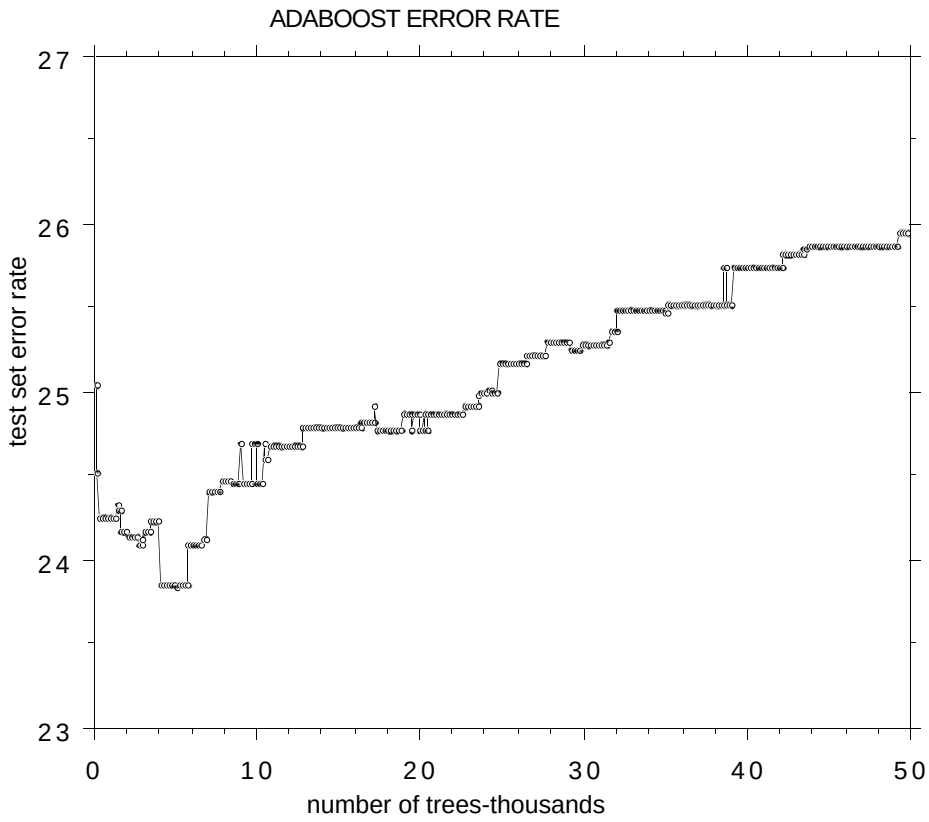
$$a_k \geq 2 / N$$

So there was an essential difference between the population case and the finite sample case no matter how large N

ADABOOST IS INCONSISTENT

Recent work by Jiang, Lugosi, and Bickel-Ritov have clarified the situation.

The graph below illustrates. The2-dimensional data consists of two circular Gaussians with about 150 cases in each with some overlap. Error is estimated using a 5000 case test set. Stumps (one split trees) were used.



The minimum occurs at about 5000 trees. Then the error rate begins climbing.

WHAT ADABOOST DOES

In its first stage, Adaboos tries to emulate the population version. This continues for thousands of trees. Then it gives up and moves into a second phase of increasing error .

Both Jiang and Bickel-Ritov have proofs that for each sample size N, there is a stopping time h(N) such that if Adaboost is stopped at h(N), the resulting sequence of ensembles is consistent.

There are still questions--what is happening in the second phase? But this will come in the future.

For years I have been telling everyone in earshot that the behavior of Adaboost, particularly consistency, is a problem that plagues Machine learning.

Its solution is at the fascinating interface between algorithmic behavior and statistical theory.

THANK YOU