Data Analysis, Statistics, Machine Learning

Leland Wilkinson

Adjunct Professor
UIC Computer Science
Chief Scientist
H2O.ai

leland.wilkinson@gmail.com

Learning

- Machine Learning (ML) methods look for patterns that persist across a large collection of data objects
- ML learns from new data
- Key concepts
  - Curse of dimensionality
  - Random projections
  - Regularization
  - Kernels
  - Bootstrap aggregation
  - Boosting
  - Ensembles
  - Validation
  - No Free Lunch Theorem
- Methods
  - Supervised
    - Classification (Discriminant Analysis, Support Vector Machines, Trees, Set Covers)
    - Prediction (Regression, Trees, Neural Networks)
  - Unsupervised
    - Neural Networks
    - Clustering
    - Projections (PC, MDS, Manifold Learning)

Learning

The Curse of Dimensionality

Bellman, R. (1961). *Adaptive Control Processes: A Guided Tour*. NJ: Princeton University Press.

When the dimensionality of a space increases, the volume of the space increases so fast that points in that space become sparse.

- Local methods are less local when the dimension increases
- Neighborhoods with fixed points are less concentrated as dimension increases
- High dimensional functions tend to have more complex features than low-dimensional functions, and hence are harder to estimate
- The histogram of interpoint distances tends toward a spike
- Volume of a (unit) hypercube grows exponentially with dimensionality

This has nothing to do with computational complexity increasing with dimensionality or the difficulties of exploring in many dimensions

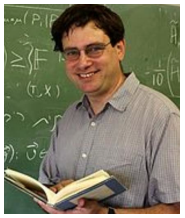
> "The curse of dimensionality is a popular way of stigmatizing the whole set of troubles encountered in high-dimensional data analysis; finding relevant projections, selecting meaningful dimensions, and getting rid of noise, being only a few of them."

Remediations

- Projections
  - Next
- Regularization
  - Next(Next)

Learning

Random Projections



Johnson-Lindenstrauss lemma

Johnson, W.B., and Lindenstrauss, J. (1984). Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics 26*. Providence, RI: American Mathematical Society, 189–206.

A set of n points in high dimensional Euclidean space can be mapped into an $O(\log n/\epsilon^2)$ -dimensional Euclidean space such that the distance between any two points changes by only a factor of $(1 \pm \epsilon)$

$$\mathbf{X}^*_{nk} = \mathbf{X}_{np}\mathbf{R}_{pk}$$

$$\mathbf{R}'\mathbf{R} \approx \mathbf{I}$$

Achlioptas, D. (2001). Database-friendly random projections. *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems - PODS '01*, 274.

Li, P., Hastie, T.J., and Church,, K.W. (2006). Very sparse random projections. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '06), 287-296.

$$r_{ji} = \sqrt{p} \begin{cases} 1 & \text{with probability } \frac{1}{2\sqrt{p}} \\ 0 & \text{with probability } 1 - \frac{1}{\sqrt{p}} \\ -1 & \text{with probability } \frac{1}{2\sqrt{p}} \end{cases}$$

Learning

Regularization

Model parameters

$$\boldsymbol{\theta} = \{\theta_j : j = 1, \cdots, p\}$$

Objective function (Loss + Regularization)

$$O(\boldsymbol{\theta}) = L(\boldsymbol{\theta}) + \Omega(\boldsymbol{\theta})$$

Loss

$$L_1 = \sum_{i=1}^{n} |y_i - \hat{y}_i| \qquad \text{(least absolute values)}$$

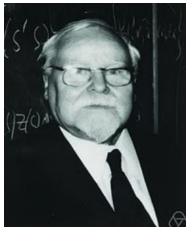
$$L_2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \qquad \text{(least squares)}$$

Regularization (penalty for complexity)

$$\Omega_1(\boldsymbol{\theta}) = \lambda ||\boldsymbol{\theta}||_1 \qquad \text{(lasso)}$$

$$\Omega_2(\boldsymbol{\theta}) = \lambda ||\boldsymbol{\theta}||^2 \qquad \text{(L2 norm)}$$

The objective function specifies a tradeoff between bias and variance



Tikhonov

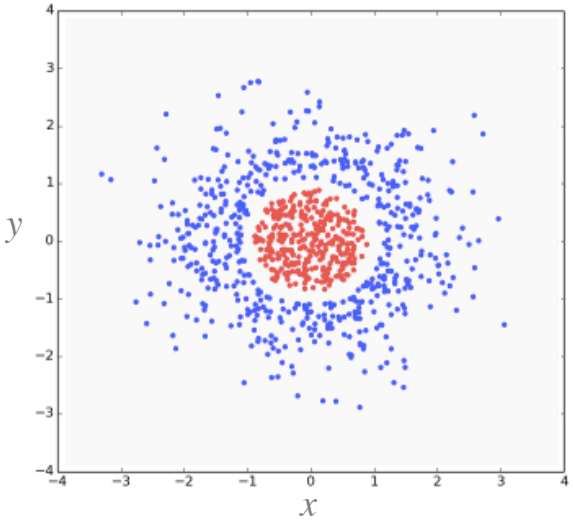
Wahba

Poggio

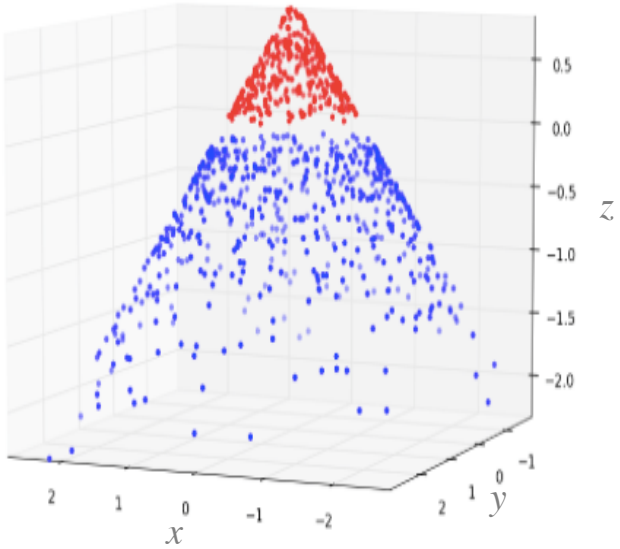
Learning

Kernels

Suppose we have the circular configurations of two sets of points below
We use the map $\mathbb{R}^2 \mapsto \mathbb{R}^3$ to get a linear boundary that separates the two sets



$$z = 1 - \sqrt{x^2 + y^2}$$

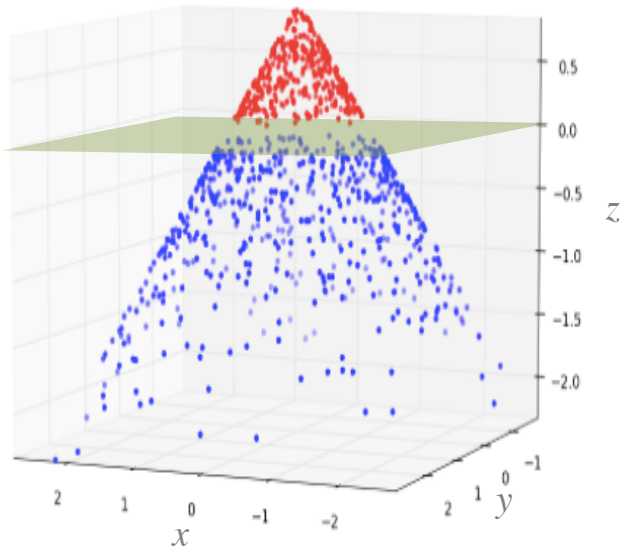


Learning

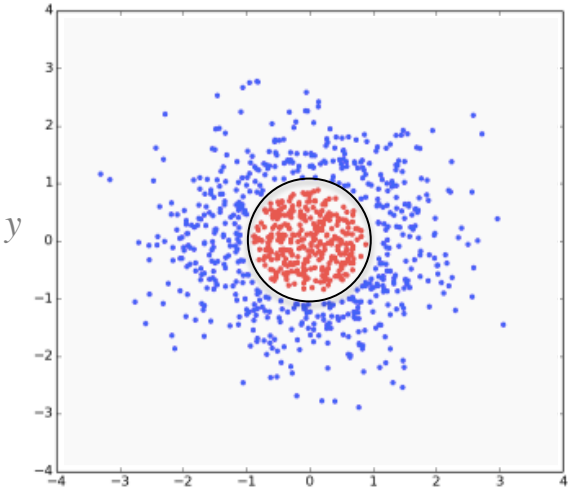
Kernels

Invert equation for decision boundary plane on left $0 = 1 - \sqrt{x^2 + y^2}$

And we get the decision boundary circle on right $x^2 + y^2 = 1$



$$x^2 + y^2 = 1$$



Learning

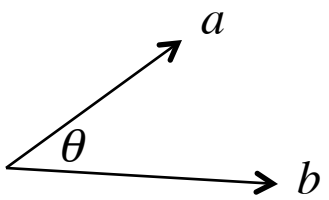
Kernels

The Kernel Trick (Hilbert, Mercer, Wahba, others)

If our algorithm employs dot products, we can use a kernel trick

The kernel works in $\mathbb{R}^2$ (or $\mathbb{R}^p$) to do what we were doing in $\mathbb{R}^3$ (or $\mathbb{R}^d$)

$$\begin{aligned}\mathbf{a} \cdot \mathbf{b} &= \mathbf{a}'\mathbf{b} \\ &= a_1 b_1 + a_2 b_2 + \cdots a_n b_n \\ &= \sum_{i=1}^{n} a_i b_i \\ &= ||a||||b|| \cos\theta\end{aligned}$$



Given two vectors $\mathbf{a}$ and $\mathbf{b}$ in $\mathbb{R}^p$,

A kernel is a function $K(\mathbf{a}, \mathbf{b})$ that implements $\mathbf{a} \cdot \mathbf{b}$ in $\mathbb{R}^d$

There is a more general formulation for high-dimensional data analysis

It involves Reproducing Kernel Hilbert Space

But we don't need it here, and it is beyond the scope of this presentation

RKHS allows infinite dimensions and can be a space of functions and abstractions

It was a major advance for high-dimensional analytics (see Donoho)

Learning

Kernels

The Kernel Trick

The kernel function saves time and space (when p << d)

But we are not home free

We still have to identify a kernel that corresponds to the function we want

Suppose our function is $\phi(\mathbf{.})$,

Then $K(\mathbf{a}, \mathbf{b}) = <\phi(\mathbf{a}), \phi(\mathbf{b})>$

Popular kernels are

Polynomial: $K(\mathbf{a}, \mathbf{b}) = (\alpha \mathbf{a}'\mathbf{b} + c)^d$

Gaussian: $K(\mathbf{a}, \mathbf{b}) = \exp\left(-\frac{||\mathbf{a} - \mathbf{b}||^2}{2\sigma^2}\right)$ (This is a radial basis function)

Sigmoid: $K(\mathbf{a}, \mathbf{b}) = \tanh(\alpha \mathbf{a}'\mathbf{b} + c)$

Learning

Bagging (Bootstrap Aggregation)

Breiman L. (1996). Bagging Predictors. *Machine Learning, 24*, 123-140.

Construct a bunch of bootstrap samples (sampling with replacement)

Fit each sample

Plurality vote determines prediction

This procedure reduces variance by aggregating

Learning

Boosting

Schapire, R.E. (1990). "The Strength of Weak Learnability". *Machine Learning,* 5, 197–227.

Train a bunch of "weak learners" (stumps, subset models, etc.)

Compute prediction accuracy for each

Combine them into aggregate prediction, weighting vote by accuracy

Result is "strong learner"

Learning

Ensembles

Hansen, L.K., Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence 12*, 993–1001.

- An ensemble is a set of base learners whose individual decisions are combined in some way, typically by weighted or unweighted voting, to classify new examples.
- Bagging and boosting are both ensemble methods
- But ensemble methods are more general
- We can combine completely different learners and benefit
- A necessary and sufficient condition for an ensemble of learners to be more accurate than any of its individual members is if the learners are relatively *accurate* and *diverse*.
- A learner is accurate if it has an error rate better than random guessing
- A set of learners is diverse if they make different errors on new data points
- This works because uncorrelated errors of individual classifiers can be reduced through averaging

Learning

Validation

A fundamental aspect of empirical science is replication

- There is no such thing as a critical experiment
- Experiments change our prior beliefs through likelihoods
  - Replication increases our confidence
  - Failure to replicate decreases our confidence
- We can never really replicate an experiment
  - Randomization works best for large samples
  - Conditions change
- But it is the best method we know
  - We do our best to identify the population from which we will sample
  - We do our best to replicate the random sampling procedure
  - We do our best to replicate the random assignment protocol
  - We do our best to use the same experimental procedures
  - We do our best to use the same analytic methods

Learning

Validation

Applying a model to a new sample shrinks goodness of fit

Wherry, R. J. (1931). A New Formula for Predicting the Shrinkage of the Coefficient of Multiple Correlation. *Annals of Mathematical Statistics, 2*, 440—457.

- Psychologists became aware of this problem in the 1960s
  - Robyn Dawes, Lou Goldberg, Paul Slovic, Lee Cronbach, Amos Tversky, …
  - The shrank their R^2 values with Wherry's formula when presenting models
  - Soon thereafter they did cross validation for other models
  - They used cross validation to guard against over-fitting
- Cross validation
  - Split a sample in half
    - Early in the game, they used first half-second half
    - But this risked bias, so they did a random split
  - Fit the model based on the first (training set) to the second (test set)
  - The empirical error on the test set is an estimate of model generalization

Learning

Cross Validation

Types

- Split-half
- Leave one out (impractical)
- K-fold CV (popular)
  - Split file into k pieces
  - For each k, train on other k-1 pieces, test on the kth
  - Average k goodness-of-fit statistics

Problems

- What is the population?
  - This is a major problem for Big Data
  - Other researchers testing (replicating) your method can't use your data
    - They can't find another dataset from the same population
    - because yours was a convenience batch (you had the whole population)
- K-fold CV is not replication (in the same sense that scientists use the word)
  - Yu, B. (2013). Stability. *Bernoulli, 19*, 1484-1500.
- Researchers often use CV to select best model or optimize parameter values

Learning

Accuracy

How do we assess the accuracy of a model on a set of data?

Confusion matrix (2 categories)

		Predicted: Yes	Predicted: No
Actual	**Yes**	True Positive	False Negative
	No	False Positive	True Negative

Tally each cell and compute error rate from tallies

Overall error rate is falses divided by totals

Problems

When table is unbalanced, interpretation of overall error is difficult

This problem has a long history in the statistics of tables

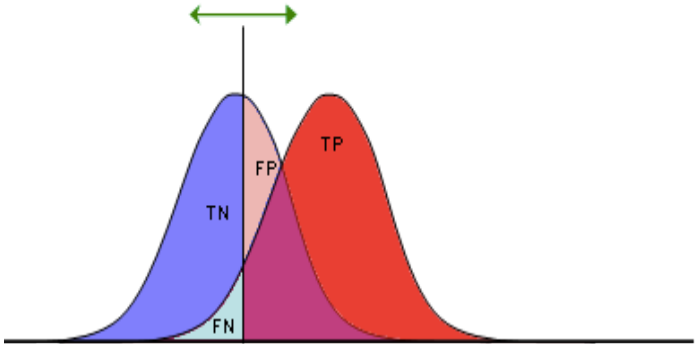
There are well-known corrections (Cohen's kappa), but they are not widely used by ML people

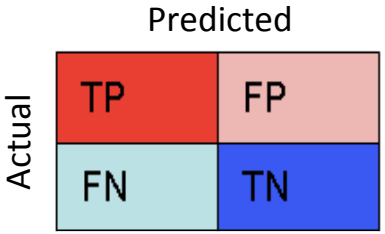
Learning

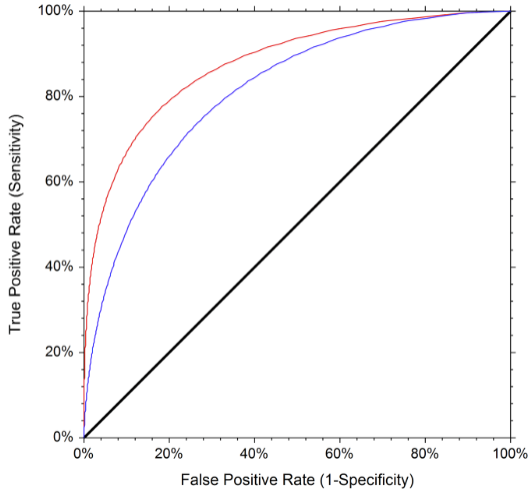
Accuracy

The Receiver Operating Characteristic (ROC) curve

- Sensitivity is Predicted Positives divided by Actual Positives
- Specificity is Predicted Negatives divided by Actual Negatives
- Plot sensitivity (hit rate) against 1–specificity (false alarm rate)
- Changing threshold in classifier alters positions on curve (step function)
  - High thresholds reject almost everything (southwest)
  - Low thresholds accept almost everything (northeast)







Adapted from Wikipedia and NCSS

Learning

Accuracy

Area Under the Curve (AUC)

Computing area under ROC curve allows comparisons of different classifiers

But

Hand, D. (2009). Measuring classifier performance: a coherent alternative to the area under the ROC curve. *Machine Learning, 77*, 103–123.

AUC is equivalent to using different metrics to evaluate different classification rules.

i.e., using one classifier, misclassifying a class 1 point is p times as serious as misclassifying a class 0 point, but, using another classifier, misclassifying a class 1 point is q times as serious, where $p \neq q$.

Learning

No Free Lunch Theorem

Wolpert, D.H., Macready, W.G. (1997)

If an algorithm performs well on a certain class of problems then it necessarily pays for that with degraded performance on the set of all remaining problems

Learning

- Supervised Learning
  - Classification
    - Vast field
    - Given a set of categories and associated metadata, predict categories
    - Chief methods
      - Linear Discriminant Analysis (LDA)
      - Quadratic Discriminant Analysis
      - Support Vector Machines
      - Decision Trees
      - Random Forests
      - Set covers

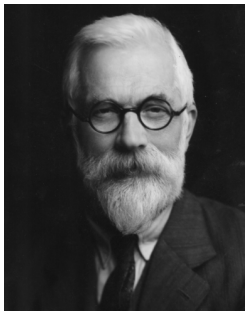
Learning

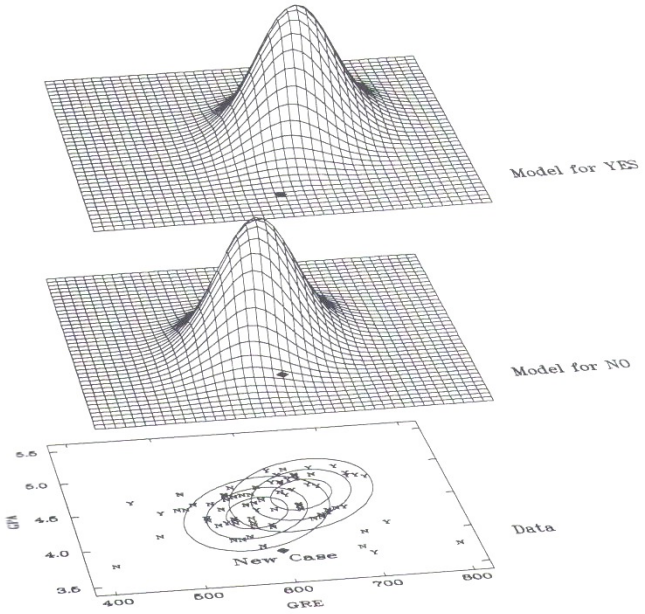
Supervised Learning

Classification

Fisher's Linear Discriminant Function– Two groups

Covariance matrices assumed to be the same multivariate normal





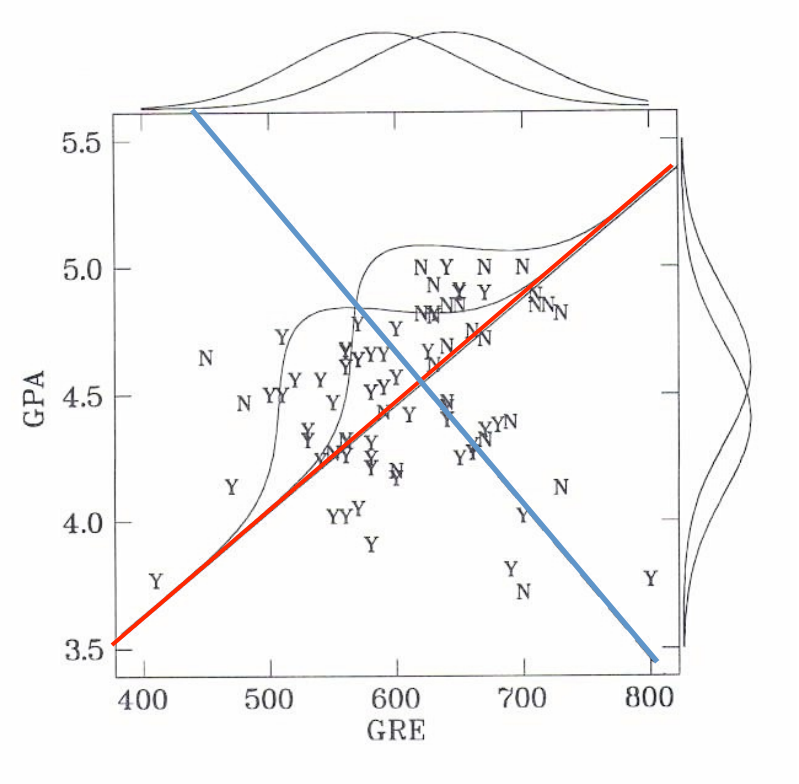
Wilkinson, Blank & Gruber, 1996

Learning

Supervised Learning

Classification

Linear Discriminant Analysis



$$\begin{aligned}
\mathbf{B} &= \text{between groups covariance matrix} \\
b_{ij} &= \frac{1}{g-1}\sum_{k=1}^{g} n_k(\bar{x}_{ik}-\bar{x}_i)(\bar{x}_{jk}-\bar{x}_j) \\
\mathbf{W} &= \text{within groups covariance matrix} \\
\mathbf{W} &= \frac{1}{n-g}\sum_{k=1}^{g}(n_k-1)\mathbf{S}_k
\end{aligned}$$

$$\max_{\mathbf{v}} \lambda = \frac{\mathbf{v}'\mathbf{B}\mathbf{v}}{\mathbf{v}'\mathbf{W}\mathbf{v}}$$

$$(\mathbf{B}-\lambda\mathbf{W})\mathbf{v} = \mathbf{0}$$

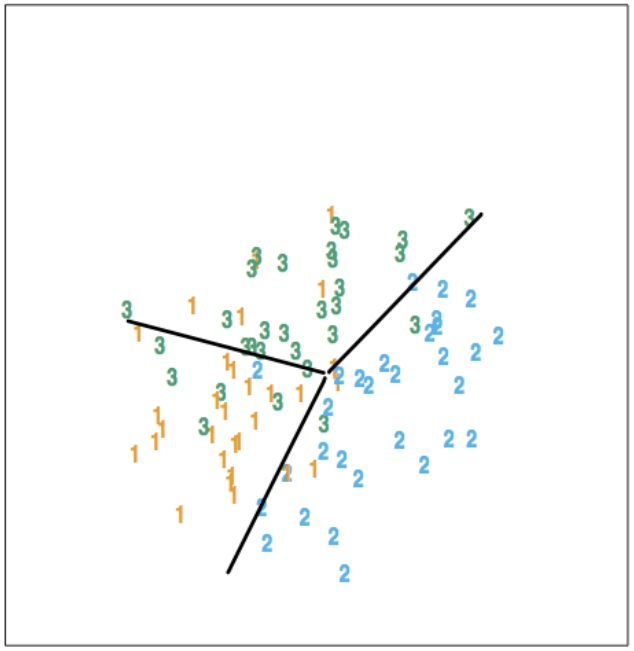
Generalized eigenvalue problem

Learning

Supervised Learning

Classification

Linear Discriminant Analysis – Three groups



Hastie, Friedman & Tibshirani, 2011

Learning

Supervised Learning

Classification

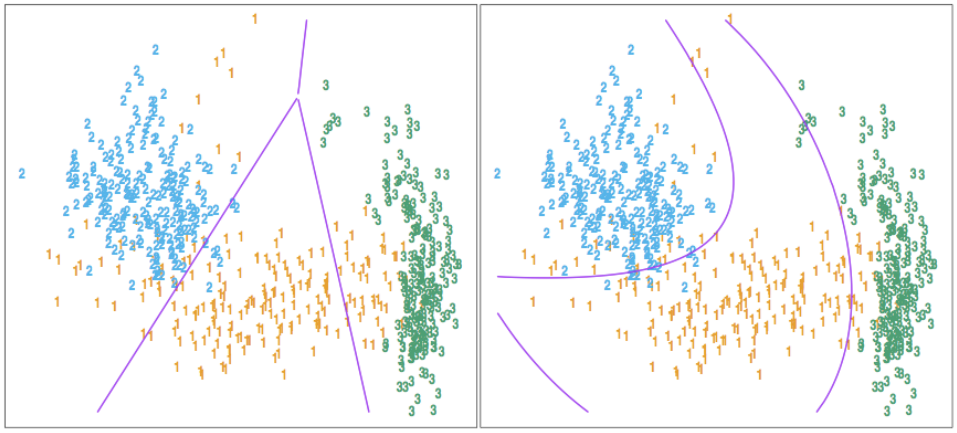
Quadratic Discriminant Analysis (QDA) – Three groups

This example shows fits assuming covariance matrices are the same

Obviously violated in this case

QDA fits separate covariance matrices

But often it is not needed because adding quadratic terms to linear model suffices



Hastie, Friedman & Tibshirani, 2011

Learning

Supervised Learning

Classification

Quadratic Discriminant Analysis (QDA) – Three groups

Here's what Hastie, Friedman, Tibshirani have to say:

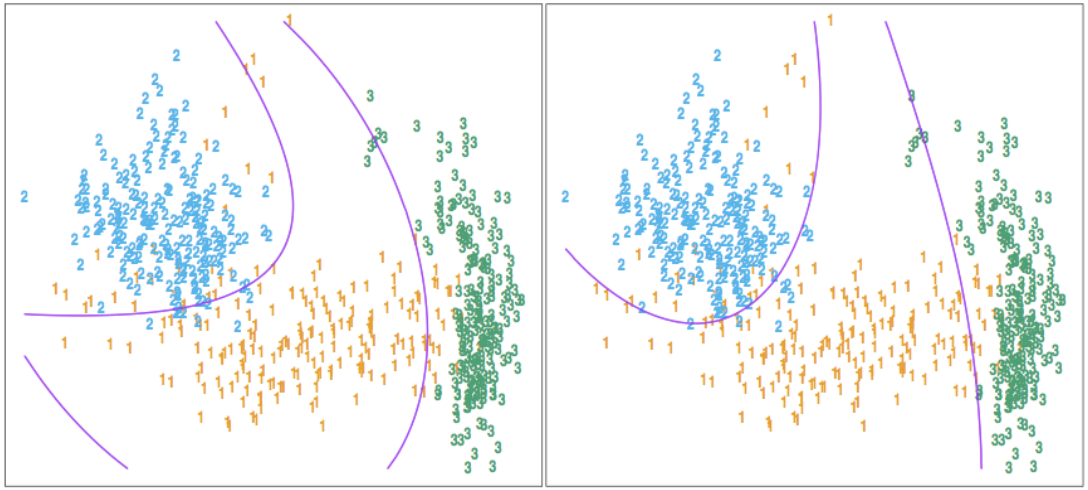


FIGURE 4.6. *Two methods for fitting quadratic boundaries. The left plot shows the quadratic decision boundaries for the data in Figure 4.1 (obtained using LDA in the five-dimensional space $X_1, X_2, X_1X_2, X_1^2, X_2^2$). The right plot shows the quadratic decision boundaries found by QDA. The differences are small, as is usually the case.*

Learning

Supervised Learning

Classification

Simple parametric discriminant models LDA and QDA

Both LDA and QDA perform well on an amazingly large and diverse set of classification tasks. For example, in the STATLOG project (Michie et al., 1994) LDA was among the top three classifiers for 7 of the 22 datasets, QDA among the top three for four datasets, and one of the pair were in the top three for 10 datasets. Both techniques are widely used, and entire books are devoted to LDA. It seems that whatever exotic tools are the rage of the day, we should always have available these two simple tools. The question arises why LDA and QDA have such a good track record. The reason is not likely to be that the data are approximately Gaussian, and in addition for LDA that the covariances are approximately equal. More likely a reason is that the data can only support simple decision boundaries such as linear or quadratic, and the estimates provided via the Gaussian models are stable.

Hastie, Friedman & Tibshirani, 2011

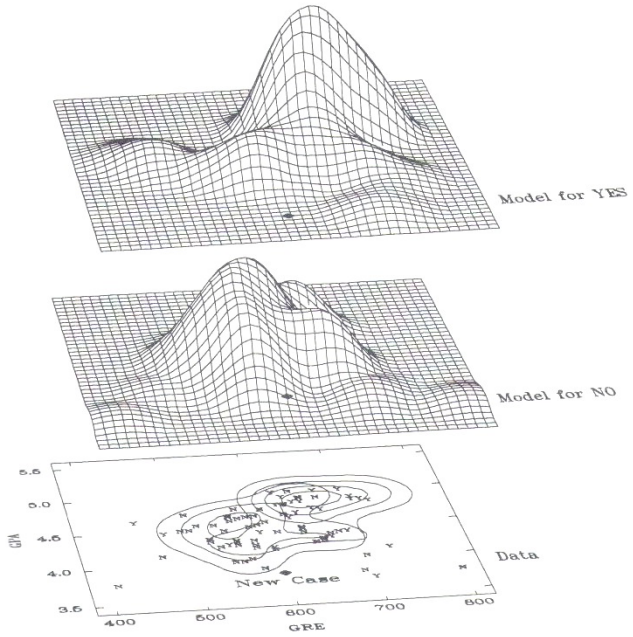
Amen

Learning

Supervised Learning

Classification

Kernel Discriminant Analysis



Wilkinson, Blank & Gruber, 1996

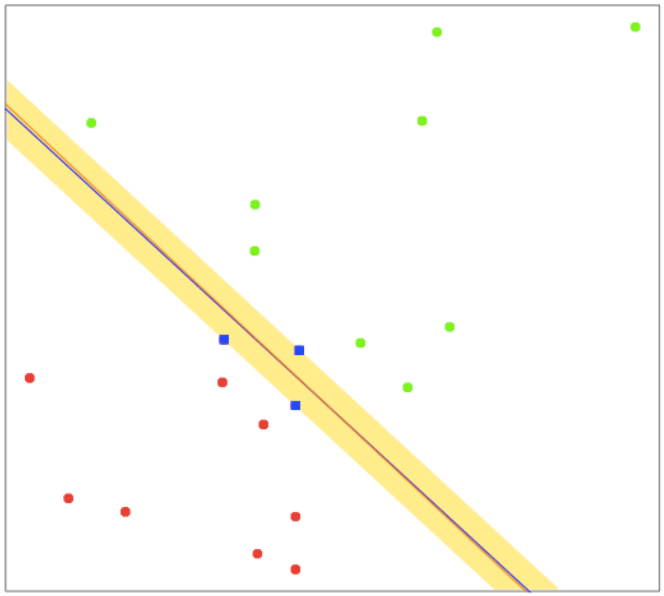
Learning

Supervised Learning

Classification

Support Vector Machines (SVM)

Vapnik, V. N. (1998). *Statistical Learning Theory*. John Wiley & Sons.



Aim for wide margin separating 2 classes
Focuses estimation on points near margin
But less effective on Gaussian data
Vladimir Vapnik devised this formulation
But real power comes from pairing SVM with kernels

Hastie, Friedman & Tibshirani, 2011

Learning

- Supervised Learning
  - SVM with kernels was the darling of ML people for the last decade
    - It's mathematically appealing
    - It has spawned endless papers offering minor refinements
    - It leverages kernels (but other methods can too)
  - But there are issues
    - Picking kernels and parameters is a black art
      - Proponents try to use cross-validation to do this automatically
        - That increases the possibility of overfitting
    - SVMs are slow and a pig on memory
    - SVMs work only with pairs of classes
      - Proponents have developed one-against-all modifications
        - This increases complexity
    - SVMs have not been found to outperform other classifiers
      - Random forests and logistic regression trees often do better

Learning

Supervised Learning

Decision trees (Recursive Partitioning)

Automatic Interaction Detection (AID)

Morgan, James N. and Sonquist, John A. (1963). Problems in the analysis of survey data, and a proposal. *Journal of the American Statistical Association 58*, 415-435.

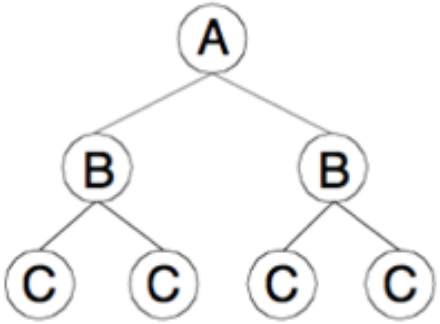
The authors presented AID as a method for analyzing survey data

The I in AID referred to interactions because it represented them directly in a tree

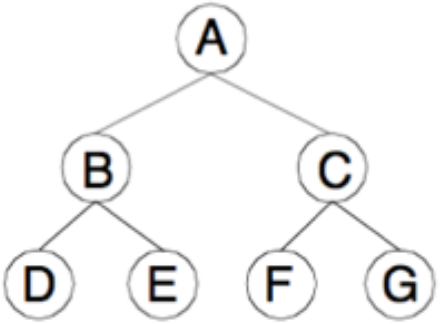
The idea was to eliminate non-significant interactions in ANOVA models

A VERY clever idea, for which they do not get sufficient credit

No Interaction



Interaction



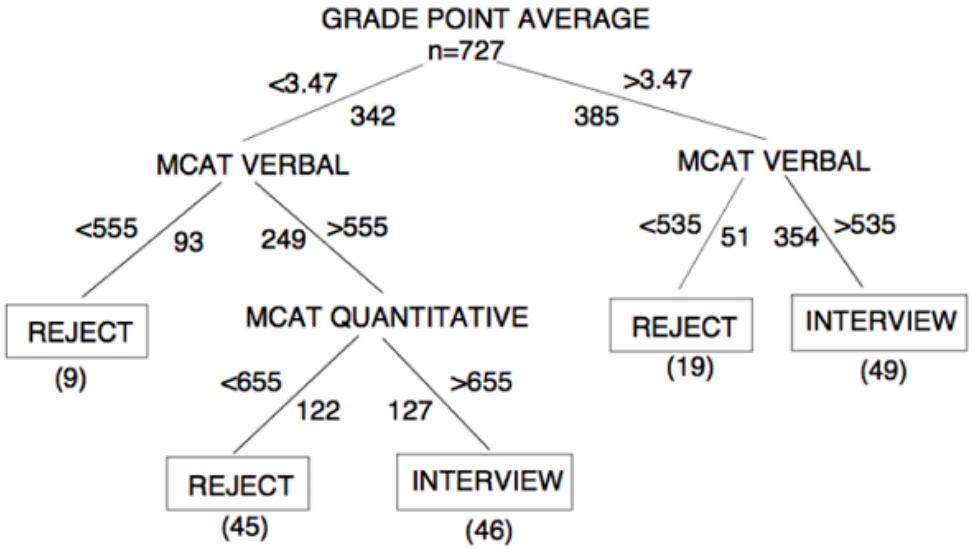
Learning

Supervised Learning

Decision trees

Automatic Interaction Detection (AID)

Predicting admission decisions at Yale Medical School



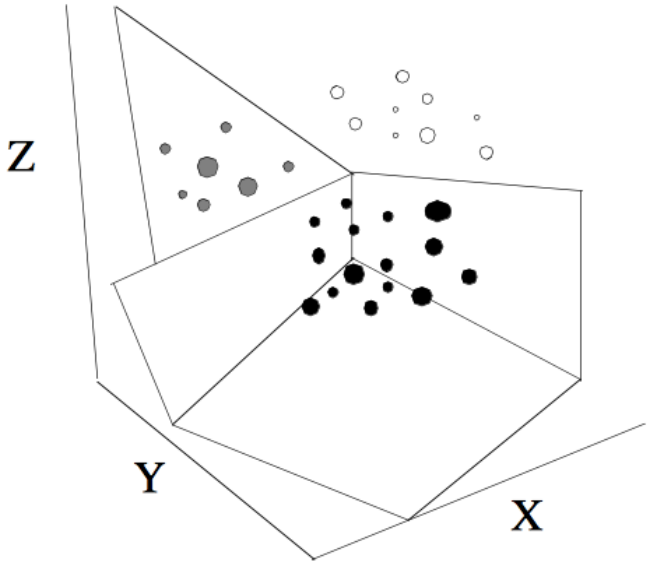
Milstein, Burrow, Wilkinson, Kessen (1975)

Learning

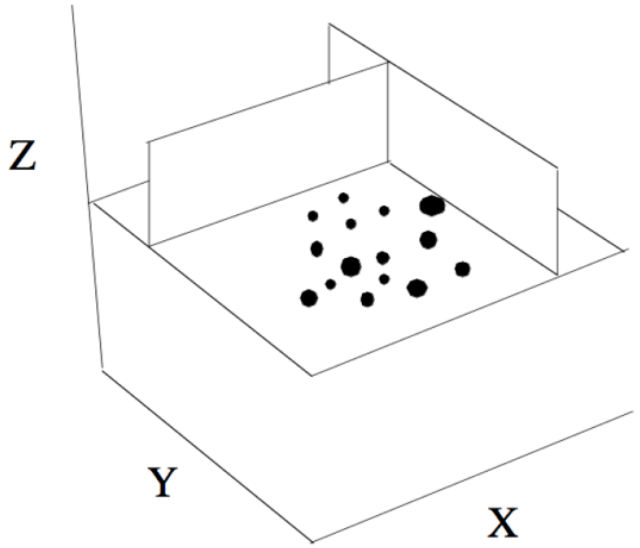
Supervised Learning

How decision trees carve up space

Linear classifier



Recursive partitioner



Learning

Supervised Learning

Decision trees

Splitting functions

Predictors

Dependent

	Categorical	Continuous
Categorical	Phi-square	Phi-square
Continuous	SSWithin	Least Squares

Impurity measures

- Gini
- Twoing
- Entropy / chi-square

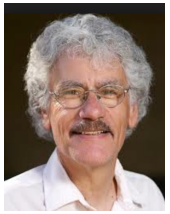
Categorical predictors

- Need to consider every combination of categories
  - EXPENSIVE!
  - Cheesy alternative is to scale a predictor by sorting

Learning

Supervised Learning

Decision tree programs

AID

Sonquist, J. A., Morgan, J. N. (1964). The Detection of Interaction Effects. Survey Research Center, University of Michigan.

CHAID

Kass, G.V. (1980) An Exploratory Technique for Investigating Large Quantities of Categorical Data, *Applied Statistics, 29*, 119–127.

Chi-square AID

Multinomial splits at single node

CART

Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J.(1984). *Classification and Regression Trees*. Belmont, CA: Wadsworth.

Introduced portfolio of loss functions and gave statistical grounding to AID

Introduced Pruning and Classification

ID3/C4.5/C5.0

Quinlan, J.R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

Derivative of CART

Learning

Supervised Learning

Visualizing Decision trees

Mobiles

Wilkinson, L. (1992). Tree Structured Data Analysis: AID, CHAID and CART. Sun Valley, ID: Sawtooth/SYSTAT Joint Software Conference.

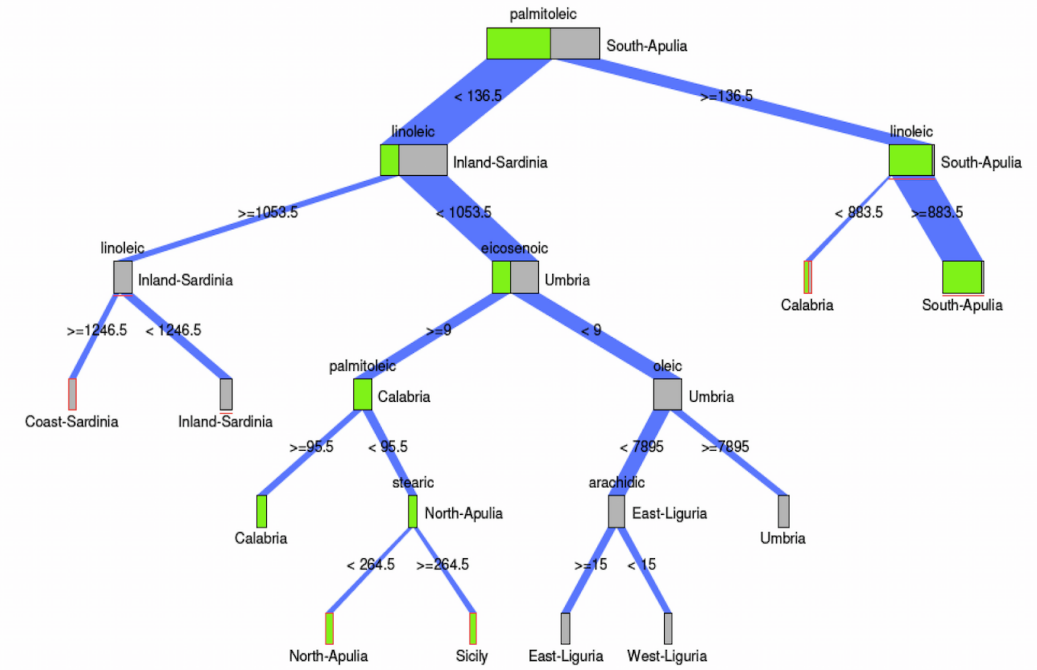
Learning

Supervised Learning

Visualizing Decision trees

Vach, W. (1995). Classification trees. *Computational Statistics, 10*, 9–14.

Urbanek, S. (2003). Many faces of a tree. In *Computing Science and Statistics: Proceedings of the 35th Symposium on the Interface*.



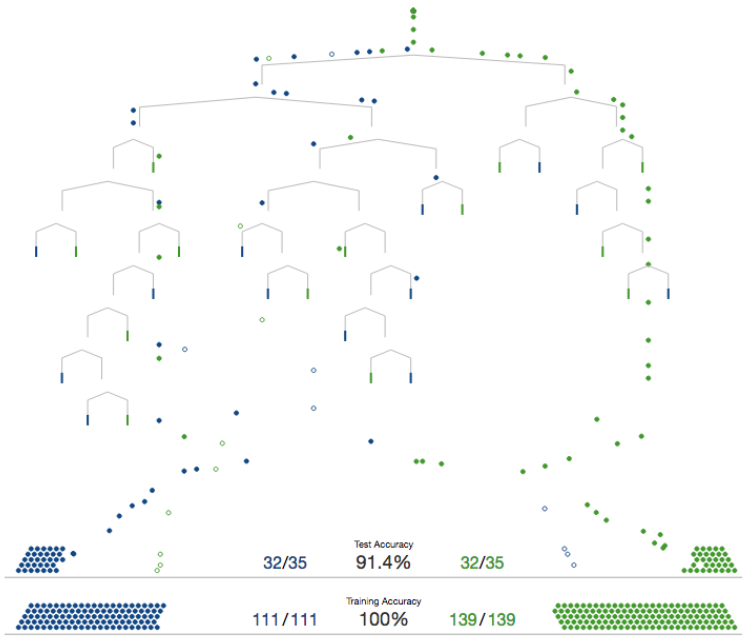
Learning

Supervised Learning

Visualizing Decision trees

Check out this site done by one of our most talented young designers and his statistician partner.

http://www.r2d3.us/visual-intro-to-machine-learning-part-1/



Learning

Supervised Learning

Decision Tree Pros

- Handles continuous, categorical, and ordered variables in one model
- Greedy and fast
- Invariant over monotone transformations of predictor variables
- Robust to outliers
- Missing values handled intrinsically
- Easy to interpret

Cons

- High variance
  - Initial splits lead to very different trees
  - Error at top is propagated down tree
- Greedy
  - Best fitting tree may not be found (similar to a local minimum)
- Pruning methods vary and involve chance of over/under fitting

Learning

Supervised Learning

Random Forests

Breiman L. (2001). Random forests. *Machine Learning, 45*, 5-32.

Follow bagging procedure

- Construct a bunch of bootstrap samples (sampling with replacement)
- Fit tree to each sample
- Plurality vote determines class prediction

BUT

- At each split for a given sample, choose a random subset of the predictors
  - Breiman called resulting trees "stumps"
  - This reduces correlation of trees across samples due to powerful splitters in early splits

RF may be the most powerful ML prediction method

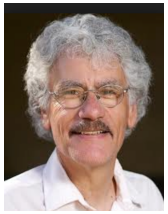
- Breiman claimed it was
- Surveys show it (or variants) does beat other ensembles

Learning

Supervised Learning

Gradient Boosted Trees

Friedman, J.F. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics, 29*, 1189–1232.



Build a series of stumps

Build each stump from residuals of previous fit

Stochastic boosting randomly samples from residuals at each step

Learning

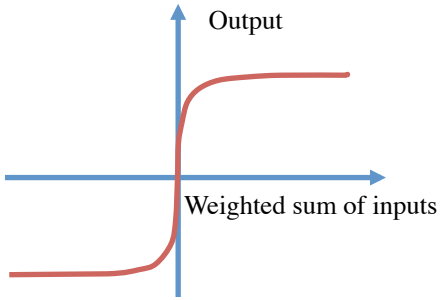
Supervised/Unsupervised Learning

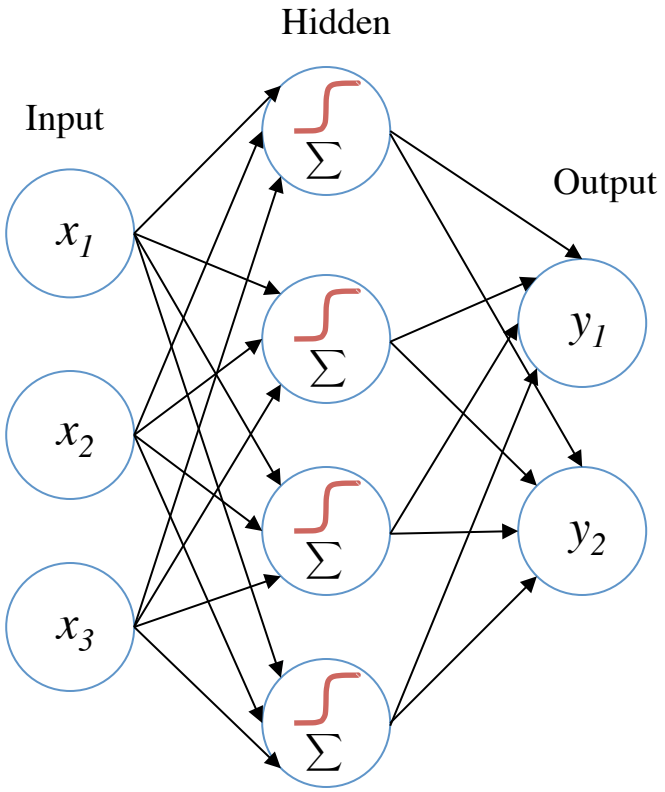
Neural Networks

Psychologists, Biologists, Statisticians

McCulloch, Hebb, Rumelhart, Hinton, …

Sigmoid activation function





negative input values lower the sum and suppress the neuron

positive input values increase the sum and cause the neuron to fire

Fundamental model is sums of nonlinearly transformed linear models

Learning

Supervised/Unsupervised Learning

Neural Networks

- Of all the machine learning algorithms, NNs are the most “black box”
  - They are really just another nonlinear algebraic statistical model
  - The sigmoid activation function introduces a wider class of models
  - If the activation function is an identity, then we just have a set of linear models
  - NN make their own features, rather than being fed a fixed set
- Deep learning models are networks with more than one hidden layer
- Fitting weight parameters (on each edge of the graph) done by various methods
  - Most popular is back propagation, but this is slow
- There is a danger of overfitting
  - So regularization is frequently employed (adding a penalty)
- Like SVMs, this is a black art
  - Don’t try this at home, folks