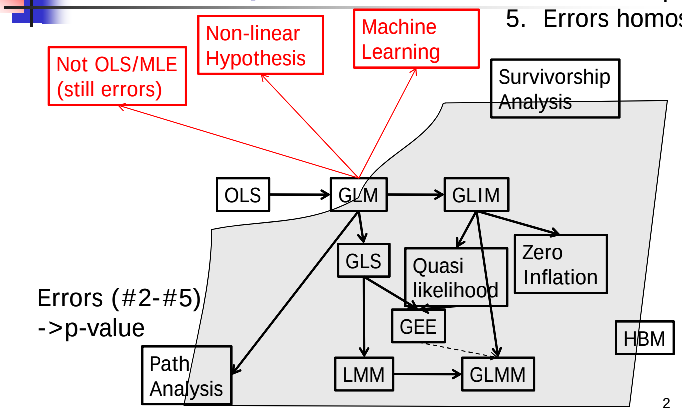
Modern Regression

Road map

1. Linear
2. Error in Y only
3. Errors normal
4. Errors independ
5. Errors homosce



2

Modern regression

- Three goals: prediction, significance, summarization
- Focus on prediction/summarization – not significance tests
  - Some have significance tests
  - Usually through bootstrapping/Monte Carlo

Modelling

- y=f(x,β)+noise
  - No formal error model (e)
  - y non-linear, general
    - GLIM had picklist (log, logit)
    - GLIM had linear interactions between x - xβ
- Goals
  - Prediction if know f, can predict y for new x
  - Model assessment (→mechanism??)
  - Observational, not experimental

Types

- Error still
  - Robust (don't overweigh outliers)
  - Quantile – don't focus on central tendencies
- Nonlinear hypothesis
  - Nonlinear
  - Piecewise
- Machine learning
  - Smoothing
  - GAM
  - MARS
  - Neural Net
  - CART

Types

- Error still
  - Robust (don't overweigh outliers)
  - Quantile – don't focus on central tendencies
- Nonlinear hypothesis
  - Nonlinear
  - Piecewise
- Machine learning
  - Smoothing
  - GAM
  - MARS
  - Neural Net
  - CART

Robust regression

- "The problem with likelihood is that outliers are just too unlikely"
- Minimizing $\Sigma(x-\mu)^2$ gives special emphasis to more distant points
  - Appropriate if truly from normal
  - Real world is often a mixture – 95% normal, 5% totally unrelated (aka outliers)
- We studied GLM outliers and had only two choices keep them in or throw them out

Robust regression II

- Robust regression provides alternatives
- M-estimation
  - Minimize $\Sigma w_i f(\varepsilon_i/k)$
  - $k=c*s*\text{sqrt}(1-h)$
    - c=method specific tuning constant
    - s=nonparametric estimate of s (MAD/0.6745)
    - h=leverage

	W	f
LS	w=1	$f(\varepsilon)=\varepsilon^2$
LAD	w=1	$f(\varepsilon)=\|\varepsilon\|$
Huber	1 if $\|\varepsilon\|<k$ $1/\|\varepsilon\|$ if $\|\varepsilon\|>k$	½ ε^2 if $\varepsilon<k$ ~0 if $\|\varepsilon\|>k$
Bisquare	1 if $\|\varepsilon\|<k$ 0 otherwise	$(1-r^2)^2$

Robust III

- LTS (least-trimmed squares)
  - Least squares but throw out bigger ε_ι
  - Use smallest n/2+(k+2)/2 ε_ι
- In R

```
## LAD =quantile w/ tau=50% - see next section
## M-estimation
library(MASS)
#M-estimate  #M=Huber (default), MM=Bisquare
m=rlm(dep~indep,data=?,sub=?,method="M"|"MM")
plot(m$w)
identify(1:length(dep),m$w,rownames(data) # x,y,text
##LTS (also in MASS)
m=ltsreg(formula,…)
```

Types

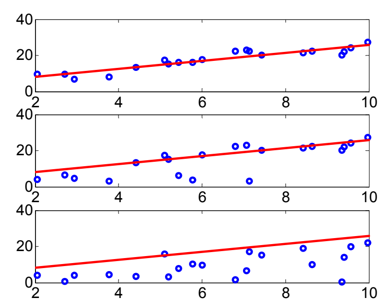
- Error still
  - Robust (don't overweigh outliers)
  - Quantile – don't focus on central tendencies
- Nonlinear hypothesis
  - Nonlinear
  - Piecewise
- Machine learning
  - Smoothing
  - GAM
  - MARS
  - Neural Net
  - CART

Quantile regression

- Article for ecologists on syllabus page
- GLM (OLS) estimates the mean (~50% percentile) of the data
- What if we want to estimate the 90%-percentile or the 20%-percentile lower bound?
- Called quantile regression
- Two questions to explore:
  - Does slope change with quantile
  - Find the envelope

Envelope phenomenon

- Common in ecology to see one variable explain the upper or lower envelope of another variable

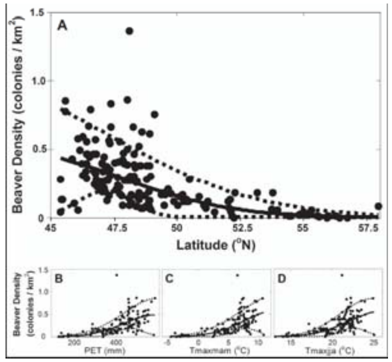


All sites limited by x

Most sites limited by x

Most sites limited by other than x
But still constrained by x

From my research



Quantile in R

- library(quantreg) #must download from CRAN
  - data(Mammals)
- Linear quantiles (lines)
  - plot(y~x,data=?)
  - m=rq(y~x,data=?,tau=0.5)
  - abline(m)
- Nonlinear spline quantiles
  - m=lprq(x,y,h=2,tau=percentile)
  - #h gives smoothing try, 1,3, 4 also
  - #doesn't take data=?
  - lines(mxx,mfv,lty=2)
- A priori nonlinear in homework

```
m=lprq(log10(Mammals$weight),Mammals$speed,h=2,tau=0.9)
plot(speed~log10(weight),data=Mammals)
lines(m$xx,m$fv)
```

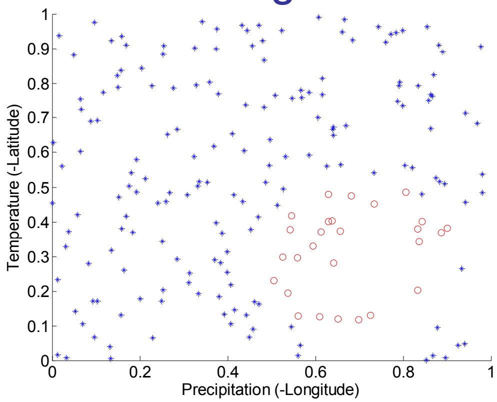
Types

- Error still
  - Robust (don't overweigh outliers)
  - Quantile – don't focus on central tendencies
- Nonlinear hypothesis
  - Nonlinear
  - Piecewise
- Machine learning
  - Smoothing
  - GAM
  - MARS
  - Neural Net
  - CART

Nonlinear regression

- Lots of relationships are non-linear
- Polynomials (quadratic,cubic) capture some of this
- Polynomials don't have asymptotes which are common in nature

Example problem: niche modelling

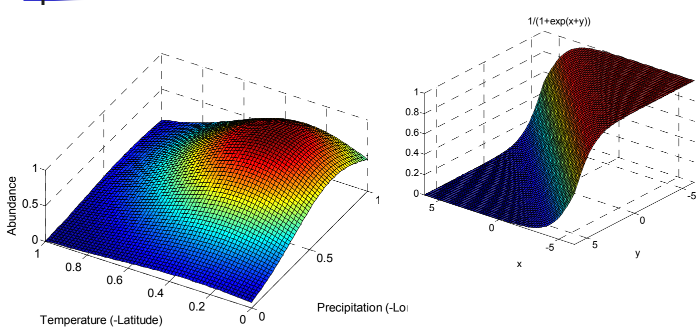


Effectiveness of traditional f values?

- Linear regression (abundance or coded presence/absence)
  - Get a plane that rises towards one corner (dependent on sample domain)
- Linear discriminant analysis (presence/absence)
  - Get a dividing line that divides off one corner (dependent on sample domain)
- Logistic (presence/absence) (NEXT SLIDE)
  - Surface with logistic curve in 1-D parallelling line in discriminant analysis

- Also a problem w/ linear combination of x – clear interaction here

Continuous dependent variable



Nonlinear regression

- Standard approach is to minimize sum squares
- Must be solved iteratively (by computer)
- Must give the iterator a starting point
- Confidence intervals
  - Often asymmetric
  - Default method just uses standard errors & symmetric CI
  - Preferred method #1
    - Uses a method similar to likelihood ratio
    - Vary parameter, calculate sum-squares, F
  - Preferred method #2
    - Bootstrapping – covered later
- Don't forget many nonlinear cases can be transformed to linear

Nonlinear regression in R

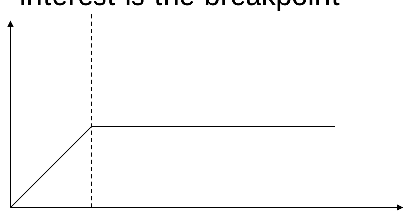
```
plot(speed~log10(weight*100),data=Mammals)
mnl=nls(speed~a*log10(weight*100)/(half+log10
  (weight*100)),start=list(a=60,half=0),trace
  =T,data=Mammals)
abline(mnl) # doesn't work only does straight
  lines
x=10^seq(-3,4,0.2)
lines(x,predict(mnl,data.frame(weight=x)))
plot(m), summary(m), etc
```

Types

- Error still
  - Robust (don't overweigh outliers)
  - Quantile – don't focus on central tendencies
- Nonlinear hypothesis
  - Nonlinear
  - Piecewise
- Machine learning
  - Smoothing
  - GAM
  - MARS
  - Neural Net
  - CART

Piecewise linear regression

- Several lines with breakpoints
  - 5 parameters for one breakpoint
  - Only recommended when main variable of interest is the breakpoint



Piecewise

- Fit multiple linear models over different domains
  - Approximation to nonlinear – bad reason
  - A priori hypothesis of threshold effect – good reason

In R

```
library(quantreg)
lata(Mammals)
mammals$logWeight=log10(Mammals$weight*100)
m.lm<-lm(speed~logWeight,data=Mammals)
m.seg<-segmented(m.lm,seg.Z=~logWeight,
  psi=list(logWeight=c(4)))
m.Seg
summary(m.seg)
confint(m.seg)
plot(speed~logWeight,data=Mammals)
plot(m.seg,add=T)
```

Types

- Error still
  - Robust (don't overweigh outliers)
  - Quantile – don't focus on central tendencies
- Nonlinear hypothesis
  - Nonlinear
  - Piecewise
- Machine learning
  - Smoothing
  - GAM
  - MARS
  - Neural Net
  - CART

Statistical learning

- Let the computer figure out the function, f
  - Smooth regression (spline/local)
  - GAM (General Additive Model)
  - MARS (Multivariate Adaptive Regressive Splines)
  - Neural Networks
  - CART (Classification & Regression Tree)
- All have EXTREMELY different ideas of f
  - But all predictive y=f(X)
  - All automatically handle non-linearities & interactions
  - They vary greatly on a spectrum from black-box to human-interpretable

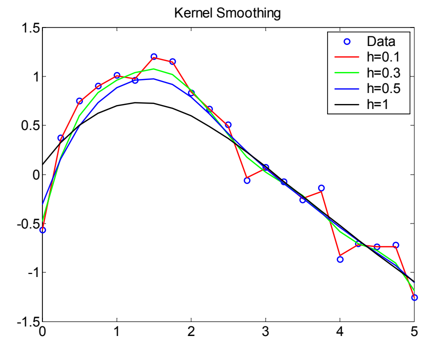
Why not just use nonlinear

- Computational issues – more complex functions may never converge
- Must have a priori expectation of nonlinearities & interactions
- May not have these in three cases
  - Data mining – data more abundant than understanding, looking for patterns as a starting point
  - Complexity – so many factors involved it is hard to imagine the nature of a model involving all factors
  - Rapidity – could build model with a couple of years of research but need results now (conservation applications)
- All apply to our example case

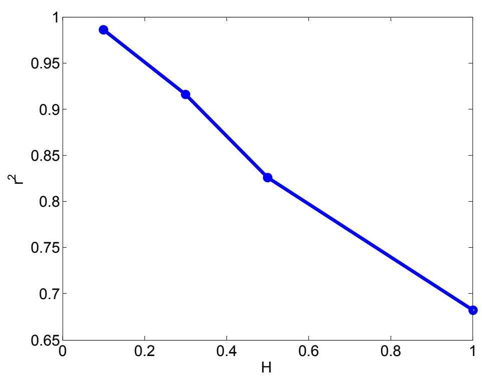
Overtraining

- Challenge – more parameters leads to better fit
  - Enough parameters leads to perfect fit to noisy data
  - E.g. polynomial of order n-1 fits n data points perfectly

Kernel smoothing



Effect of parameter H on r^2

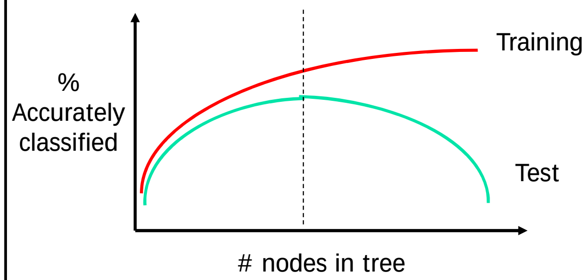


Cross-validation

- Split data into two categories:
  - Training
  - Test
  - Often 2/3 randomly into training
- Run the tree on training data
- Evaluate accuracy on test
- Parallels with bootstrapping
- Can be used on even linear regression

Optimal complexity

- Plot accuracy vs. complexity, pick optimum accuracy



Types

- Error still
  - Robust (don't overweigh outliers)
  - Quantile – don't focus on central tendencies
- Nonlinear hypothesis
  - Nonlinear
  - Piecewise
- Machine learning
  - Polynomial
  - Smoothing
  - GAM
  - MARS
  - Neural Net
  - CART

Polynomial surface

- Fit a quadratic, cubic, quartic (don't go higher)
- Usually include interaction terms (e.g. $x*y^2$)
- Use lm command

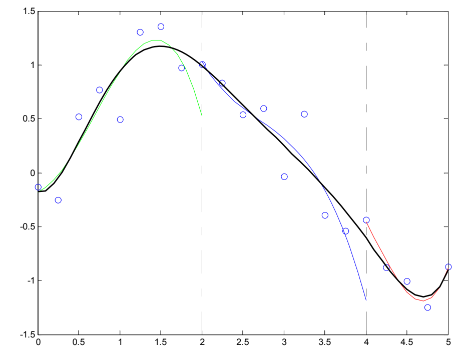
Types

- Error still
  - Robust (don't overweigh outliers)
  - Quantile – don't focus on central tendencies
- Nonlinear hypothesis
  - Nonlinear
  - Piecewise
- Machine learning
  - Smoothing
  - GAM
  - MARS
  - Neural Net
  - CART

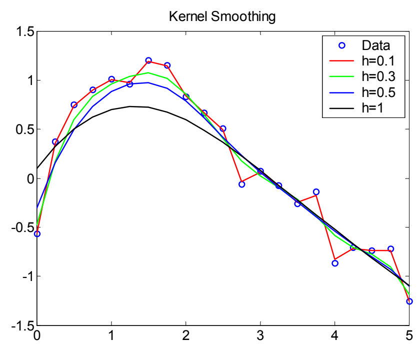
Smooth (aka nonparametric or local) regression

- The high-tech equivalent of drawing a fitting curve by hand
  - High-order polynomials
  - Cubic splines (NEXT SLIDE)
    - http://www.wam.umd.edu/~petersd/interp.htm
  - Local regression (LOWESS)
    - Create list of points evenly spread, take a window around each such point, weight data w/in window by distance from point, get local slope from linear regression
  - Kernel
    - Similar but no window, weight data by a kernel such as Gaussian
  - Human interpretable only up to 2 dimensions (visual)
  - Many of these have a smoothing parameter that ranges from perfect fit to the data to perfectly smooth (e.g. polynomial degree)

High order polynomial (6) vs piecewise splines (3)



Kernel smoothing



Smooth regression in R

```
#traditional formula based
m=loess(formula,span=0.75,data=?)
#shortcut for plotting
lines(loess.smooth(x,y))
ysmooth=ksmooth(x,y,bandwidth=0.5)
library(KernSmooth)
ysmooth=locpoly(x,y,bandwidth=0.5)
```

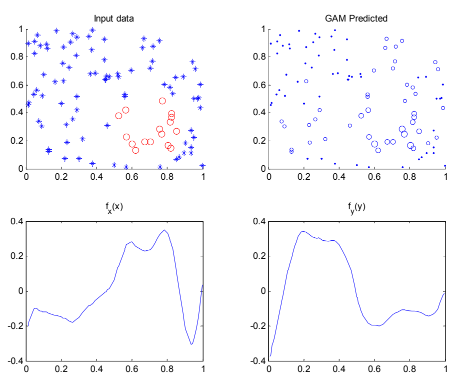
Types

- Error still
  - Robust (don't overweigh outliers)
  - Quantile – don't focus on central tendencies
- Nonlinear hypothesis
  - Nonlinear
  - Piecewise
- Machine learning
  - Smoothing
  - GAM
  - MARS
  - Neural Net
  - CART

GAM – General Additive model

- $y=a+f(x_1)+f(x_2)+f(x_3)+\ldots$
  - Or logit: $\log(y/(1-y))= a+f(x_1)+f(x_2)+f(x_3)+\ldots$
- Nonlinearity but only additive interaction
- Highly interpretable (look at f for each variable)
- What to use for f??
  - Smooth regression (e.g. cubic splines)

GAM result



GAM in R

```
library(mgcv)
m=gam(y~s(x)+s(y)+lo(z)+disc,…)
plot(m) # plots functions

mgam=gam(speed~s(log10(weight))+hoppers
  ,data=Mammals)
plot(mgam)
summary(mgam)
```

Types

- Error still
  - Robust (don't overweigh outliers)
  - Quantile – don't focus on central tendencies
- Nonlinear hypothesis
  - Nonlinear
  - Piecewise
- Machine learning
  - Smoothing
  - GAM
  - MARS
  - Neural Net
  - CART

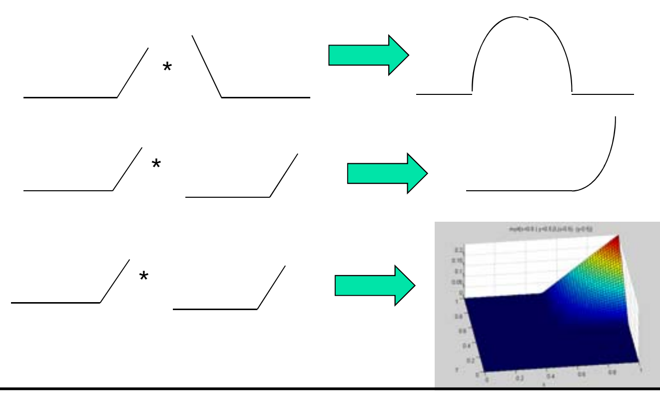
MARS

- Variation on GAM
  - Don't allow spline f, use specific f for each x_i

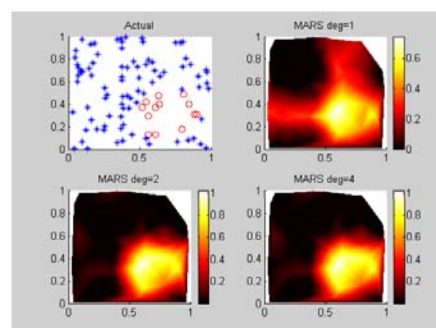
$f_i(x_i)=$

  - Do allow products (interaction)
  - $y=a + \Sigma\, b_i \_/_i(x_i) + \Sigma\, b_{ij} \_/_i(x_i)\_/_j(x_j) + \ldots$
- The region of zero allows for very local fitting
- The multiplication allows for nonlinearity & interaction
- MARS is human interpretible??

Multiplication → nonlinearity, interaction



MARS result



MARS in R

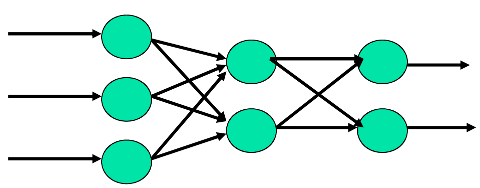
- library(mda)
- m=mars(xmatrix,y,degree=n)

Types

- Error still
  - Robust (don't overweigh outliers)
  - Quantile – don't focus on central tendencies
- Nonlinear hypothesis
  - Nonlinear
  - Piecewise
- Machine learning
  - Smoothing
  - GAM
  - MARS
  - Neural Net
  - CART

Neural Nets

- Very accurate black box
- Hard to interpret
- Weight at one "neuron" is sum of input weights run through "threshhold" (logistic) function)
- Easy to get nonlinearities, interactions



Neural nets in R

- library(nnet)
- m=nnet(formula, size=n)

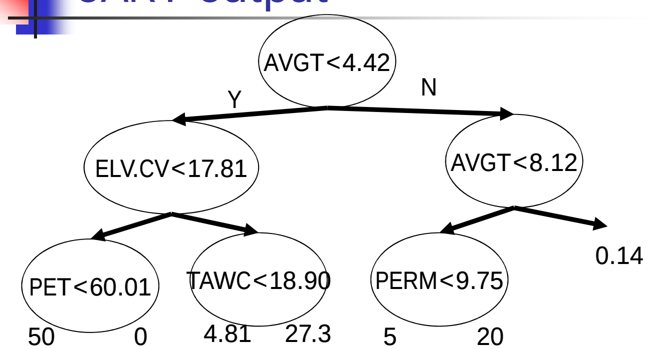
Types

- Error still
  - Robust (don't overweigh outliers)
  - Quantile – don't focus on central tendencies
- Nonlinear hypothesis
  - Nonlinear
  - Piecewise
- Machine learning
  - Smoothing
  - GAM
  - MARS
  - Neural Net
  - CART

CART

- Classification and regression tree
- In my opinion the best of the techniques
- Handles all sorts of nonlinearities/interactions
- Easily interpretible

CART output



Iverson et al *Ecological Monographs* 1998 – N of *Populus tremulodies*

CART Building algorithm

- Need optimization criterion
  - Depends on if dependent variable is metric or categorical
- In principle could optimize tree globally
  - In practice too computationally intensive

1. For each independent variable
   Select optimal split
   Calculate improvement of split
2. Select best split
3. Repeat for subgroup assigned to left
4. Repeat for subgroup assigned to right

CART Optimization criteria

- For metric dependent variable
  - Goal is reduction of variance
    - $Var_{group}=\Sigma\ (y_i-\mu_y)^2$
    - $Var_{split}= \Sigma_{left}\ (y_i-\mu_y)^2 + \Sigma_{right}\ (y_i-\mu_y)^2$
  - Optimization method depends on independent variable
    - Metric – order and try split between each set
      - 3 4 5.2 9.3 11.8 14.6
    - Categorical – try all permuations
      - A vs B,C A,B vs C, A,C vs B

Worked example

- Y= 1 2 3 4 6 7 8
- X_i=4 3 2 5 6 9 9
- Splits on X_i <3, <4, <5, <6, <9
- Original Y
  - Mean=4.23, Var=$(1-4.23)^2+\ldots$=41.71
- Split X_i<6
  - Y1=(1,2,3,4), Y2=(6,7,8)
  - Mean1=2.5, mean2=7
  - Var1=5, var2=2
  - Varsplit = 7
- Split X_i<5
  - Y1=(1,2,3), Y2=(4,6,7,8)
  - Varsplit=10.75

Cart optimzation criteria

- Instead of value at each leaf, have accuracy
  - E.g. 73/80=91.25% are absence
- Multiple choices if dependent is categorical
  - Variance (binomial np(1-p)
  - Gini: impurity
  - Twoing
  - Stratification (maximize accuracy of one node)

CART Pruning

- Will produce extremely large trees
- Need to “prune”
- Simplistic:
  - Maximum depth
  - Maximum # nodes (may be unbalanced)
  - Stop accuracy (% correct>x, or CV<y)
- Cross-validation

Summary of CART

- Specify dependent and independent variables
- Produces tree
- Prune simplistically or optimally
- Pros
  - Handles some nonlinearities, all interactions
  - Handles mixed types of data
  - Very easy to interpret
  - Handles missing data
  - Robust to outliers
  - Computationally manageable
- Cons
  - Not best with certain nonlinearities
  - Unstable (small changes in data make big changes in tree)
  - Less accurate than NN, MARS

CART in R

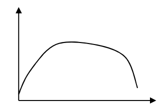
```
library(rpart)
m=rpart(formula,…)

data(iris)
mtree=rpart(Species~.,data=iris)
print(mtree)
print(mtree,cp=0.4)
plot(mtree,uniform=T,branch=0)
text(mtree,digits=3)
printcp(mtree)
plotcp(mtree)
mtree2=prune(mtree,cp=0.4)
```

Advanced techniques

- Thresholding
- ROC
- Bagging

Thresholding

- Output of dependent variable is often always metric (MARS, NN, GAM), even if want to study binary
- Use thresholding to convert to binary – i.e. y'=1 if y>x
- Optimal thresholding:
  - Plot % correctly classified vs. threshold



ROC

- Receiver operating characteristic
- False positives vs. false negatives not always equally important

ROC terms

true positive (TP)
    eqv. with hit
true negative (TN)
    eqv. with correct rejection
false positive (FP)
    eqv. with false alarm, Type I error
false negative (FN)
    eqv. with miss, Type II error

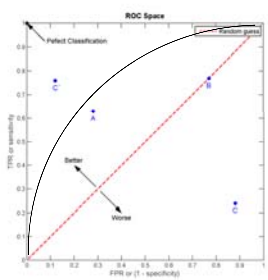
Source: Fawcett (2004).

sensitivity or true positive rate (TPR)
    eqv. with hit rate, recall
    $TPR = TP / P = TP / (TP + FN)$
false positive rate (FPR)
    eqv. with false alarm rate, fall-out
    $FPR = FP / N = FP / (FP + TN)$
accuracy (ACC)
    $ACC = (TP + TN) / (P + N)$
specificity (SPC) or True Negative Rate
    $SPC = TN / N = TN / (FP + TN) = 1 - FPR$
positive predictive value (PPV)
    eqv. with precision
    $PPV = TP / (TP + FP)$
negative predictive value (NPV)
    $NPV = TN / (TN + FN)$
false discovery rate (FDR)
    $FDR = FP / (FP + TP)$
Matthews correlation coefficient (MCC)

$$MCC = (TP\,TN - FP\,FN)/\sqrt{P N P' N'}$$

ROC

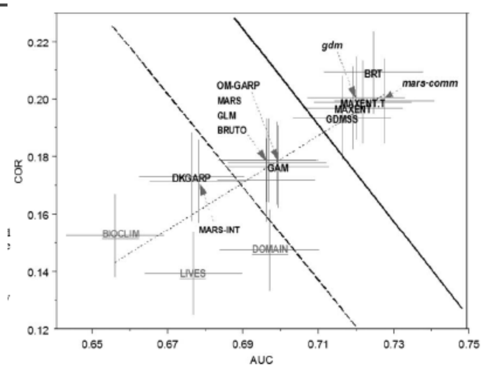
- Plot Sensitivity (true positive|positive) vs. 1-specificity (true negative|negative)=probability false alarm
- Tresholding (red-line = tune model to accept/reject)
- Idealized is right angles
- Area under curve (AUC) is measure of quality
  - 0.5=random
  - 1=perfect



Bagging

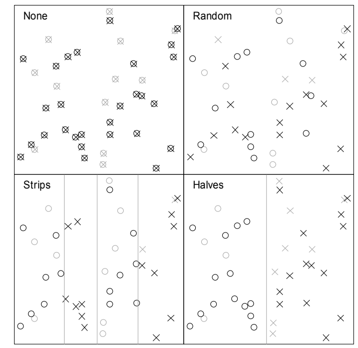
- Bagging
  - Build many trees in bootstrap fashion (sequential subsetting of data)
  - Prediction averaged across trees
- Boosting
  - A more sophisticated version
  - Can profoundly increase predictibility in cases where simple methods fail

Which technique is best

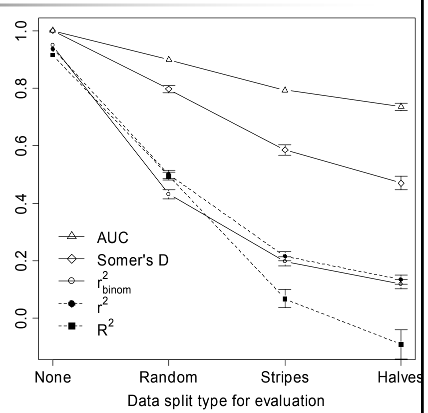


- Elith et al 2006

Spatial autocorrelation and non-independence



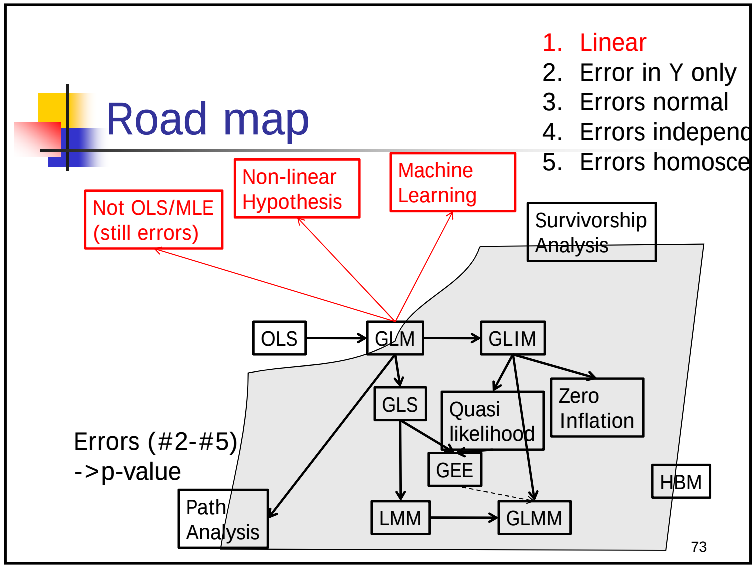
Evaluating predictive success without space



Bahn & McGill

Summary

- Predictive statements are vital in ecology
- Regression, GLIM, likelihood model selection are best of both worlds if have a priori expectation and understanding of errors
- Statistical learning is great for:
  - No a priori expectation (new field/data)
  - Quality of prediction is critical (applied problems)
  - Complexity precludes ever developing a priori expectation
- Statistical learning tools include:
  - Smoothing –local regression – good but only up to 2 dimensions
  - GAM – easy to understood, only additive effects
  - MARS – very accurate, difficult to understand
  - NN – very accurate, impossible to understand
  - CART – accurate, easy to understand


Road map
1. Linear
2. Error in Y only
3. Errors normal
4. Errors independ
5. Errors homosce
Not OLS/MLE (still errors)
Non-linear Hypothesis
Machine Learning
Survivorship Analysis
OLS
GLM
GLIM
GLS
Quasi likelihood
Zero Inflation
GEE
LMM
GLMM
HBM
Path Analysis
Errors (#2-#5) ->p-value
73