Neural Networks & Boosting

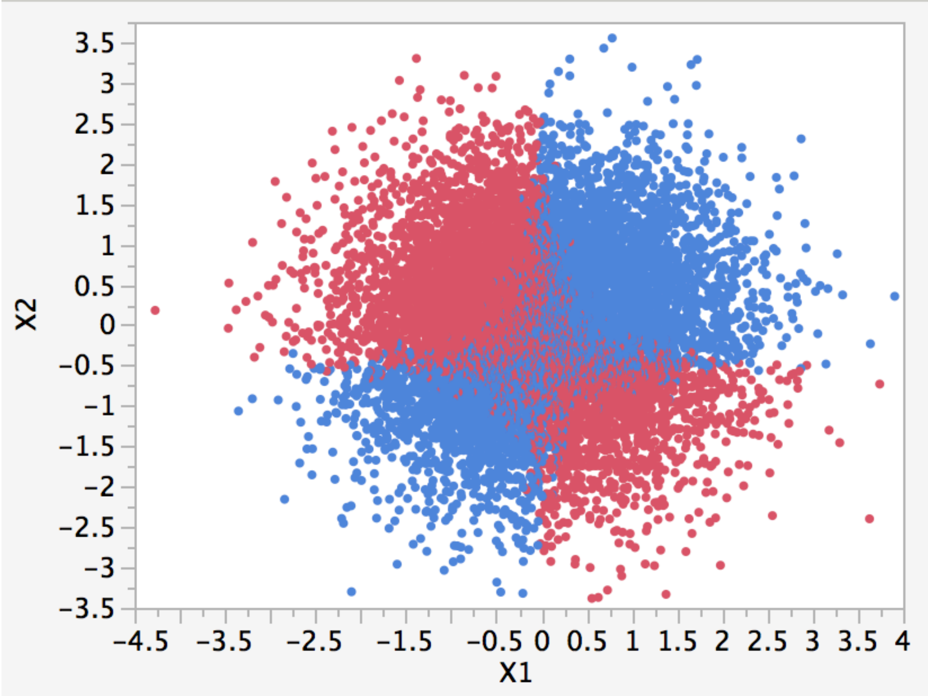
Bob Stine
Dept of Statistics, Wharton School
University of Pennsylvania

Questions

- How is logistic regression different from OLS?
  - Logistic mean function for probabilities
  - Larger weight to cases with $\hat{y} \approx 0$ or $\hat{y} \approx 1$.
  - Multiplicative structure rather than linear
- What's a good reference?
  - Try Agresti
- Other questions?

Simulated Example

- Simulated data with nonlinear structure
  - Two features X_1 and X_2 are iid N(0,1)
  - Mean $\mu = X_1 + 2\, X_1\, X_2 = X_1\,(1 + 2\, X_2)$
  - Add noise, set to 1 if positive and 0 if negative

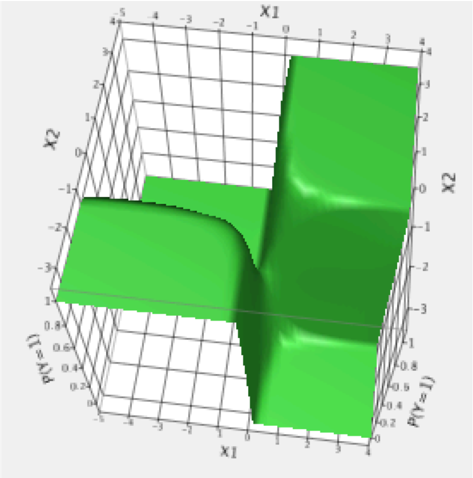


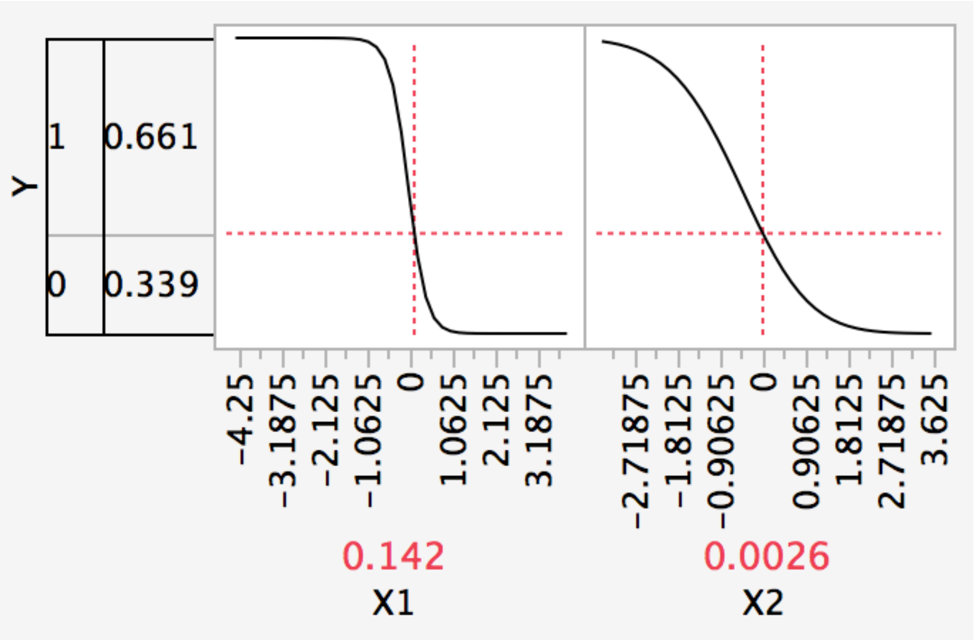
- Linear logistic
  - AUC=0.68 with R^2=0.07
  - Improve it?

Zeng (1999) SMR

Fit of Neural Network

- "Out of the box" default settings
  - Same response, same features X_1 and X_2
  - Accept other settings from JMP
- Network model captures structure
  - AUC=0.96
  - Visuals

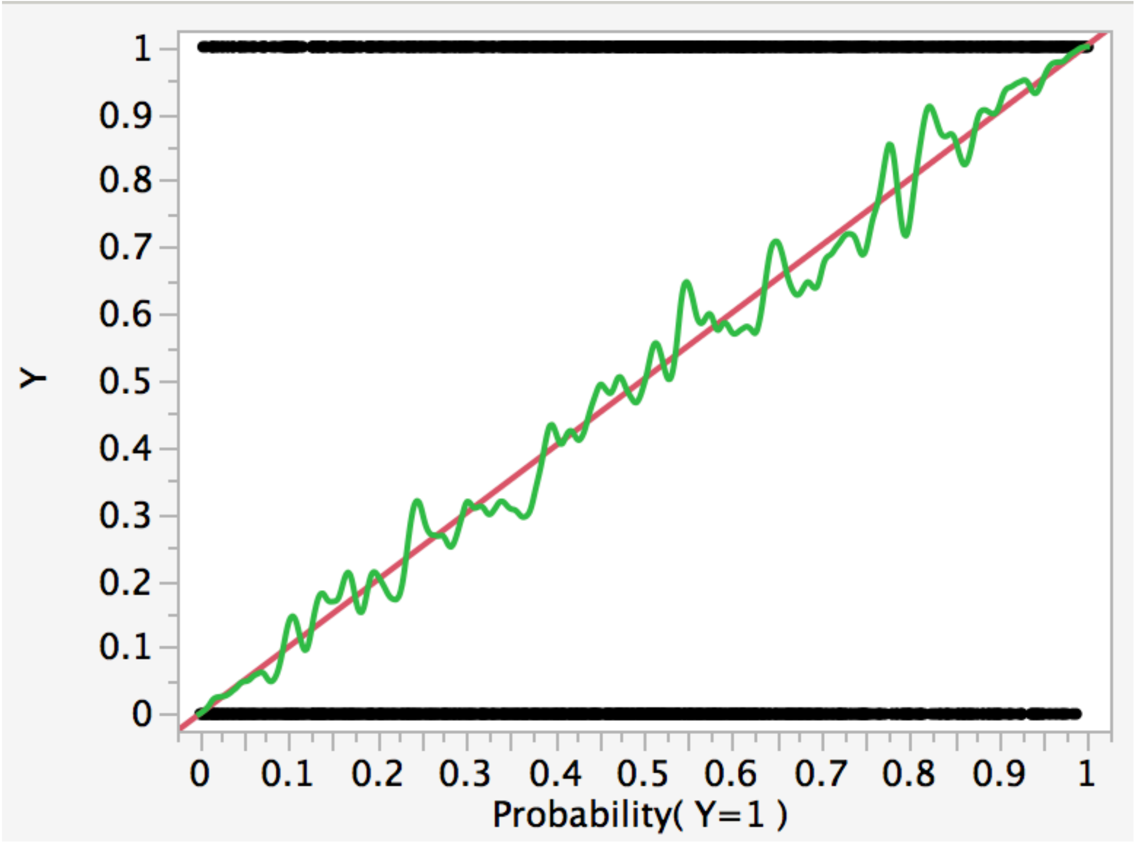




Wharton
Department of Statistics

Calibration Plot

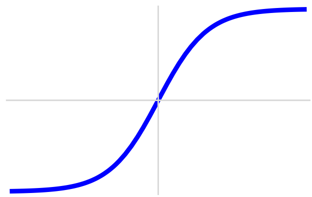
- Treat response as numerical
  - Same column of 0/1, treated differently
- Well calibrated...



Neural Network

- Combines logistic regression models
  - Latent variables in one or more "hidden layers"

$$\hat{Y} = G(\, G_1(Z_1) + \ldots + G_m(Z_m)\,)$$



  - G_j is logistic or similar sigmoidal function
  - Many parameters

    Possible over-fitting

    Gradient ascent from random starting value

    Optimization is not convex, so may not find best

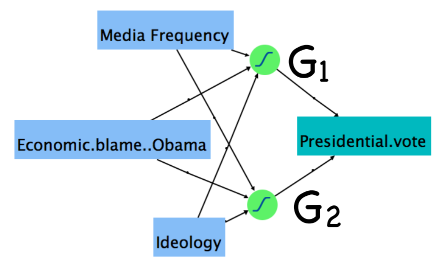
- Context
  - Neuron analogy

    Nonlinear response to stimulus, activation function

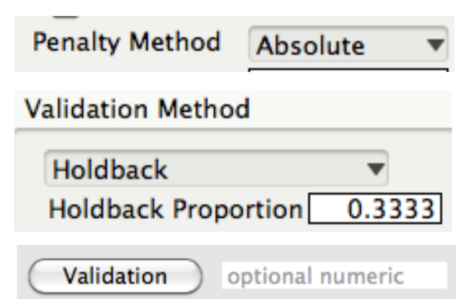
  - Predated by projection pursuit regression
  - Best for low noise, complex mean function

Basic Structure

- Each node/neuron represents response to mixture of all input features
  - $G_1(X_1,X_2,X_3) = G_1(b_{10}+b_{11}X_1+b_{12}X_2+b_{13}X_3)$
    $G_2(X_1,X_2,X_3) = G_2(b_{20}+b_{21}X_1+b_{22}X_2+b_{23}X_3)$
    Combine "hidden layer" in final node
    $$\hat{Y} = G(G_1, G_2)$$
  - Options include choice of functionG, number of nodes and number of layers

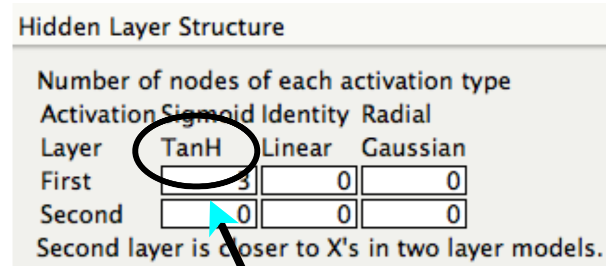


- Avoiding over-fitting
  - Regularization (lasso-style penalty)
  - Best fitting model in "validation sample"
  - Three-way cross-validation in JMP Pro



Building Network

- Complexity choice
  - Number of "hidden" nodes
  - Latent variables

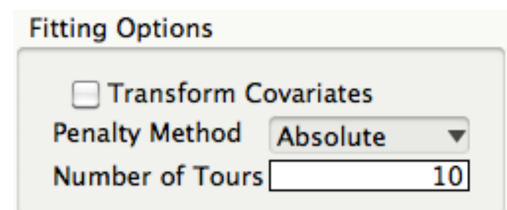


another name for logistic

- Three hidden nodes by default
  - TanH = rescaled, centered logistic
  - Use hidden nodes as needed
- Fitting options
  - Transform covariates mitigates skewness in predictors
  - Optional penalty as in Lasso (L_1)
  - Tours: # random starting points when estimating

    Picks the best of these in the "validation" sample

    Easy for the tours to get lost in the jungle of the parameter space!

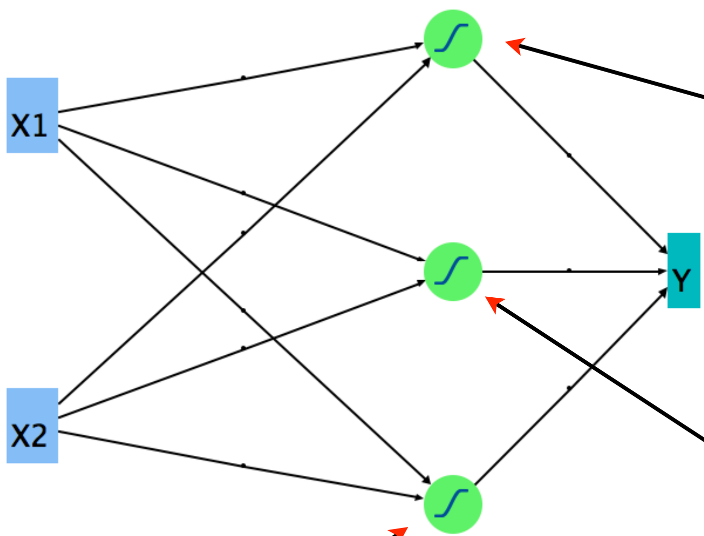


Cross-Validation in NN

- Cross-validation requires 3-way split
  - Training: Estimate parameters
  - Tuning: Evaluate goodness of fit, tune parms
  - Testing: How well does chosen network do?
- Comparison to validation in regression
  - CV needs 2-way split if use a selection criterion
  - Lasso methods use 3-way split
  - Selection criteria not well established for nets
- 3-way validation: train, tune, test
  - Ought to repeat splitting to reduce variation
  - Automated in some software

Simple Network

- Fit prior network using validation column
  - Two observed inputs, X_1 and X_2
  - One hidden layer, three nodes, lasso-style penalty
  - Default hidden nodes are logistic curves (tanh)



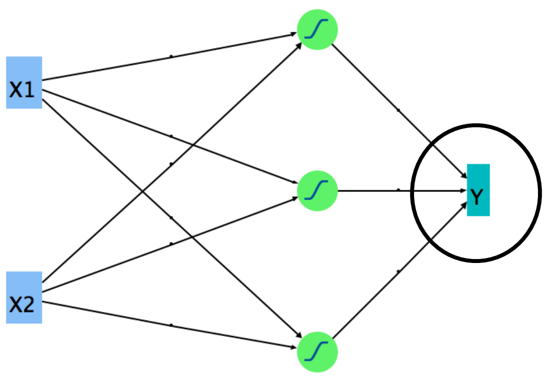
$$\mathrm{TanH}\left(0.5 * \left(-2.6707385457385 + 0.0999250017437 * X1 + 0.14676427865311 * X2\right)\right)$$

$$\mathrm{TanH}\left(0.5 * \left(-1.4775238392815 + 0.08526338256851 * X1 + -0.0196465760076 * X2\right)\right)$$

$$\mathrm{TanH}\left(0.5 * \left(-2.8086985363148 + -0.0281985679725 * X1 + 0.11456875377288 * X2\right)\right)$$

Basic Neural Network

- Response is categorical
  - Combine output of top layer in logistic function
  - Estimated probability



$$\frac{1}{1 + \mathrm{Exp}\left[4157.74655522397 + -3850.9802325838 * H1\_1 + 2439.99613197002 * H1\_2 + 6743.98810187317 * H1\_3\right]}$$

- If response is numerical
  - Handle final score differently
  - Linear function $b_0 + b_1 H_1 + b_2 H_2 + b_3 H_3$

Neural Net Results

- Fit default 3 node network
  - One layer, 3 nodes
  - 2000 for training, 1000 for tuning, 7000 for test
  - Absolute penalty, 5 tours
- Results

Training

Y

Measures	Value
Generalized RSquare	0.7820139
Entropy RSquare	0.6370428
RMSE	0.2863217
Mean Abs Dev	0.1654289
Misclassification Rate	0.12
−LogLikelihood	503.15243
Sum Freq	2000

Confusion Matrix

Actual	Predicted	
Y	0	1
0	861	145
1	95	899

Confusion Rates

Actual	Predicted	
Y	0	1
0	0.85586	0.14414
1	0.09557	0.90443

Validation

Y

Measures	Value
Generalized RSquare	0.7835747
Entropy RSquare	0.6391902
RMSE	0.2838839
Mean Abs Dev	0.1640437
Misclassification Rate	0.121
−LogLikelihood	249.93194
Sum Freq	1000

Confusion Matrix

Actual	Predicted	
Y	0	1
0	413	72
1	49	466

Confusion Rates

Actual	Predicted	
Y	0	1
0	0.85155	0.14845
1	0.09515	0.90485

Test

Y

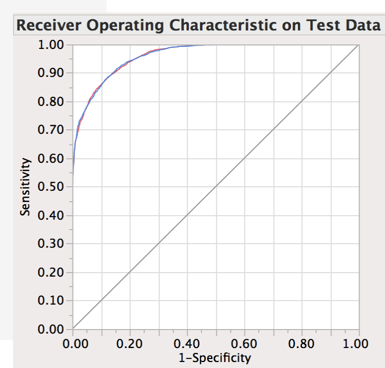
Measures	Value
Generalized RSquare	0.7874762
Entropy RSquare	0.6442211
RMSE	0.2836858
Mean Abs Dev	0.165101
Misclassification Rate	0.121
−LogLikelihood	1726.2497
Sum Freq	7000

Confusion Matrix

Actual	Predicted	
Y	0	1
0	2988	513
1	334	3165

Confusion Rates

Actual	Predicted	
Y	0	1
0	0.85347	0.14653
1	0.09546	0.90454



Wharton
Department of Statistics

As good as it gets?

- Nice feature of simulated example
  - Know true probability that Y=1

$$P(Y=1) = 1 - \Phi(\text{-signal/noise sd})$$

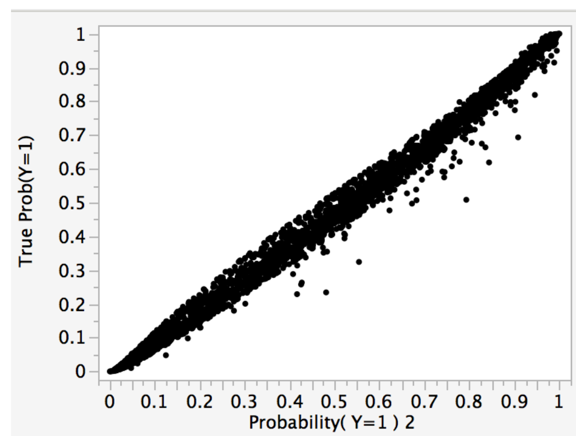
- Bayes classifier
  - Classify into group with largest probability
- Remove substantial random variation
  - How well do estimated probabilities approximate Pr(Y=1) rather than classify values?

Most Likely Y 2

Y	Count / Row %	0	1	
	0	2988	513	3501
		85.35	14.65	
	1	334	3165	3499
		9.55	90.45	
		3322	3678	7000

Bayes Most Likely Y

Y	Count / Row %	0	1	
	0	3091	410	3501
		88.29	11.71	
	1	419	3080	3499
		11.97	88.03	
		3510	3490	7000



Wharton
Department of Statistics

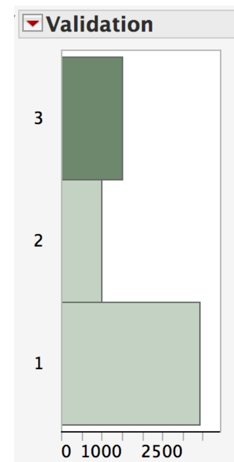
Attitudes to Gay Marriage

- Response: Favor or Oppose extremes
  - Harder modeling challenge
  - Variables are a variety of issue responses
- Start with logistic regression
  - Set baseline for neural models

    Which explanatory variables do you want to try?
  - Validation

    Exclude test data from modeling for logistic

Other examples in bibliography



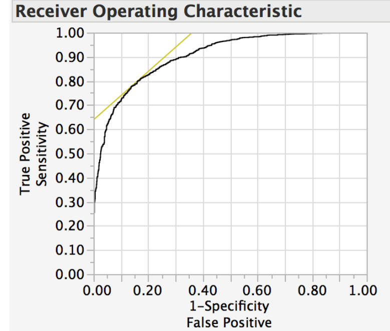
Party.identification
Ideology
Support.for.Obamacare
Abortion.attitude{Never p
Equality.Index
Legalization.of.marijuana
Women.in.the.home{Muc
Gender
Race{Black&Missing&Oth
Age{75+&Missing&65 -
Education
Church.attendance{Every
Church.attendance{Almo

Model	-LogLikelihood	DF	ChiSquare	Prob>ChiSq
Difference	762.1353	16	1524.271	<.0001*
Full	1009.4916			
Reduced	1771.6269			

RSquare (U)	0.4302
AICc	2053.22
BIC	2153
Observations (or Sum Wgts)	2653

Measure	Training	Definition
Entropy RSquare	0.4302	1-Loglike(model)/Loglike(0)
Generalized RSquare	0.5930	(1-(L(0)/L(model))^(2/n))/(1-L(0)^(2/n))
Mean -Log p	0.3805	$\sum$ -Log(ρ[j])/n
RMSE	0.3507	$\sqrt{\sum(y[j]-\rho[j])^2/n}$
Mean Abs Dev	0.2435	$\sum$ |y[j]-ρ[j]|/n
Misclassification Rate	0.1824	$\sum$ (ρ[j]≠ρMax)/n
N	2653	n

Receiver Operating Characteristic



Using Allow Gay Marriage?='Yes' to be the positive level

AUC
0.90344

Most Likely Allow Gay Marriage?

Allow Gay Marriage?	Count / Row %	No	Yes	
No	Count	344	143	487
	Row %	70.64	29.36	
Yes	Count	87	646	733
	Row %	11.87	88.13	
		431	789	1220

Attitudes to Gay Marriage

- Build several neural networks
  - Compare performace to logistic
  - Explore several choices for network

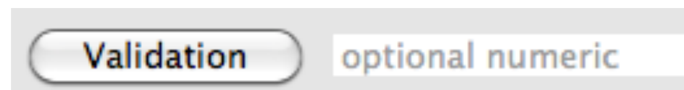
    Different numbers of hidden nodes

    One or two layers

    L_1 regularization (ie, lasso type shrinkage)

NN does not use sample weights

- Three-way cross-validation



  - Training, tuning (aka, 'validation'), testing
  - 20 "tours" to find best fit
- Software facilitates comparison
  - "Model launch" allows fitting different networks

Results of Several

- Three-way cross-validation reduces n

Which model do you prefer?

Model NTanH(3)

Training — Allow Gay Marriage?

Measures	Value
Generalized RSquare	0.6211914
Entropy RSquare	0.4581092
RMSE	0.3409437
Mean Abs Dev	0.2388433
Misclassification Rate	0.1711122
-LogLikelihood	720.08275
Sum Freq	1987

Validation — Allow Gay Marriage?

Measures	Value
Generalized RSquare	0.5976086
Entropy RSquare	0.4354169
RMSE	0.3499172
Mean Abs Dev	0.2410794
Misclassification Rate	0.1756757
-LogLikelihood	249.94907
Sum Freq	666

Test — Allow Gay Marriage?

Measures	Value
Generalized RSquare	0.5535245
Entropy RSquare	0.3913833
RMSE	0.364238
Mean Abs Dev	0.2503337
Misclassification Rate	0.1959016
-LogLikelihood	499.47171
Sum Freq	1220

Model NTanH(1)

Training — Allow Gay Marriage?

Measures	Value
Generalized RSquare	0.6052991
Entropy RSquare	0.4421099
RMSE	0.345097
Mean Abs Dev	0.2526547
Misclassification Rate	0.172622
-LogLikelihood	741.34322
Sum Freq	1987

Validation — Allow Gay Marriage?

Measures	Value
Generalized RSquare	0.5862856
Entropy RSquare	0.4243254
RMSE	0.3519547
Mean Abs Dev	0.2529952
Misclassification Rate	0.1771772
-LogLikelihood	254.85942
Sum Freq	666

Test — Allow Gay Marriage?

Measures	Value
Generalized RSquare	0.5563231
Entropy RSquare	0.3939925
RMSE	0.3634832
Mean Abs Dev	0.2612669
Misclassification Rate	0.1918033
-LogLikelihood	497.33043
Sum Freq	1220

Model NTanH(2)NTanH2(2)

Training — Allow Gay Marriage?

Measures	Value
Generalized RSquare	0.6158523
Entropy RSquare	0.4526959
RMSE	0.3413808
Mean Abs Dev	0.2464796
Misclassification Rate	0.1716155
-LogLikelihood	727.27619
Sum Freq	1987

Validation — Allow Gay Marriage?

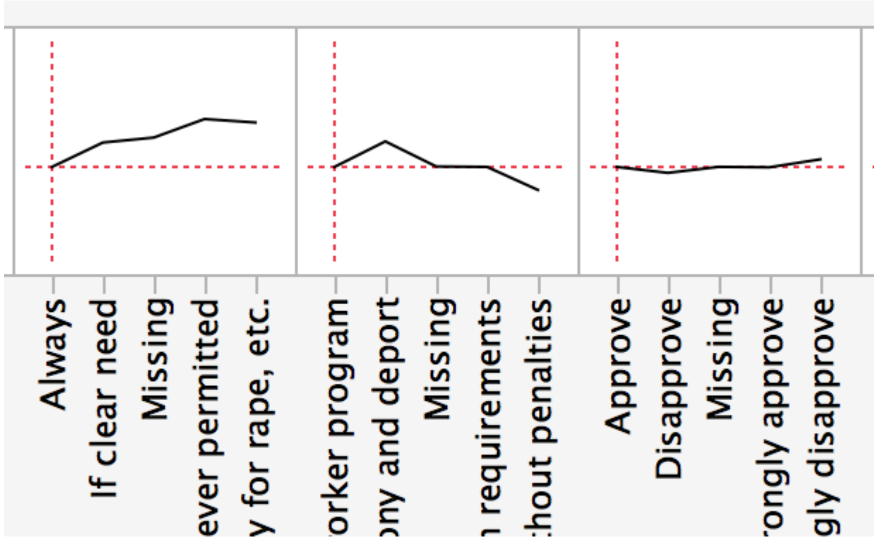
Measures	Value
Generalized RSquare	0.582498
Entropy RSquare	0.4206515
RMSE	0.3537524
Mean Abs Dev	0.2511623
Misclassification Rate	0.1846847
-LogLikelihood	256.48594
Sum Freq	666

Test — Allow Gay Marriage?

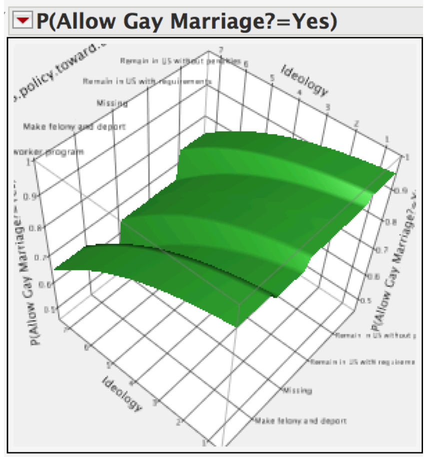
Measures	Value
Generalized RSquare	0.5577458
Entropy RSquare	0.3953225
RMSE	0.3628187
Mean Abs Dev	0.2572601
Misclassification Rate	0.1901639
-LogLikelihood	496.23894
Sum Freq	1220

Exploring NN Model

- Model with 2 nodes in 2 layers
- Profiling



- Surface profiles



Comparison to LR

- Logistic regression
  - Use Training and Tuning samples for fit,

Most Likely Allow Gay Marriage?

Allow Gay Marriage?	Count Row %	No	Yes	
	No	344 70.64	143 29.36	487
	Yes	87 11.87	646 88.13	733
		431	789	1220

- Neural network using 2 hidden layers

Confusion Matrix

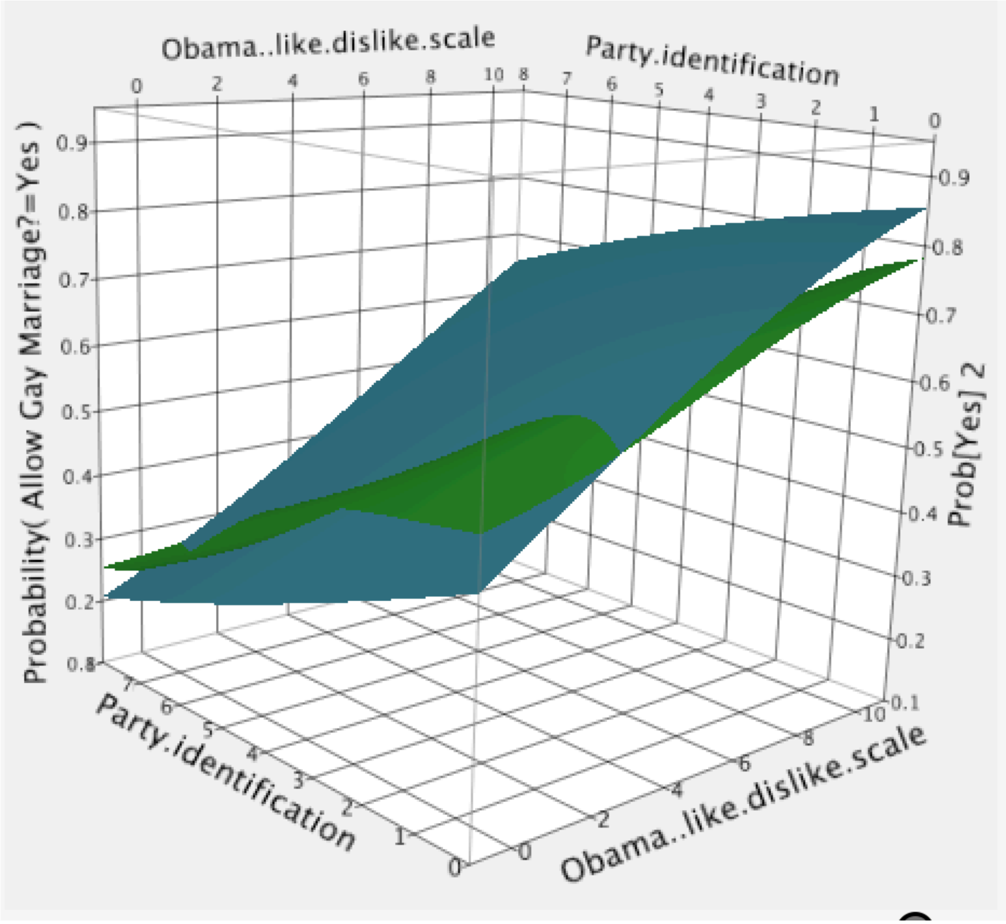
Actual **Allow Gay Marriage?**	Predicted **No**	**Yes**
No	314	173
Yes	78	655

Randomness in optimization implies you don't get the same results each time

Wharton
Department of Statistics

Visual Comparison

- Superimposed prediction surfaces
  - Save formulas, use Graph>Surface



NN typically has the more 'wavy' fit

Boosting

- General method for improving predictive model
  - Build additive sequence of predictive models (ensemble)

    Final prediction is accumulated over many models.
  - Start with initial predictive model
  - Compute residuals from current fit
  - Build model for residuals
  - Repeat

Original method called Adaboost

- Implication: Use simple model at each step
  - Weak learner: single-layer, 1 or 2 nodes
  - Next response = (current response) - (learning rate) x fit

    0.1 or smaller
- Weaknesses
  - Loss of ‘interpretability’, at what gain?

Boosted Net

Model Launch

Hidden Layer Structure

Number of nodes of each activation type

Activation	Sigmoid	Identity	Radial
Layer	TanH	Linear	Gaussian
First	1	0	0
Second	0	0	0

Simple

Second layer is closer to X's in two layer models.

Boosting

Fit an additive sequence of models scaled by the learning rate.

Number of Models: 100

Learning Rate: 0.05

Not too many

Fitting Options

☐ Transform Covariates

Penalty Method: Absolute

Number of Tours: 5

Takes a while!

Model NTanH(1)NBoost(52)

Training — Allow Gay Marriage?

Measures	Value
Generalized RSquare	0.6259873
Entropy RSquare	0.4630055
RMSE	0.3354314
Mean Abs Dev	0.2438098
Misclassification Rate	0.1560141
-LogLikelihood	713.57648
Sum Freq	1987

Validation — Allow Gay Marriage?

Measures	Value
Generalized RSquare	0.6174001
Entropy RSquare	0.4552056
RMSE	0.3386178
Mean Abs Dev	0.2427621
Misclassification Rate	0.1681682
-LogLikelihood	241.18833
Sum Freq	666

Worthwhile?

Test — Allow Gay Marriage?

Measures	Value
Generalized RSquare	0.5494252
Entropy RSquare	0.3875778
RMSE	0.3651859
Mean Abs Dev	0.2621704
Misclassification Rate	0.1901639
-LogLikelihood	502.59472
Sum Freq	1220

Actual	Predicted	
Allow Gay Marriage?	No	Yes
No	341	146
Yes	86	647

Discussion

- Resurgence of interest in neural networks
  - Had fallen out of favor
    - Too hard to fit, too complex, huge collinearities
  - Deep networks have produced surprising results: 15% improvements over standard
  - Deep network: many layers, 1000's of nodes
    - Context: text mining
    - Papers by Hinton and colleagues on 'deep learning'
- Interpretation requires graphics
  - Too many parameters, nonlinear
    - Would be handy to have way to "sort" features in level of importance
  - Comparison of profiles, surfaces
    - But can be overwhelmed by larger numbers of predictors

Take-Aways

- Neural network
  - Combines several logistic regressions (latent vars)
    Allow either numerical or categorical response
  - Complex fitting process with many parameters
    Not as fast as trees that we'll see next.
  - Requires 3-way cross-validation
    Tuning sample is necessary; unfortunately cannot easily automate repeats
- Boosting
  - Refit simple model to residuals
  - Combines sequence of simple models to capture structure rather than rely on one complex model
- Model visualization is essential
- Small n: Hard to beat a simple, accurate model

Some questions to ponder...

- How are neural networks and logistic regression related?
- How can you use repeated CV to convey the stability of predictions?
- To pick the best neural network, we used 3-way cross-validation. To get an unbiased estimate of how well this network fits, what do we need?
- How would boosting a linear regression work?

Next Time

- Classification trees
  - Partitioning data into homogeneous subsets

- Thursday is Newberry Lab session