Intelligible Models for Classification and Regression

Yin Lou[1] Rich Caruana[2] Johannes Gehrke[1]

Department of Computer Science[1]
Cornell University

Microsoft Research[2]
Microsoft Corporation

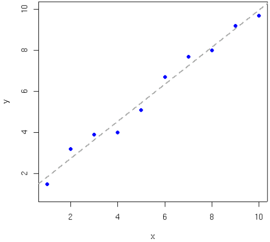
Aug. 13, 2012

Motivation

Simple Model

- Linear regression, logistic regression
- Regression: $y = \beta_0 + \beta_1 x_1 + ... + \beta_n x_n$
- Classification: $logit(y) = \beta_0 + \beta_1 x_1 + ... + \beta_n x_n$

Linear Regression

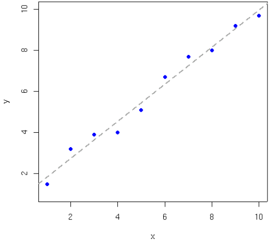


Motivation

Simple Model

- Linear regression, logistic regression
- Regression: $y = \beta_0 + \beta_1 x_1 + ... + \beta_n x_n$
- Classification: $logit(y) = \beta_0 + \beta_1 x_1 + ... + \beta_n x_n$

Linear Regression



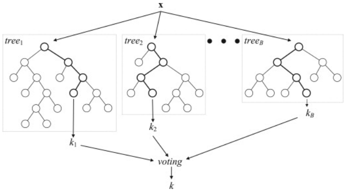
Intelligible but usually less accurate

Motivation

Complex Model

- Random forest, SVMs with RBF kernel, etc.
- $y = f(x_1, ..., x_n)$

Random Forest

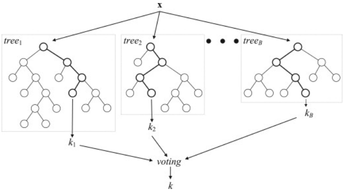


Motivation

Complex Model

- Random forest, SVMs with RBF kernel, etc.
- $y = f(x_1, ..., x_n)$

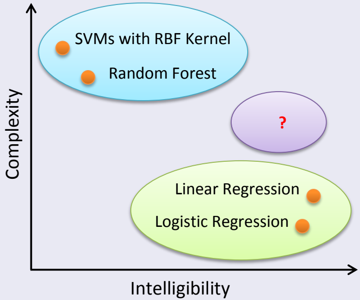
Random Forest



Unintelligible but usually more accurate

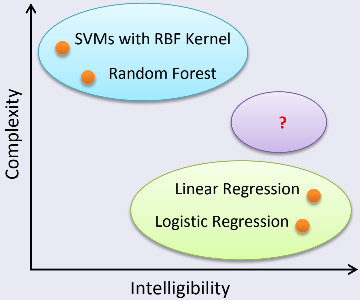
Motivation

The tradeoff



Motivation

The tradeoff



Intelligibility is important

- Medical applications
- Domains where we want scientific understanding
- Efficient model engineering
  - Impact of features in a ranker

Outline

Outline

Generalized Additive Models

- Developed by Hastie and Tibshirani
- Regression: $y = f_1(x_1) + ... + f_n(x_n)$
- Classification: $logit(y) = f_1(x_1) + ... + f_n(x_n)$
- Each feature is "shaped" by shape function f_i
- Intelligible and accurate

T. Hastie and R. Tibshirani.
Generalized additive models.
Chapman & Hall/CRC, 1990.

Example

$$y = x_1 + x_2^2 + \sqrt{x_3} + \log x_4 + e^{x_5} + 2\sin x_6 + \epsilon$$

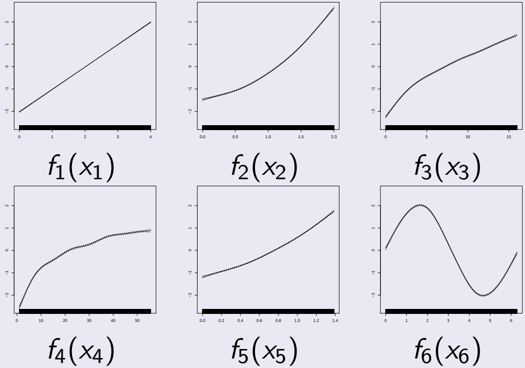


Figure: Shape Functions for Synthetic Dataset.

Model Space

Model	Form	Intelligibility	Accuracy
Linear Model	$y = \beta_0 + \beta_1 x_1 + ... + \beta_n x_n$	+++	+
Generalized Linear Model	$g(y) = \beta_0 + \beta_1 x_1 + ... + \beta_n x_n$	+++	+
Additive Model	$y = f_1(x_1) + ... + f_n(x_n)$	++	++
Generalized Additive Model	$g(y) = f_1(x_1) + ... + f_n(x_n)$	++	++
Full Complexity Model	$y = f(x_1, ..., x_n)$	+	+++

Table: From Linear to Additive Models.

Outline

Fitting GAMs

$$g(y) = f_1(x_1) + ... + f_n(x_n)$$

Shape Functions

- Splines (SP)
- Single Tree (TR)
- Bagged Trees (bagTR)
- Boosted Trees (bstTR)
- Boosted Bagged Trees (bbTR)

Fitting GAMs

$$g(y) = f_1(x_1) + ... + f_n(x_n)$$

Shape Functions

- Splines (SP)
- Single Tree (TR)
- Bagged Trees (bagTR)
- Boosted Trees (bstTR)
- Boosted Bagged Trees (bbTR)

Learning Methods

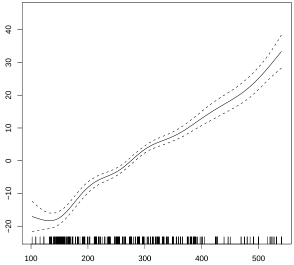
- Penalized Least Squares (P-LS/P-IRLS)
- Backfitting (BF)
- Gradient Boosting (BST)

Fitting GAMs

$$g(y) = f_1(x_1) + \ldots + f_n(x_n)$$

Shape Function: Splines (SP)

- $f_i(x_i) = \sum_{k=1}^{d} \beta_k b_k(x_i)$

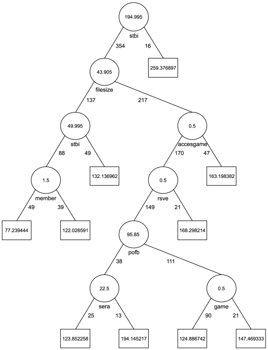


Fitting GAMs

$$g(y) = f_1(x_1) + \ldots + f_n(x_n)$$

Shape Function: Single Tree (TR)

- $f_i(x_i) = RegressionTree(x_i, response)$



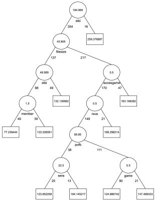
Fitting GAMs

$g(y) = f_1(x_1) + ... + f_n(x_n)$

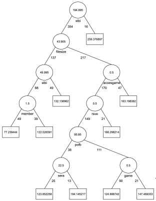
Shape Function: Bagged Trees (bagTR)

- $f_i(x_i) = \frac{1}{B} \sum_{j=1}^{B} RegressionTree(x_i, bootstrap\ sample\ j)$

$\frac{1}{B}$ (



$+ \quad ... \quad +$



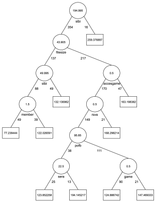
)

Fitting GAMs

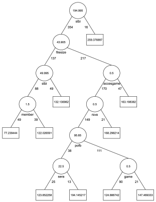
$$g(y) = f_1(x_1) + \ldots + f_n(x_n)$$

Shape Function: Boosted Trees (bstTR)

- $f_i(x_i) = \sum_{j=1}^{B} RegressionTree(x_i, residual_j)$



$+ \quad \ldots \quad +$



Fitting GAMs

$g(y) = f_1(x_1) + ... + f_n(x_n)$

Shape Function: Boosted Bagged Trees (bbTR)

- $f_i(x_i) = \sum_{j=1}^{B} BaggedRegressionTree(x_i, residual_j)$

Fitting GAMs

$$g(y) = f_1(x_1) + ... + f_n(x_n)$$

Learning Method: Penalized Least Squares (P-LS/P-IRLS)

- Works only on Splines ($f_i(x_i) = \sum_{k=1}^{d} \beta_k b_k(x_i)$)
- Converts the optimization problem to fitting linear regression/logistic regression with different basis

S. Wood.
Generalized additive models: an introduction with R.
CRC Press, 2006.

Fitting GAMs

$$g(y) = f_1(x_1) + ... + f_n(x_n)$$

Learning Method: Backfitting (BF)

$f_j \leftarrow 0$
for $m = 1$ to M **do**
  for $j = 1$ to n **do**
    $\mathcal{R} \leftarrow \{x_{ij}, y_i - \sum_{k \neq j} f_k\}_1^N$
    Learn shaping function $S : x_j \rightarrow y$ using $\mathcal{R}$ as training dataset
    $f_j \leftarrow S$
  end for
end for

T. Hastie and R. Tibshirani.
Generalized additive models.
Chapman & Hall/CRC, 1990.

Fitting GAMs

$$g(y) = f_1(x_1) + ... + f_n(x_n)$$

Learning Method: Gradient Boosting (BST)

$f_j \leftarrow 0$
for $m = 1$ to M **do**
  for $j = 1$ to n **do**
    $\mathcal{R} \leftarrow \{x_{ij}, y_i - \sum_k f_k\}_1^N$
    Learn shaping function $S : x_j \rightarrow y$ using $\mathcal{R}$ as training dataset
    $f_j \leftarrow f_j + S$
  end for
end for

J. Friedman.
Greedy function approximation: a gradient boosting machine.
Annals of Statistics, 29:1189–1232, 2001.

Contributions

- First large-scale study that uses trees as shape function for GAMs
- Novel methods for using trees as shape functions
- Largest empirical study of fitting GAMs

Outline

Datasets

	Dataset	Size	Attributes	%Pos
Regression	Concrete	1030	9	-
	Wine	4898	12	-
	Delta	7192	6	-
	CompAct	8192	22	-
	Music	50000	90	-
	Synthetic	10000	6	-
Classification	Spambase	4601	58	39.40
	Insurance	9823	86	5.97
	Magic	19020	11	64.84
	Letter	20000	17	49.70
	Adult	46033	9/43	16.62
	Physics	50000	79	49.72

Methods

Shape Function	Least Squares	Gradient Boosting	Backfitting
Splines	P-LS/P-IRLS	BST-SP	BF-SP
Single Tree	N/A	BST-TRx	BF-TR
Bagged Trees	N/A	BST-bagTRx	BF-bagTR
Boosted Trees	N/A	BST-TRx	BF-bstTRx
Boosted Bagged Trees	N/A	BST-bagTRx	BF-bbTRx

Table: Notation for learning methods and shape functions.

- 9 different methods
- 5-fold cross validation for each method

Results

Model	Regression	Classification	Mean
Linear/Logistic			
P-LS/P-IRLS			
BST-SP			
BF-SP			
BST-bagTR2			
BST-bagTR3			
BST-bagTR4			
BST-bagTRX			
Random Forest			

Results

Model	Regression	Classification	Mean
Linear/Logistic	1.68	1.22	1.45
P-LS/P-IRLS			
BST-SP			
BF-SP			
BST-bagTR2			
BST-bagTR3			
BST-bagTR4			
BST-bagTRX			
Random Forest			

Results

Model	Regression	Classification	Mean
Linear/Logistic	1.68	1.22	1.45
P-LS/P-IRLS			
BST-SP			
BF-SP			
BST-bagTR2			
BST-bagTR3			
BST-bagTR4			
BST-bagTRX			
Random Forest	0.88	0.80	0.84

Results

Model	Regression	Classification	Mean
Linear/Logistic	1.68	1.22	1.45
P-LS/P-IRLS	1.00	1.00	1.00
BST-SP	1.04	1.00	1.02
BF-SP	1.00	1.00	1.00
BST-bagTR2			
BST-bagTR3			
BST-bagTR4			
BST-bagTRX			
Random Forest	0.88	0.80	0.84

Results

Model	Regression	Classification	Mean
Linear/Logistic	1.68	1.22	1.45
P-LS/P-IRLS	1.00	1.00	1.00
BST-SP	1.04	1.00	1.02
BF-SP	1.00	1.00	1.00
BST-bagTR2	0.96	0.96	0.96
BST-bagTR3	0.97	0.95	0.96
BST-bagTR4	0.99	0.95	0.97
BST-bagTRX	0.95	0.94	0.95
Random Forest	0.88	0.80	0.84

Results

Model	Regression	Classification	Mean
Linear/Logistic	1.68	1.22	1.45
P-LS/P-IRLS	1.00	1.00	1.00
BST-SP	1.04	1.00	1.02
BF-SP	1.00	1.00	1.00
BST-bagTR2	0.96	0.96	0.96
BST-bagTR3	0.97	0.95	0.96
BST-bagTR4	0.99	0.95	0.97
BST-bagTRX	0.95	0.94	0.95
Random Forest	0.88	0.80	0.84

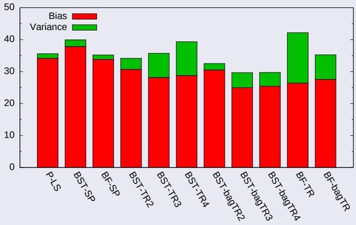
Observations

- Two accuracy gaps: shaping and interactions
- Tree-base shaping methods are more accurate than spline-based methods

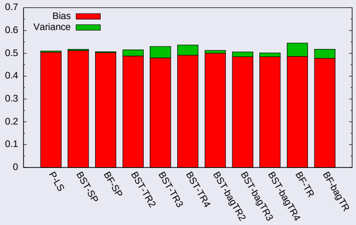
Outline

Bias Variance Decomposition

$$Expected\ Loss = (bias)^2 + variance + noise$$



(a) Concrete



(b) Wine

Figure: Bias-variance analysis.

Learned Shaped Function: Splines vs. Trees

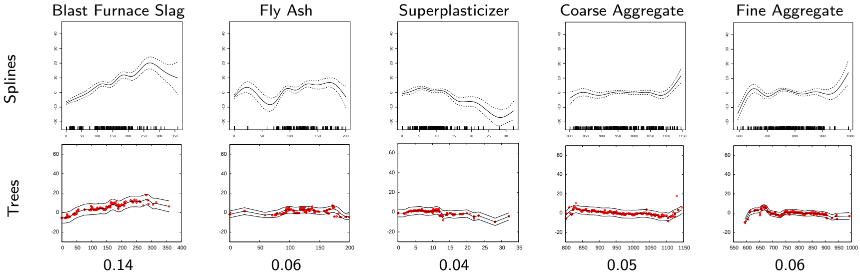


Figure: Shapes of features for the "Concrete" dataset produced by P-LS (top) and BST-bagTR3 (bottom).

Outline

Conclusion

- Generalized additive models are accurate and intelligible
- Tree has low bias but high variance
- Bagging reduces variance and makes tree-based method stand out
- Bagged shallow trees with gradient boosting are most accurate

Future Work

- Feature selection
- Scalability
- Statistical interaction detection

Thank You

Questions?