INT 303 BIG DATA ANALYTICS

- # Lecture11: Bagging

Jia WANG

Jia.wang02@xjtlu.edu.cn

Xi'an Jiaotong-Liverpool University

西交利物浦大學

GOAL OF SUPERVISED LEARNING?

- Minimize the probability of model prediction errors on ***future*** data

- **Goal:** learn predictor h(x)
  - High accuracy (low error)
  - Using training data $\{(x_1,y_1),\ldots,(x_n,y_n)\}$

OUTLINE

- **Bias/Variance Tradeoff**

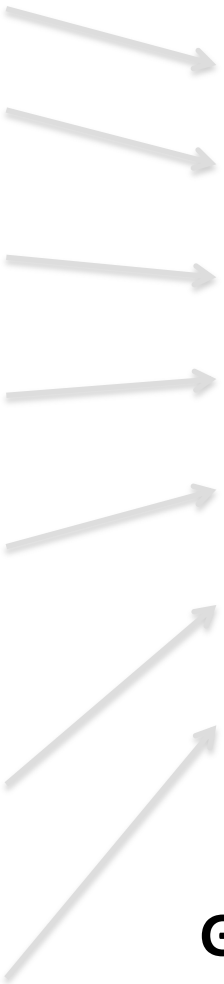
- Ensemble methods that minimize variance
  - Bagging
  - Random Forests

- Ensemble methods that minimize bias
  - Functional Gradient Descent
  - Boosting
  - Ensemble Selection

GENERALIZATION ERROR

- **"True" distribution:** P(x,y)
  - Unknown to us
- **Train:** h(x) = y
  - Using training data S = $\{(x_1,y_1),\ldots,(x_n,y_n)\}$
  - Sampled from P(x,y)
- **Generalization Error:**
  - $\mathcal{L}(h) = E_{(x,y)\sim P(x,y)}[\, f(h(x),y)\,]$
  - E.g., $f(a,b) = (a-b)^2$

Person	Age	Male?	Height > 55”
James	11	1	1
Jessica	14	0	1
Alice	14	0	1
Amy	12	0	1
Bob	10	1	1
Xavier	9	1	0
Cathy	9	0	1
Carol	13	0	1
Eugene	13	1	0
Rafael	12	1	1
Dave	8	1	0
Peter	9	1	0
Henry	13	1	0
Erin	11	0	0
Rose	7	0	0
Iain	8	1	1
Paulo	12	1	0
Margaret	10	0	1
Frank	9	1	1
Jill	13	0	0
Leon	10	1	0
Sarah	12	0	0
Gena	8	0	0
Patrick	5	1	1



Person	Age	Male?	Height > 55”
Alice	14	0	1
Bob	10	1	1
Carol	13	0	1
Dave	8	1	0
Erin	11	0	0
Frank	9	1	1
Gena	8	0	0

y **h(x)**

Generalization Error:

$$\mathcal{L}(h) = E_{(x,y)\sim P(x,y)}[\, f(h(x),y)\,]$$

BIAS/VARIANCE TRADEOFF

- Treat h(x|S) has a machine learning function
  - Depends on data x sampled from the training data S

- $\mathcal{L} = E_{(x,y)\sim P(x,y)}[\, f(h(x|S),y)\,]$
  - Expected generalization error over the randomness sampling of S

BIAS/VARIANCE TRADEOFF

- Squared loss: $f(a,b) = (a-b)^2$
- Consider one data point (x,y)
- Notation:
  - $Z = h(x|S) - y$
  - $\check{z} = E[Z]$
  - $Z - \check{z} = h(x|S) - E[h(x|S)]$

$$E_S[(Z-\check{z})^2] = E[Z^2 - 2Z\check{z} + \check{z}^2]$$
$$= E[Z^2] - 2E[Z]\check{z} + \check{z}^2$$
$$= E[Z^2] - \check{z}^2$$

Expected Error

$$E[f(h(x|S),y)] = E[Z^2]$$
$$= E[(Z-\check{z})^2] + \check{z}^2$$

Variance **Bias**

Bias/Variance for all (x,y) is expectation over P(x,y).

Can also incorporate measurement noise.

(Similar flavor of analysis for other loss functions.)

BIAS/VARIANCE TRADEOFF

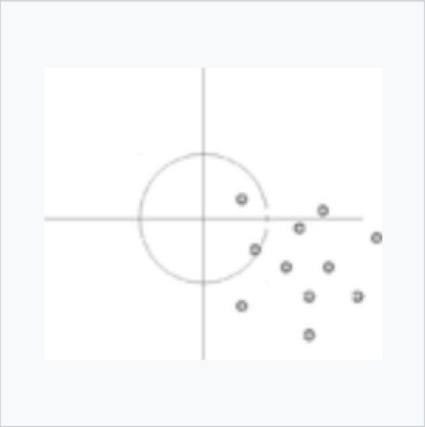
- The bias error is an error from erroneous assumptions in the learning algorithm.
  - High bias can cause an algorithm to miss the relevant relations between features and target outputs (underfitting).
- The variance is an error from sensitivity to small fluctuations in the training set.
  - High variance may result from an algorithm modeling the random noise in the training data (Overfitting).
- The **bias–variance** decomposition is a way of analyzing a learning algorithm's expected generalization error .

BIAS/VARIANCE TRADEOFF

bias low, variance low

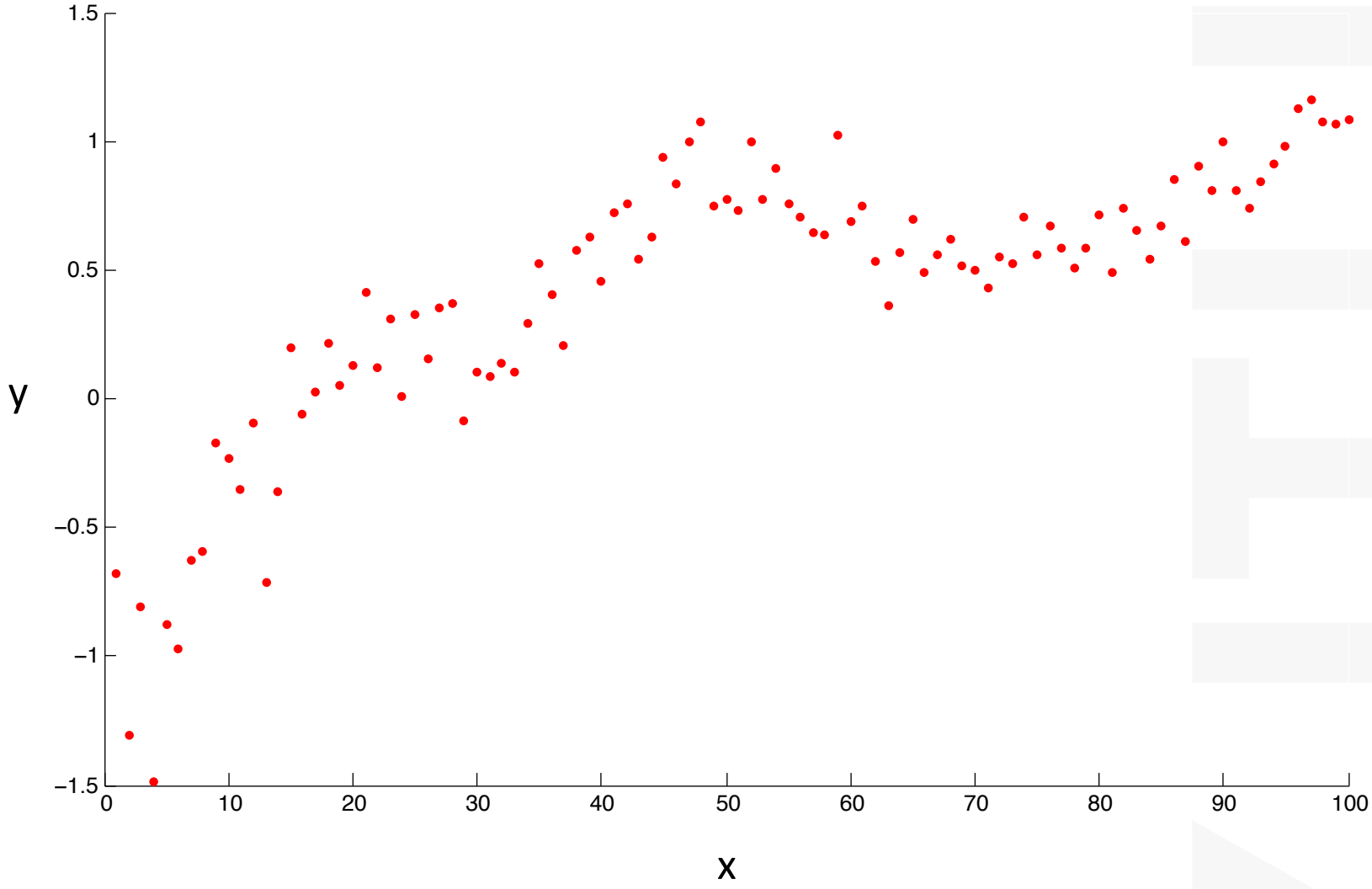
bias high,
variance low:

bias low,
variance high:

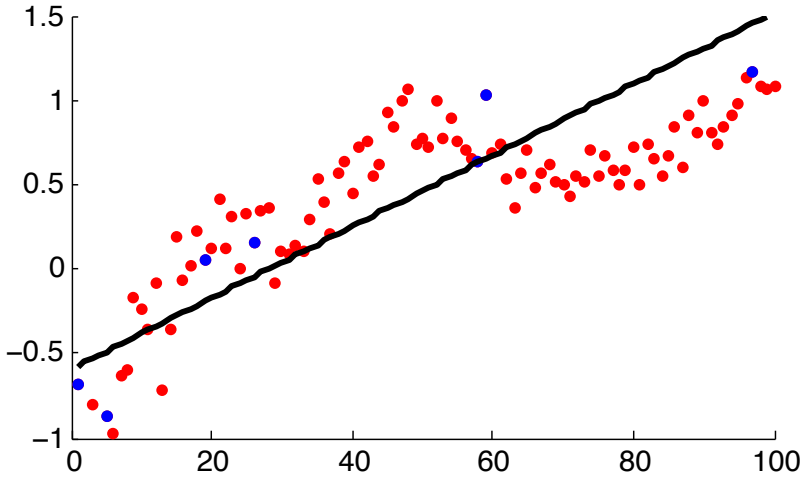


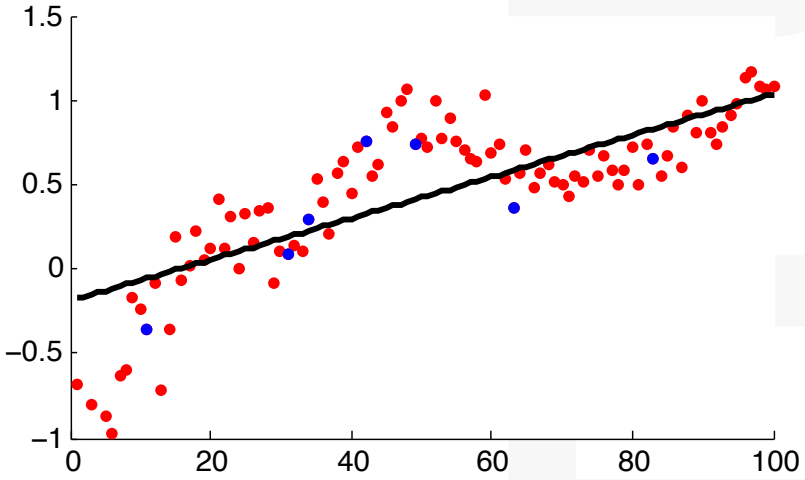
bias high,
variance high:

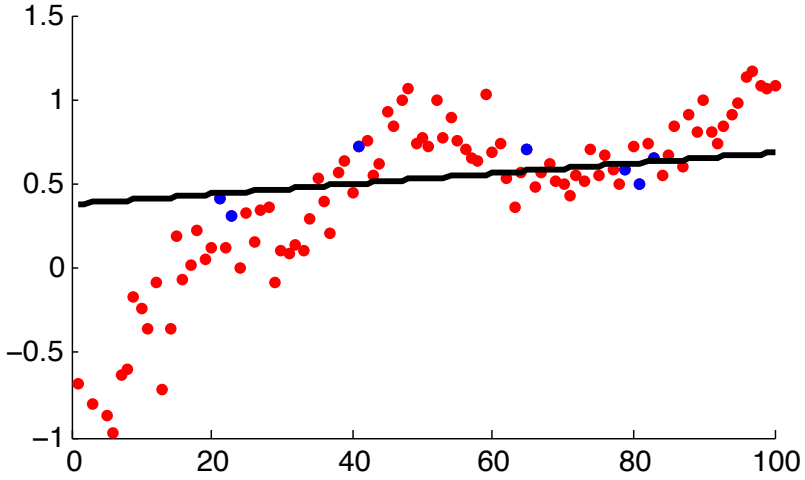
Example

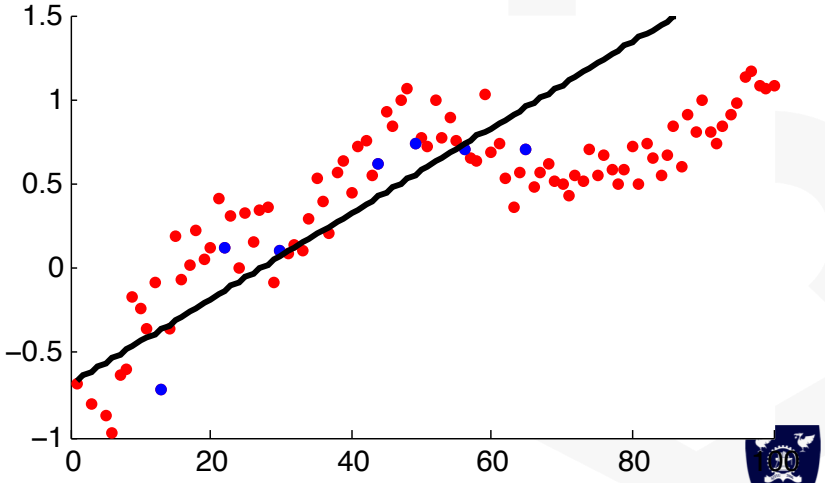


h(x|S)

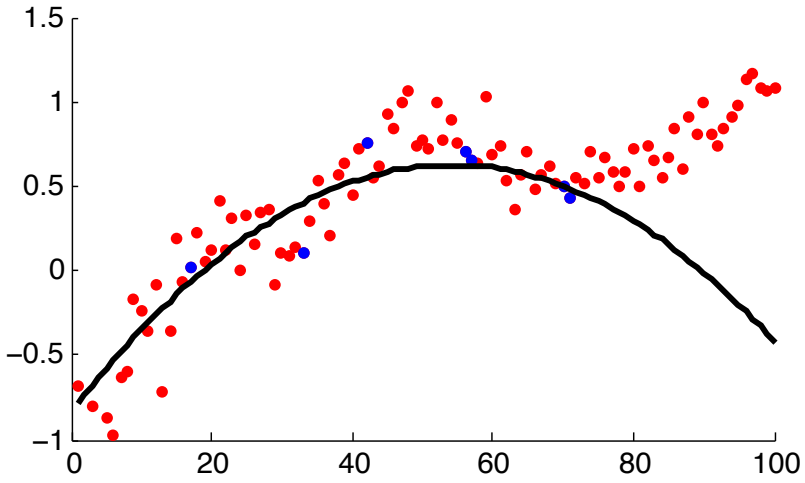

1.5
1
0.5
0
−0.5
−1
0
20
40
60
80
100

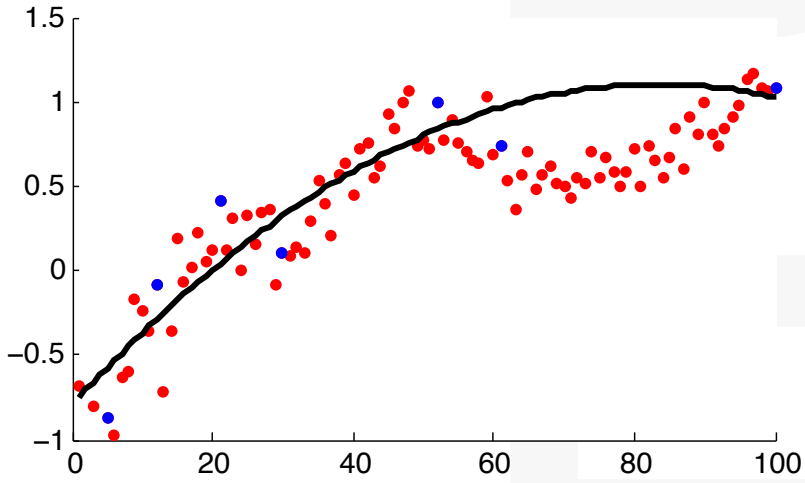

1.5
1
0.5
0
−0.5
−1
0
20
40
60
80
100

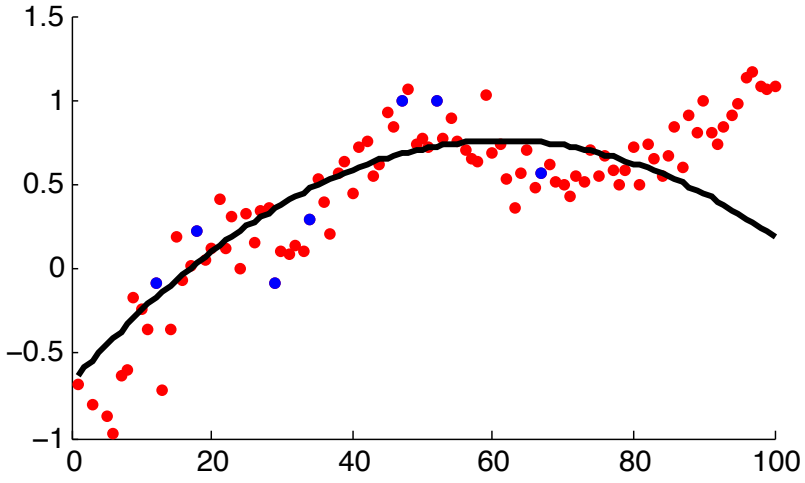

1.5
1
0.5
0
−0.5
−1
0
20
40
60
80
100

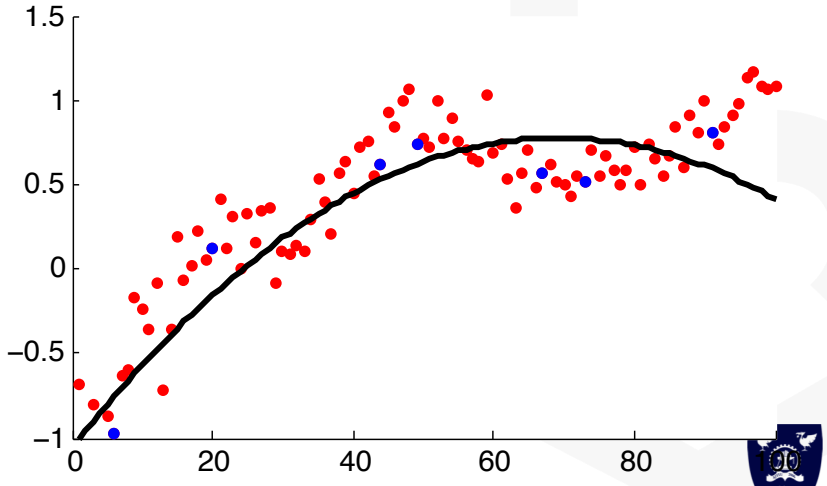

1.5
1
0.5
0
−0.5
−1
0
20
40
60
80
100

h(x|S)

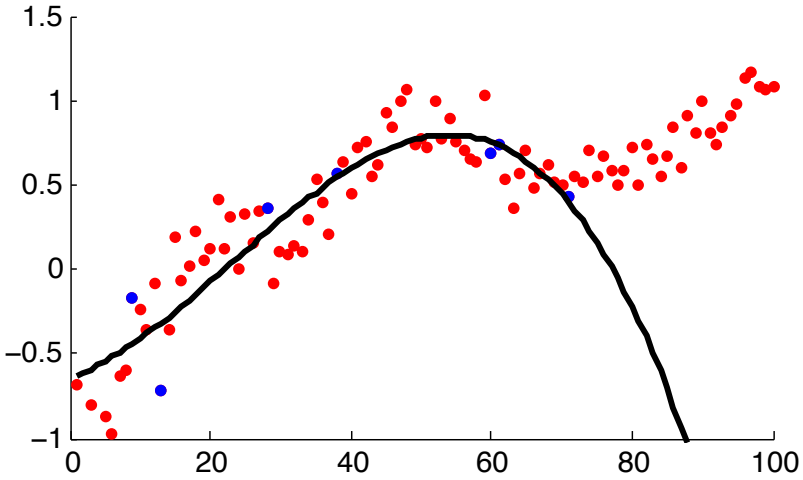

1.5
1
0.5
0
−0.5
−1
0
20
40
60
80
100

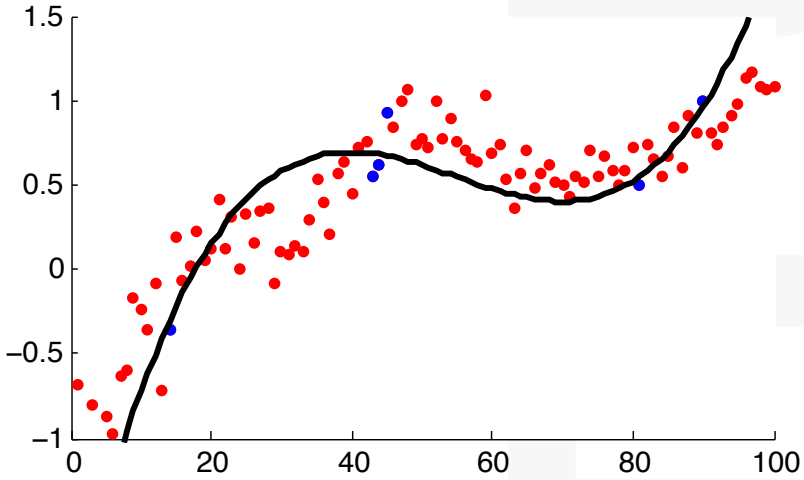

1.5
1
0.5
0
−0.5
−1
0
20
40
60
80
100

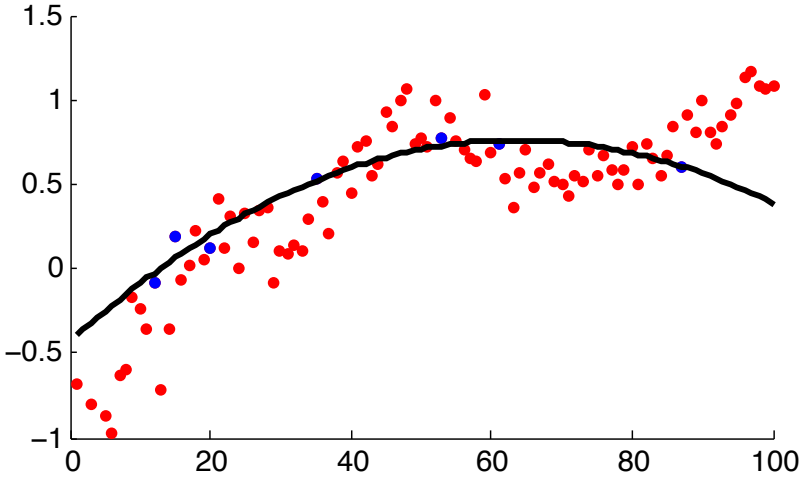

1.5
1
0.5
0
−0.5
−1
0
20
40
60
80
100

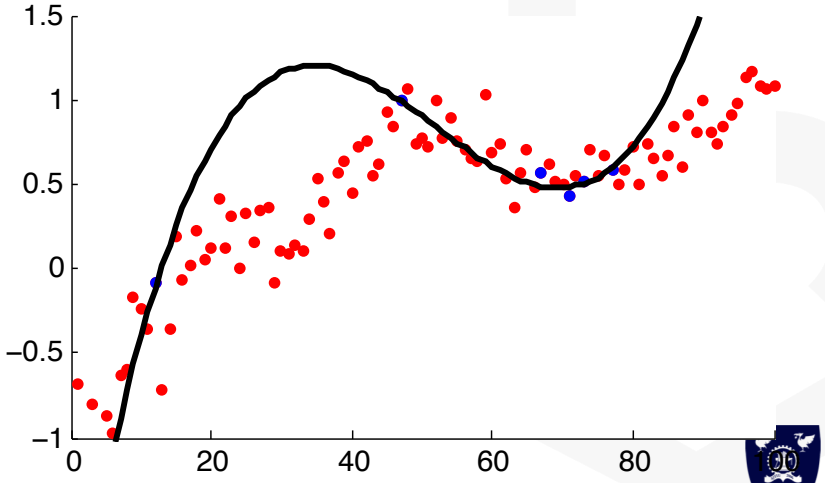

1.5
1
0.5
0
−0.5
−1
0
20
40
60
80
100

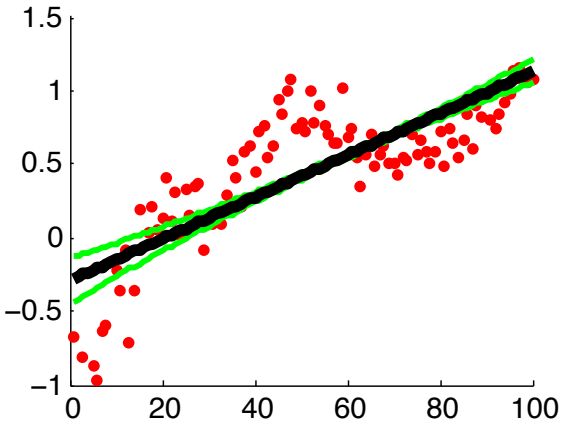
h(x|S)

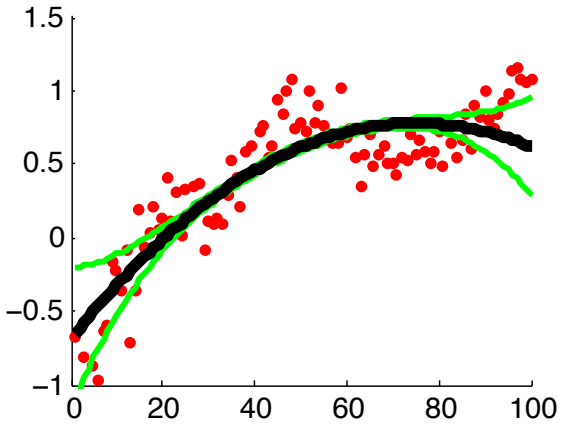

1.5
1
0.5
0
−0.5
−1
0
20
40
60
80
100

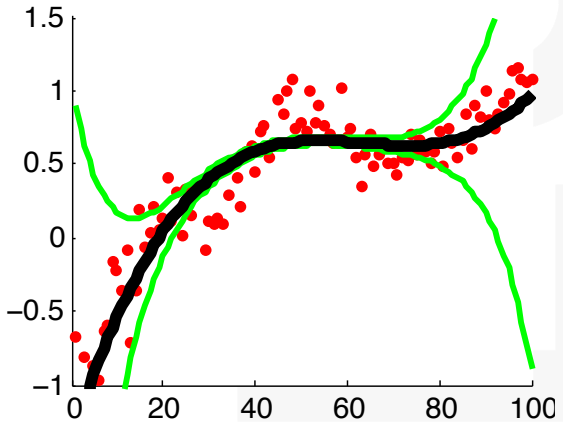

1.5
1
0.5
0
−0.5
−1
0
20
40
60
80
100

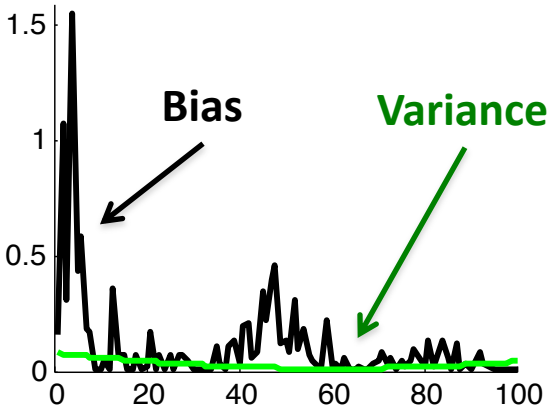

1.5
1
0.5
0
−0.5
−1
0
20
40
60
80
100

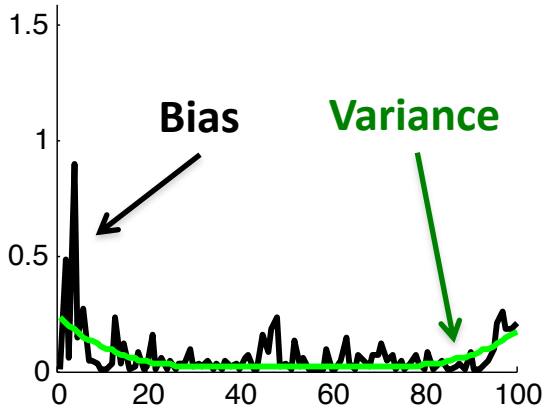

1.5
1
0.5
0
−0.5
−1
0
20
40
60
80
100

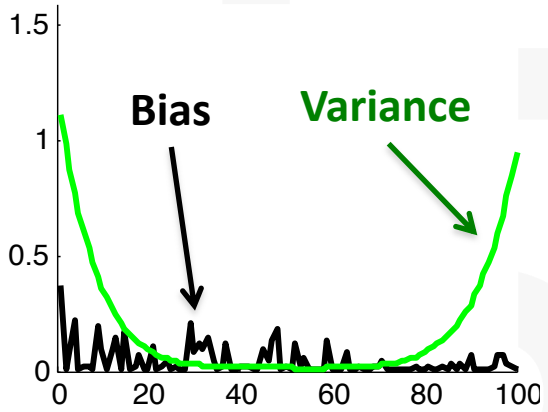












$$E[(h(x|S) - y)^2] = E[(Z-\check{z})^2] + \check{z}^2$$

Expected Error — Variance — Bias

$$Z = h(x|S) - y$$

$$\check{z} = E[Z]$$

HOW TO IMPROVE?

- Two Competing Methodologies
  - Build **one** really good model
    - Traditional approach
  - Build **many** models and average the results
    - Ensemble learning (more recent)

THE SINGLE MODEL PHILOSOPHY

- Motivation: Occam's Razor
  - "one should not increase, beyond what is necessary, the number of entities required to explain anything"
- Infinitely many models can explain any given dataset
  - Might as well pick the smallest one…

WHICH MODEL IS SMALLER?

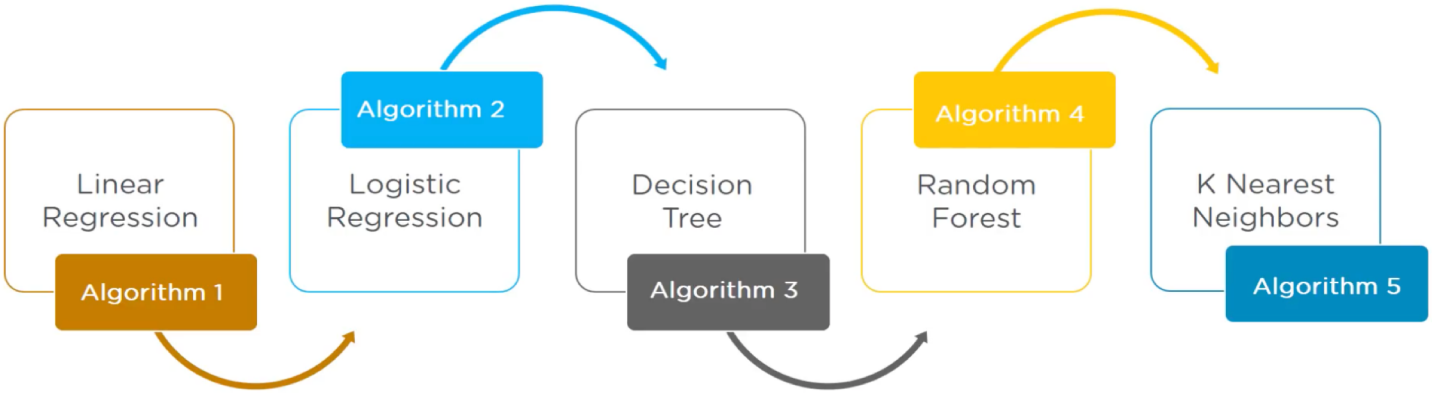
$$\hat{y} = f_1(x) = \sin(x)$$

or

$$\hat{y} = f_2(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots$$

- In this case $f_1(x) \equiv f_2(x)$
- It's not always easy to define small!

SINGLE MODELS:



ENSEMBLE PHILOSOPHY

- Build many models and combine them
- Only through averaging do we get at the truth!
- It's too hard (*impossible*?) to build a single model that works best
- Two types of approaches:
  - Models that don't use randomness
  - Models that incorporate randomness

ENSEMBLES OF CLASSIFIERS

- Idea
  - Combine the classifiers to improve the performance
- Ensembles of Classifiers
  - Combine the classification results from different classifiers to produce the final output
    - Unweighted voting
    - Weighted voting

EXAMPLE: WEATHER FORECAST

Reality							
1							
2							
3							
4							
5							
Combine							

HOW TO ENSEMBLE?

ENSEMBLE APPROACHES

- Bagging
  - **B**ootstrap aggregating
- Boosting
- Random Forests
  - Bagging reborn

OUTLINE

- **Bias/Variance Tradeoff**

- Ensemble methods that minimize variance
  - Bagging
  - Random Forests

- Ensemble methods that minimize bias
  - Functional Gradient Descent
  - Boosting
  - Ensemble Selection

OUTLINE

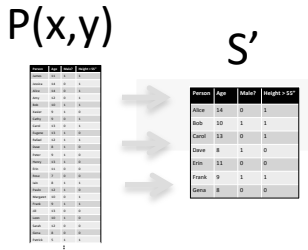
- Bias/Variance Tradeoff

- **Ensemble methods that minimize variance**
  - **Bagging**
  - **Random Forests**

- Ensemble methods that minimize bias
  - Functional Gradient Descent
  - Boosting
  - Ensemble Selection

BAGGING

- **Goal:** reduce variance
- **Ideal setting:** many training sets S'
  - Train model using each S'
  - Average predictions



sampled independently

Variance reduces linearly
Bias unchanged

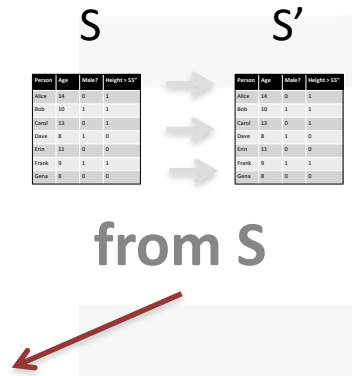
$$E_S[(h(x|S) - y)^2] = E[(Z-\check{z})^2] + \check{z}^2$$

Expected Error — **Variance** — **Bias**

$$Z = h(x|S) - y$$
$$\check{z} = E[Z]$$

"Bagging Predictors" [Leo Breiman, 1994] http://statistics.berkeley.edu/sites/default/files/tech-reports/421.pdf

BAGGING



- **Goal:** reduce variance
- **In practice:** resample S' with replacement
  - Train model using each S'
  - Average predictions

Variance reduces sub-linearly
(Because S' are correlated)
Bias often increases slightly

$$E[(h(x|S) - y)^2] = E[(Z-\check{z})^2] + \check{z}^2$$

Expected Error — **Variance** — **Bias**

$$Z = h(x|S) - y$$
$$\check{z} = E[Z]$$

Bagging = Bootstrap Aggregation

"Bagging Predictors" [Leo Breiman, 1994]
http://statistics.berkeley.edu/sites/default/files/tech-reports/421.pdf

THE BAGGING ALGORITHM

Given data: $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$

For $m = 1 : M$

- Obtain bootstrap sample D_m from the training data D
- Build a model $G_m(\mathbf{x})$ from bootstrap data D_m

THE BAGGING MODEL

- Regression

$$\hat{y} = \frac{1}{M}\sum_{m=1}^{M} G_m(\mathbf{x})$$

- Classification:
  - Vote over classifier outputs $G_1(\mathbf{x}),\ldots,G_M(\mathbf{x})$

BAGGING DETAILS

- Bootstrap sample of N instances is obtained by drawing N examples at random, with replacement.

- On average each bootstrap sample has 63% of instances
  - Encourages predictors to have uncorrelated errors
    - This is why it works

BAGGING

- **Question:** Do you see any problems?

- Still some overfitting if the trees are too large
- If trees are too shallow it can still underfit.
- interpretability
  - The **major drawback** of bagging (and other ***ensemble methods*** that we will study) is that the averaged model is no longer easily interpretable - i.e. one can no longer trace the 'logic' of an output through a series of decisions based on predictor values!

RANDOM FORESTS

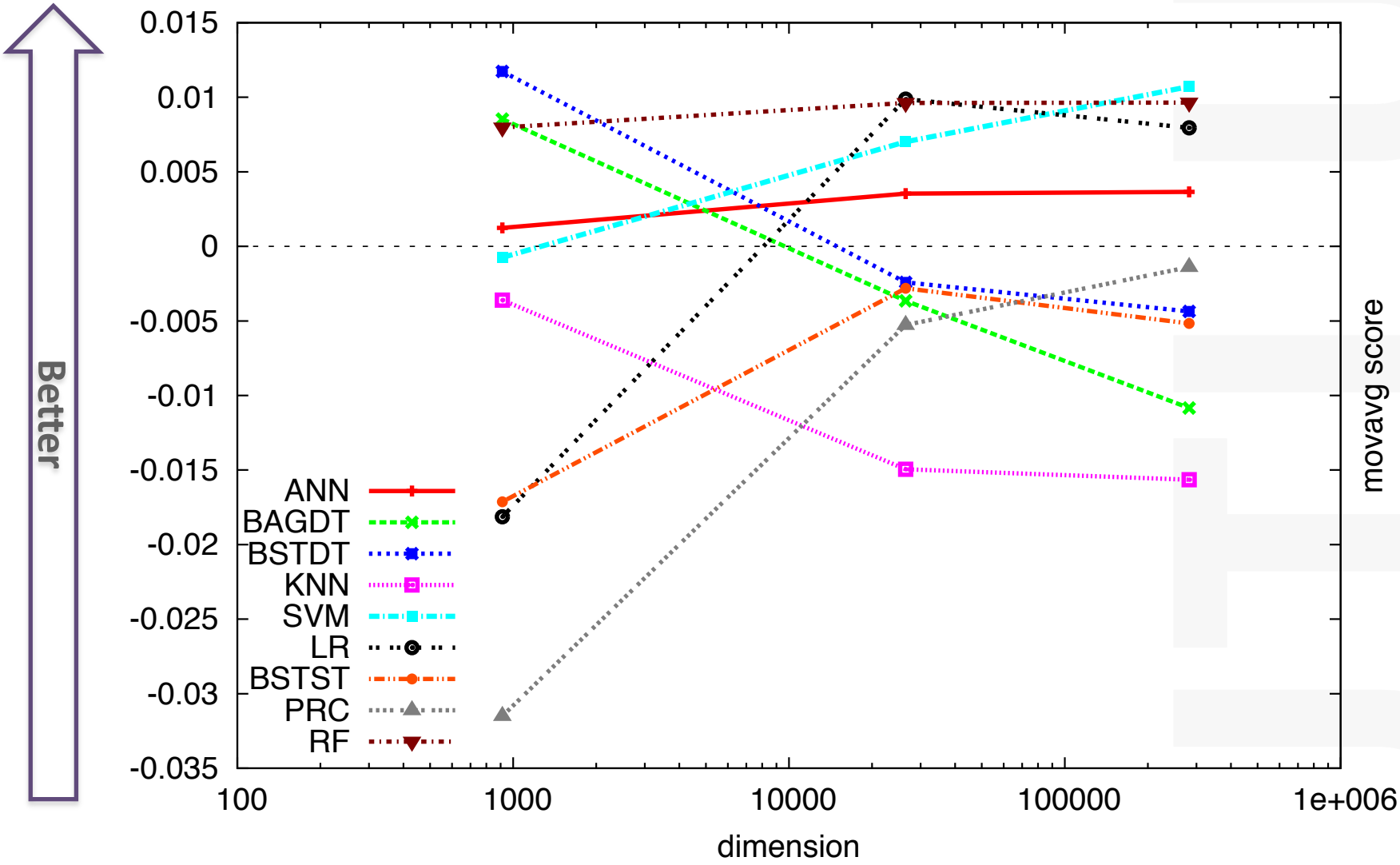
RANDOM FORESTS

- **Goal:** reduce variance
  - Bagging can only do so much
  - Resampling training data

- **Random Forests:** sample data & features!
  - Sample S'
  - Train DT
    - At each node, sample features (sqrt)
  - Average predictions

Further de-correlates trees

"Random Forests – Random Features" [Leo Breiman, 1997]
http://oz.berkeley.edu/~breiman/random-forests.pdf



Average performance over many datasets
Random Forests perform the best

"An Empirical Evaluation of Supervised Learning in High Dimensions"
Caruana, Karampatziakis & Yessenalina, ICML 2008

RANDOM FORESTS

Original Data

X	Y
X_1	y_1
X_2	y_2
X_3	y_3
X_4	y_4
X_5	y_5
⋮	⋮
X_n	y_n

Bootstrap Sample 1

X	Y
X_4	y_4
X_{14}	y_{14}
X_1	y_1
X_2	y_2
X_{35}	y_{35}
⋮	⋮
X_k	y_k

Decision Tree 1

Used and unused data

X	Y
X_1	y_1
X_2	y_2
X_3	y_3
X_4	y_4
X_5	y_5
⋮	⋮
X_n	y_n

RANDOM FORESTS

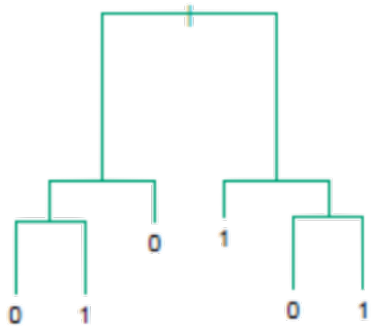
Original Data

X	Y
X_1	y_1
X_2	y_2
X_3	y_3
X_4	y_4
X_5	y_5
⋮	⋮
X_n	y_n

Bootstrap Sample 2

X	Y
X_5	y_5
X_3	y_3
X_{12}	y_{12}
X_{43}	y_{43}
X_1	y_1
⋮	⋮
X_k	y_k

Decision Tree 2



Used and unused data

X	Y
X_1	y_1
X_2	y_2
X_3	y_3
X_4	y_4
X_5	y_5
⋮	⋮
X_n	y_n

RANDOM FORESTS

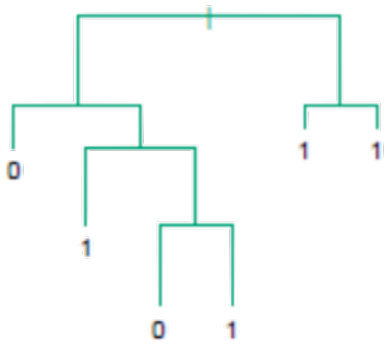
Original Data

X	Y
X_1	y_1
X_2	y_2
X_3	y_3
X_4	y_4
X_5	y_5
⋮	⋮
X_n	y_n

Bootstrap Sample 3

X	Y
X_9	y_9
X_4	y_4
X_1	y_1
X_1	y_1
X_{65}	y_{65}
⋮	⋮
X_k	y_k

Decision Tree 3



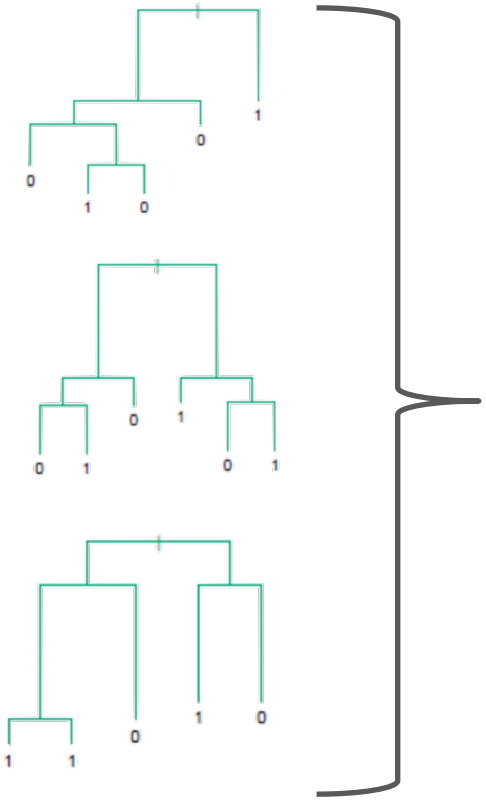
Used and unused data

X	Y
X_1	y_1
X_2	y_2
X_3	y_3
X_4	y_4
X_5	y_5
⋮	⋮
X_n	y_n

RANDOM FORESTS

X	Y
X_1	y_1
X_2	y_2
X_3	y_3
⋮	⋮
X_i	y_i
⋮	⋮
X_n	y_n

B Trees that did not see $\{X_i, y_i\}$



Classification

$$\hat{y}_{i,pw} = \text{majority}(\hat{y}_i)$$

Regression

$$\hat{y}_{i,pw} = \sum_{j \in B} \hat{y}_{i,j}$$

$$e_i = \mathbb{I}(\hat{y}_{i,pw} = y_i)$$

$$e_i = \left(y_i - \hat{y}_{i.pw}\right)^2$$

OOB ERROR

- We average the point-wise out-of-bag error over the full training set.

Classification

$$Error_{OOB} = \sum_{i}^{n} e_i = \sum_{i}^{n} \mathbb{I}(\hat{y}_{i,pw} = \mathrm{y_i})$$

Regression

$$Error_{OOB} = \sum_{i}^{n} e_i = \sum_{i}^{n} \left(y_i - \hat{y}_{i,pw}\right)^2$$

TUNING RANDOM FORESTS

- Random forest models have multiple hyper-parameters to tune:

    1. the number of predictors to randomly select at each split
    2. the total number of trees in the ensemble
    3. the minimum leaf node size

FINAL THOUGHTS ON RANDOM FORESTS

- When the number of predictors is large, but the number of relevant predictors is small, random forests can perform poorly.

- **Question**: Why?
- In each split, the chances of selected a relevant predictor will be low and hence most trees in the ensemble will be weak models.

FINAL THOUGHTS ON RANDOM FORESTS (CONT.)

- Increasing the number of trees in the ensemble generally does not increase the risk of overfitting.
- Again, by decomposing the generalization error in terms of bias and variance, we see that increasing the number of trees produces a model that is at least as robust as a single tree.
- However, if the number of trees is too large, then the trees in the ensemble may become more correlated, increase the variance.

THANK YOU

VISIT US

WWW.XJTLU.EDU.CN

FOLLOW US

@XJTLU

Xi'an Jiaotong-Liverpool University
西交利物浦大学