Lecture 11: 25 February, 2025

Madhavan Mukund

https://www.cmi.ac.in/~madhavan

Data Mining and Machine Learning January–April 2025

◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 - のへで

Limitations of classification models

- Bias : Expressiveness of model limits classification
 - For instance, linear separators
- Variance: Variation in model based on sample of training data
 - Shape of a decision tree varies with distribution of training inputs

Limitations of classification models

- Bias : Expressiveness of model limits classification
 - For instance, linear separators
- Variance: Variation in model based on sample of training data
 - Shape of a decision tree varies with distribution of training inputs

Models with high variance are expressive but unstable

- In principle, a decision tree can capture an arbitrarily complex classification criterion
- Actual structure of the tree depends on impurity calculation
- Danger of overfitting: model tied too closely to training set

Limitations of classification models

- Bias : Expressiveness of model limits classification
 - For instance, linear separators
- Variance: Variation in model based on sample of training data
 - Shape of a decision tree varies with distribution of training inputs

Models with high variance are expressive but unstable

- In principle, a decision tree can capture an arbitrarily complex classification criterion
- Actual structure of the tree depends on impurity calculation
- Danger of overfitting: model tied too closely to training set
- Is there an alternative to pruning?

Ensemble models

- Sequence of independent training data sets D_1 , D_2 , ..., D_k
- Generate models M_1 , M_2 , ..., M_k
- Take this ensemble of models and "average" them
 - For regression, take the mean of the predictions
 - For classification, take a vote among the results and choose the most popular one

4 E N

Ensemble models

- Sequence of independent training data sets D_1, D_2, \ldots, D_k
- Generate models M_1, M_2, \ldots, M_k
- Take this ensemble of models and "average" them
 - For regression, take the mean of the predictions
 - For classification, take a vote among the results and choose the most popular one
- Challenge: Infeasible to get large number of independent training samples
- Can we build independent models from a single training data set?
 - Strategy to build the model is fixed
 - Same data will produce same model

- Training data has N items
 - $TD = \{d_1, d_2, \dots, d_N\}$
- Pick a random sample with replacement

▶ < ∃ ▶</p>

- Training data has N items
 - $TD = \{d_1, d_2, \dots, d_N\}$
- Pick a random sample with replacement
 - Pick an item at random (probability $\frac{1}{N}$)
 - Put it back into the set
 - Repeat K times



< ∃ >

- Training data has N items
 - $TD = \{d_1, d_2, \dots, d_N\}$
- Pick a random sample with replacement
 - Pick an item at random (probability $\frac{1}{N}$)
 - Put it back into the set
 - Repeat K times
- Some items in the sample will be repeated

- Training data has N items
 - $TD = \{d_1, d_2, \dots, d_N\}$
- Pick a random sample with replacement
 - Pick an item at random (probability $\frac{1}{N}$)
 - Put it back into the set
 - Repeat K times
- Some items in the sample will be repeated
- If sample size is same as data size (K = N), expected number of distinct items is $(1 - \frac{1}{e}) \cdot N$ • Approx 63.2% Sould Sample



■ Sample with replacement of size *N* : bootstrap sample

Approx 2/3 of full training data

► < Ξ ►</p>

- Sample with replacement of size *N* : bootstrap sample
 - Approx 2/3 of full training data
- Take *k* such samples
- Build a model for each sample
 - Models will vary because each uses different training data

▶ < ∃ ▶</p>

- Sample with replacement of size *N* : bootstrap sample
 - Approx 2/3 of full training data
- Take k such samples
- Build a model for each sample
 - Models will vary because each uses different training data
- Final classifier: report the majority answer
 - Assumptions: binary classifier, k odd
- Provably reduces variance





Madhavan Mukund

Lecture 11: 25 February, 2025

DMML Jan–Apr 2025

6/16



Madhavan Mukund

Lecture 11: 25 February, 2025

DMML Jan–Apr 2025

< E

7/16



Lecture 11: 25 February, 2025

DMML Jan–Apr 2025



Lecture 11: 25 February, 2025

DMML Jan–Apr 2025



Madhavan Mukund

Lecture 11: 25 February, 2025

DMML Jan–Apr 2025

▶ ∢ ⊒

10/16



Lecture 11: 25 February, 2025

DMML Jan–Apr 2025

▶ ∢ ⊒

When to use bagging

- Bagging improves performance when there is high variance
 - Independent samples produce sufficiently different models

< ∃ >

When to use bagging

- Bagging improves performance when there is high variance
 - Independent samples produce sufficiently different models
- A model with low variance will not show improvement
 - k-nearest neighbour classifier
 - Given an unknown input, find k nearest neighbours and choose majority
 - Across different subsets of training data, variation in k nearest neighbours is relatively small
 - Bootstrap samples will produce similar models

Applying bagging to decision trees with a further twist



• • = • • = •

< □ > < 円

3

- Applying bagging to decision trees with a further twist
- As before, k bootstrap samples D_1, D_2, \ldots, D_k

► < Ξ ►</p>

3

- Applying bagging to decision trees with a further twist
- As before, k bootstrap samples D_1 , D_2 , ..., D_k
- For each D_i , build decision tree T_i as follows
 - Each data item has *M* attributes
 - Normally, choose maximum impurity gain among M attributes, then best among remaining $M 1, \ldots$

< □ > < 凸

(4) E (4) E (4)

- Applying bagging to decision trees with a further twist
- As before, k bootstrap samples D_1 , D_2 , ..., D_k
- For each D_i , build decision tree T_i as follows
 - Each data item has *M* attributes
 - Normally, choose maximum impurity gain among M attributes, then best among remaining $M 1, \ldots$
 - Instead, fix a small limit m < M say $m = \log_2 M + 1$
 - At each level, choose a random subset of available attributes of size m
 - Evaluate only these *m* attributes to choose next query

Image: A test in te

3

- Applying bagging to decision trees with a further twist
- As before, k bootstrap samples D_1 , D_2 , ..., D_k
- For each D_i , build decision tree T_i as follows
 - Each data item has *M* attributes
 - Normally, choose maximum impurity gain among M attributes, then best among remaining $M 1, \ldots$
 - Instead, fix a small limit m < M say $m = \log_2 M + 1$
 - At each level, choose a random subset of available actributes of size m
 - Evaluate only these *m* attributes to choose next query
 - No pruning build each tree to the maximum
- Final classifier: vote on the results returned by T_1, T_2, \ldots, T_k

Another optim "Decision strups" L'One node trees

Theoretically, overall error rate depends on two factors



- Correlation between pairs of trees higher correlation results in higher overall error rate
- Strength (accuracy) of each tree higher strength of individual trees results in lower overall error rate

DMML Jan-Apr 2025

→ < ∃→

- Theoretically, overall error rate depends on two factors
 - Correlation between pairs of trees higher correlation results in higher overall error rate
 - Strength (accuracy) of each tree higher strength of individual trees results in lower overall error rate
- Reducing *m*, the number of attributes examined at each level, reduces correlation and strength
 - Both changes influence the error rate in opposite directions

- Theoretically, overall error rate depends on two factors
 - Correlation between pairs of trees higher correlation results in higher overall error rate
 - Strength (accuracy) of each tree higher strength of individual trees results in lower overall error rate
- Reducing *m*, the number of attributes examined at each level, reduces correlation and strength
 - Both changes influence the error rate in opposite directions
- Increasing *m* increases both correlation and strength

- Theoretically, overall error rate depends on two factors
 - Correlation between pairs of trees higher correlation results in higher overall error rate
 - Strength (accuracy) of each tree higher strength of individual trees results in lower overall error rate
- Reducing *m*, the number of attributes examined at each level, reduces correlation and strength
 - Both changes influence the error rate in opposite directions
- Increasing *m* increases both correlation and strength
- Search for a value of *m* that optimizes overall error rate

< 3 >

• Each bootstrap sample omits about 1/3 of the data items

 $63\% \approx \frac{2}{3}$

▶ < ∃ ▶</p>

- Each bootstrap sample omits about 1/3 of the data items
- Hence, each data item is omitted by about 1/3 of the samples

- Each bootstrap sample omits about 1/3 of the data items
- Hence, each data item is omitted by about 1/3 of the samples
- If data item d does not appear in bootstrap sample D_i, d is out of bag (oob) for D_i

- Each bootstrap sample omits about 1/3 of the data items
- Hence, each data item is omitted by about 1/3 of the samples
- If data item d does not appear in bootstrap sample D_i, d is out of bag (oob) for D_i
- Oob classification for each *d*, vote only among those *T_i* where *d* is oob for *D_i*

- Each bootstrap sample omits about 1/3 of the data items
- Hence, each data item is omitted by about 1/3 of the samples
- If data item d does not appear in bootstrap sample D_i, d is out of bag (oob) for D_i
- Oob classification for each d, vote only among those T_i where d is oob for D_i
- Use oob samples to validate the model
 - Estimate generalization error rate of overall model based on error rate of oob classification
 - Do not require a separate test data set

4 E N

• What is the impurity gain of a feature across trees in ensemble?

- What is the impurity gain of a feature across trees in ensemble?
- Variation due to randomness of samples

- What is the impurity gain of a feature across trees in ensemble?
- Variation due to randomness of samples
- Even greater variation in a random forest

- What is the impurity gain of a feature across trees in ensemble?
- Variation due to randomness of samples
- Even greater variation in a random forest
- Compute weighted average of impurity gain
 - Weight is given by number of training samples at the node