INFO 6105
Data Science Engineering Methods and Tools

Lecture 8
Classification Trees

Ebrahim Nasrabadi
nasrabadi@northeastern.edu

College of Engineering
Northeastern University

Fall 2019

Classification Trees

- Tree-base methods can be applied to multi-class classification problems where

$$y_i \in \{\mathcal{C}_1, \ldots, \mathcal{C}_K\}.$$

Classification Trees

- Tree-base methods can be applied to multi-class classification problems where

$$y_i \in \{\mathcal{C}_1, \ldots, \mathcal{C}_K\}.$$

- Classification tees work much like regression trees. We need to modify

Classification Trees

- Tree-base methods can be applied to multi-class classification problems where

$$y_i \in \{\mathcal{C}_1, \ldots, \mathcal{C}_K\}.$$

- Classification tees work much like regression trees. We need to modify
  - criterion for splitting nodes. Instead of minimizing the RSS, we minimize a classification loss function.

Classification Trees

- Tree-base methods can be applied to multi-class classification problems where

$$y_i \in \{\mathcal{C}_1, \ldots, \mathcal{C}_K\}.$$

- Classification tees work much like regression trees. We need to modify
  - criterion for splitting nodes. Instead of minimizing the RSS, we minimize a classification loss function.
  - method for pruning tree.

Classification Trees

- Tree-base methods can be applied to multi-class classification problems where

$$y_i \in \{\mathcal{C}_1, \ldots, \mathcal{C}_K\}.$$

- Classification tees work much like regression trees. We need to modify
  - criterion for splitting nodes. Instead of minimizing the RSS, we minimize a classification loss function.
  - method for pruning tree.
- We predict the response by majority vote, i.e., pick the most common class in every region.

Notation

- Let node j represent the region R_j with N_j observations.

Notation

- Let node j represent the region R_j with N_j observations.
- Let $\hat{p}_{jk}$ be the proportion of observations within R_j in class k, i.e.,

$$\hat{p}_{jk} = \frac{1}{N_j} \sum_{i:x_i \in R_j} \mathbf{1}(y_i = k)$$

Notation

- Let node j represent the region R_j with N_j observations.
- Let $\hat{p}_{jk}$ be the proportion of observations within R_j in class k, i.e.,

$$\hat{p}_{jk} = \frac{1}{N_j} \sum_{i:x_i \in R_j} \mathbf{1}(y_i = k)$$

- The class prediction in region R_j is:

$$\hat{y}_{R_j} = \arg\max_k \hat{p}_{jk}$$

Notation

- Let node j represent the region R_j with N_j observations.
- Let $\hat{p}_{jk}$ be the proportion of observations within R_j in class k, i.e.,

$$\hat{p}_{jk} = \frac{1}{N_j} \sum_{i:x_i \in R_j} \mathbf{1}(y_i = k)$$

- The class prediction in region R_j is:

$$\hat{y}_{R_j} = \arg\max_k \hat{p}_{jk}$$

- The class probability distribution in region R_j is:

$$(\hat{p}_{j1}, \hat{p}_{j2}, \ldots, \hat{p}_{jK})$$

Classification losses

- Misclassification rate:

$$\sum_{j=1}^{|T|} q_j \sum_{i:x_j \in R_j} \mathbf{1}(y_i \neq \hat{y}_{R_j})$$

where q_j is the proportion of observations in R_j.

Classification losses

- Misclassification rate:

$$\sum_{j=1}^{|T|} q_j \sum_{i:x_j \in R_j} \mathbf{1}(y_i \neq \hat{y}_{R_j})$$

  where q_j is the proportion of observations in R_j.

- The cross-entropy

$$-\sum_{j=1}^{|T|} q_j \sum_{k=1}^{K} \hat{p}_{jk} \log\left(\hat{p}_{jk}\right)$$

Classification losses

- Misclassification rate:

$$\sum_{j=1}^{|T|} q_j \sum_{i:x_j \in R_j} \mathbf{1}(y_i \neq \hat{y}_{R_j})$$

  where q_j is the proportion of observations in R_j.

- The cross-entropy

$$-\sum_{j=1}^{|T|} q_j \sum_{k=1}^{K} \hat{p}_{jk} \log\left(\hat{p}_{jk}\right)$$

- The Gini index

$$\sum_{j=1}^{|T|} q_j \sum_{k=1}^{K} \hat{p}_{jk} \left(1 - \hat{p}_{jk}\right)$$

Classification losses

- The Gini index and cross-entropy are better measures of the purity of a region, i.e. they are low when the region is mostly one category.

Classification losses

- The Gini index and cross-entropy are better measures of the purity of a region, i.e. they are low when the region is mostly one category.
- They take on a small value if all of $\hat{p}_{jk}$'s are close to zero except one that is close to one.

Classification losses

- The Gini index and cross-entropy are better measures of the purity of a region, i.e. they are low when the region is mostly one category.
- They take on a small value if all of $\hat{p}_{jk}$'s are close to zero except one that is close to one.
- Motivation for Gini index: If instead of predicting the most likely class, we predict a random sample from the distribution

$$(\hat{p}_{j1}, \hat{p}_{j2}, \ldots, \hat{p}_{jK}),$$

the Gini index is the expected misclassification rate.

Classification losses

- The Gini index and cross-entropy are better measures of the purity of a region, i.e. they are low when the region is mostly one category.
- They take on a small value if all of $\hat{p}_{jk}$'s are close to zero except one that is close to one.
- Motivation for Gini index: If instead of predicting the most likely class, we predict a random sample from the distribution

$$(\hat{p}_{j1}, \hat{p}_{j2}, \ldots, \hat{p}_{jK}),$$

  the Gini index is the expected misclassification rate.
- It is typical to use the Gini index or cross-entropy for growing the tree, while using the misclassification rate when pruning the tree.

Binary Classification

- For two classes, if p is the proportion in the positive class, the three measures are

Binary Classification

- For two classes, if p is the proportion in the positive class, the three measures are

Binary Classification

- For two classes, if p is the proportion in the positive class, the three measures are

$$\text{Misclassification Rate} = 1 - \max(p, 1-p)$$

$$\text{Gini index} = 2p(1-p)$$

$$\text{Cross-entropy} = -p\log p - (1-p)\log(1-p)$$

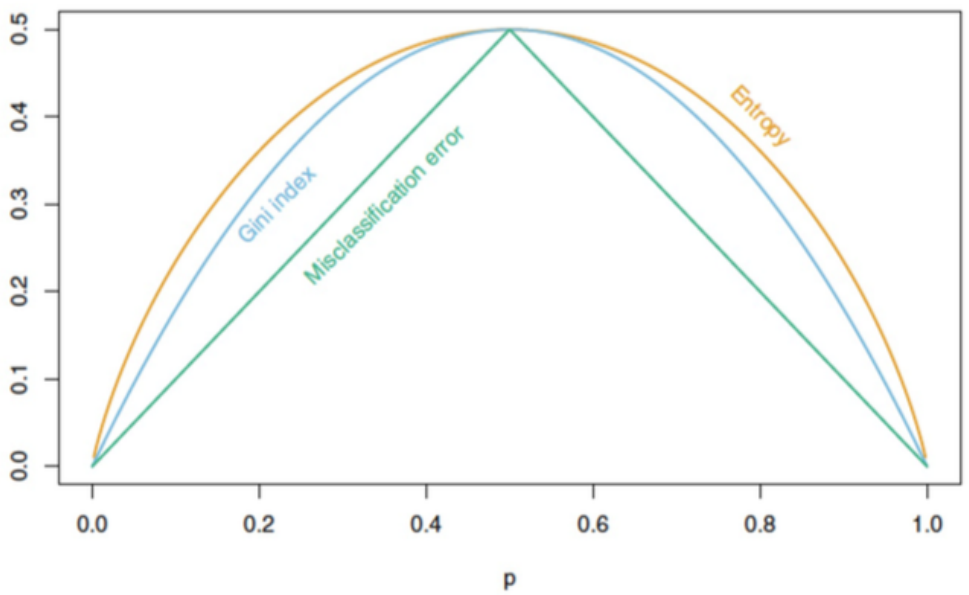
Binary Classification

- For two classes, if p is the proportion in the positive class, the three measures are

$$\text{Misclassification Rate} = 1 - \max(p, 1-p)$$

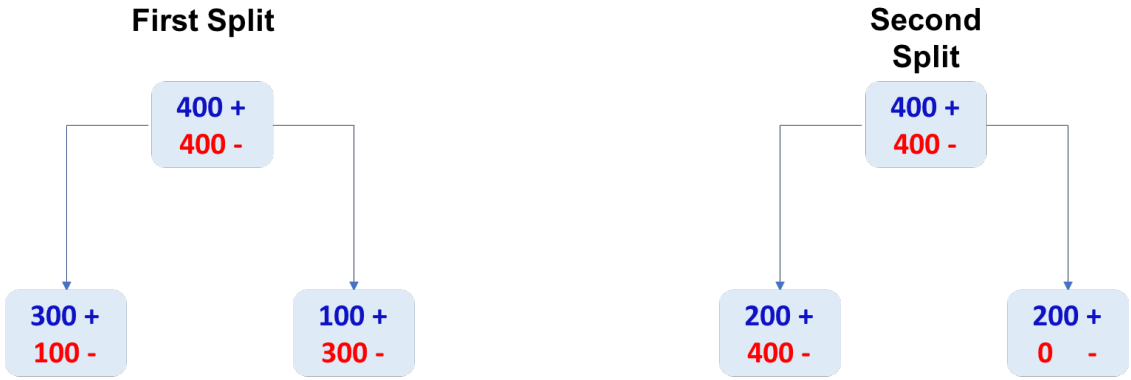
$$\text{Gini index} = 2p(1-p)$$

$$\text{Cross-entropy} = -p\log p - (1-p)\log(1-p)$$



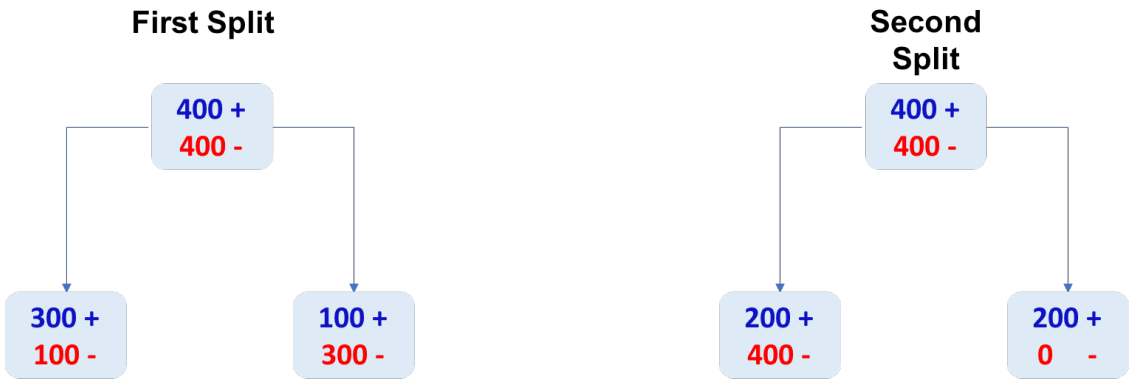
Example

- Consider a two-class problem with 400 observations in each class and two splits as:



Example

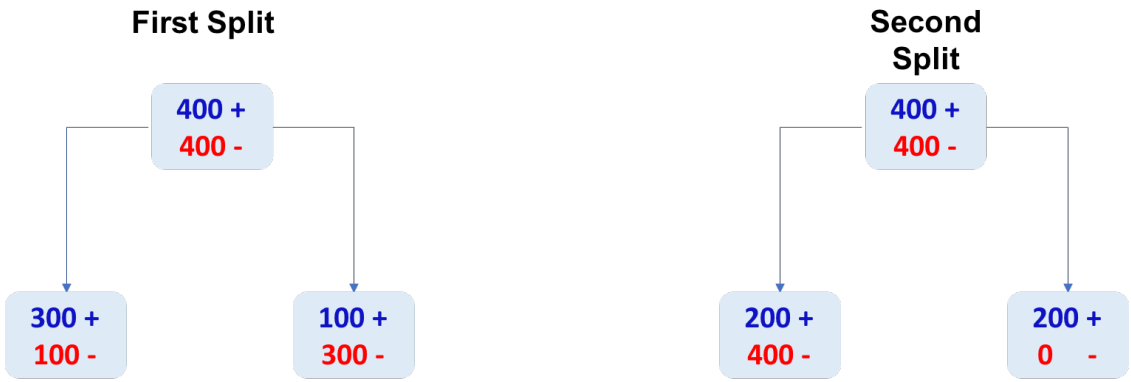
- Consider a two-class problem with 400 observations in each class and two splits as:



- Both splits produce a misclassification rate of 0.25, but the second split produces a pure node and is probably preferable.

Example

- Consider a two-class problem with 400 observations in each class and two splits as:



- Both splits produce a misclassification rate of 0.25, but the second split produces a pure node and is probably preferable.
- Both the Gini index and cross-entropy are lower for the second split. For this reason, either the Gini index or cross-entropy are typically used when growing the tree.

Selecting the best split

- Consider a node t with size N_t and impurity Q_t.

Selecting the best split

- Consider a node t with size N_t and impurity Q_t.
- For some variable j and split point s, we split node t into two nodes, say t_L and t_R with sizes N_{t_L} and N_{t_R} and impurities Q_{t_L} and Q_{t_R}

Selecting the best split

- Consider a node t with size N_t and impurity Q_t.
- For some variable j and split point s, we split node t into two nodes, say t_L and t_R with sizes N_{t_L} and N_{t_R} and impurities Q_{t_L} and Q_{t_R}
- The average decrease of impurity is

$$\Delta(j,s) = Q_t - \frac{1}{N_t}\left(\frac{N_{t_L}}{N_t}Q_{t_R} + \frac{N_{t_L}}{N_t}Q_{t_R}\right)$$

Selecting the best split

- Consider a node t with size N_t and impurity Q_t.
- For some variable j and split point s, we split node t into two nodes, say t_L and t_R with sizes N_{t_L} and N_{t_R} and impurities Q_{t_L} and Q_{t_R}
- The average decrease of impurity is

$$\Delta(j,s) = Q_t - \frac{1}{N_t}\left(\frac{N_{t_L}}{N_t}Q_{t_R} + \frac{N_{t_L}}{N_t}Q_{t_R}\right)$$

- We select at each step the splitting variable j and the split point s that maximizes $\Delta(j,s)$ or, equivalently, that minimizes the average impurity

$$\frac{N_{t_L}}{N_t}Q_{t_R} + \frac{N_{t_L}}{N_t}Q_{t_R}$$

Categorical predictors

- When splitting a categorical variable having q unordered values, there are $2^{q-1} - 1$ possible partitions of the q values into two groups.

Categorical predictors

- When splitting a categorical variable having q unordered values, there are $2^{q-1} - 1$ possible partitions of the q values into two groups.
- It becomes computationally expensive for large q.

Categorical predictors

- When splitting a categorical variable having q unordered values, there are $2^{q-1} - 1$ possible partitions of the q values into two groups.
- It becomes computationally expensive for large q.
- In the 2-class case, this computation can be simplified.

Categorical predictors

- When splitting a categorical variable having q unordered values, there are $2^{q-1} - 1$ possible partitions of the q values into two groups.
- It becomes computationally expensive for large q.
- In the 2-class case, this computation can be simplified.
  - We order the predictor values according to the proportion falling in positive class.

Categorical predictors

- When splitting a categorical variable having q unordered values, there are $2^{q-1} - 1$ possible partitions of the q values into two groups.
- It becomes computationally expensive for large q.
- In the 2-class case, this computation can be simplified.
  - We order the predictor values according to the proportion falling in positive class.
  - Then we split this predictor as if it were an ordered predictor.

Categorical predictors

- When splitting a categorical variable having q unordered values, there are $2^{q-1} - 1$ possible partitions of the q values into two groups.
- It becomes computationally expensive for large q.
- In the 2-class case, this computation can be simplified.
  - We order the predictor values according to the proportion falling in positive class.
  - Then we split this predictor as if it were an ordered predictor.
  - One can show this gives the optimal split, in terms of cross-entropy or Gini index, among all possible $2^{q-1} - 1$ splits.

Categorical predictors

- The partitioning algorithm tends to favor categorical predictors with many levels q;

Categorical predictors

- The partitioning algorithm tends to favor categorical predictors with many levels q;
- the number of partitions grows exponentially in q,

Categorical predictors

- The partitioning algorithm tends to favor categorical predictors with many levels q;
- the number of partitions grows exponentially in q,
- the more choices we have, the more likely we can find a good one for the data at hand

Categorical predictors

- The partitioning algorithm tends to favor categorical predictors with many levels q;
- the number of partitions grows exponentially in q,
- the more choices we have, the more likely we can find a good one for the data at hand
- This can lead to severe overfitting if q is large, and such variables should be avoided

Advantages and Disadvantages of Trees

- Very easy to interpret

Advantages and Disadvantages of Trees

- Very easy to interpret
- Easy to visualize (especially if they are small).

Advantages and Disadvantages of Trees

- Very easy to interpret
- Easy to visualize (especially if they are small).
- Handle qualitative predictors without the need to create dummy variables

Advantages and Disadvantages of Trees

- Very easy to interpret
- Easy to visualize (especially if they are small).
- Handle qualitative predictors without the need to create dummy variables
- Invariant to monotone transformation of predictor variables

Advantages and Disadvantages of Trees

- Very easy to interpret
- Easy to visualize (especially if they are small).
- Handle qualitative predictors without the need to create dummy variables
- Invariant to monotone transformation of predictor variables
- Unfortunately, trees generally do not have the same level of predictive accuracy as some of the other modern regression and classification approaches.