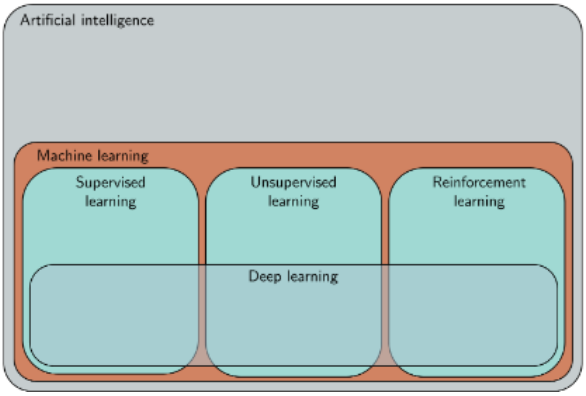
COMP 9414 Week 5 - Planning and Reasoning About Action

Author: Junji Duan

Last Update: 2025/10/19

1 W6A: Machine Learing and Decision tree



1.1 Different Subfields of AI Algorithms

- Supervised machine learning
  - Model is trained on a labeled dataset (i.e., the target or outcome variable is known).
    - Regression algorithms: predict output values by identifying linear relationships between real or continuous values (e.g., temperature, salary). Regression algorithms include linear regression, random forest and gradient boosting, as well as other subtypes.
    - Classification algorithms: predict categorical output variables (e.g., "junk" or "not junk") by labeling pieces of input data. Classification algorithms include logistic regression, k-nearest neighbors and support vector machines (SVMs), among others.
    - Naïve Bayes classifiers: enable classification tasks for large datasets. They' re also part of a family of generative learning algorithms that model the input distribution of a given class or/category. Naïve Bayes algorithms include decision trees, which can actually accommodate both regression and classification algorithms.
    - Neural networks: simulate the way the human brain works, with a huge number of linked processing nodes that can facilitate processes like natural language translation, image recognition, speech recognition and image creation.
    - Random forest algorithms: predict a value or category by combining the results from a number of decision trees.
- Unsupervised machine learning
  - Learning about a dataset without labels
    - K-means clustering: assigns data points into K groups, where the data points closest to a given centroid are clustered under the same category and K represents clusters based on their size and level of granularity. K-means clustering is commonly used for market segmentation, document clustering, image segmentation and image compression.
    - Hierarchical clustering: describes a set of clustering techniques, including agglomerative clustering—where data points are initially isolated into groups and then merged iteratively based on similarity until one cluster remains—and divisive clustering—where a single data cluster is divided based on the differences between data points.
    - Probabilistic clustering: helps solve density estimation or "soft" clustering problems by grouping data points based on the likelihood that they belong to a particular distribution.

- Reinforcement Learning
  - Type of dynamic programming that trains algorithms using a system of reward and punishment

1.2 Supervised Learning: Decision Trees

Decision trees are a classical supervised machine learning method.

Because the splitting rules can be naturally represented in a tree structure, these models are referred to as decision-tree methods.

Decision trees are widely used for both classification and regression tasks.

Basic decision trees partition the data into subsets that are increasingly homogeneous(同质/均匀) with respect to the response variable.

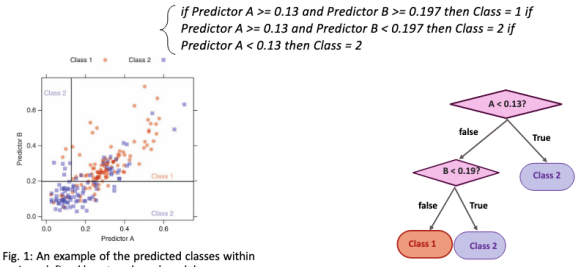


Fig. 1: An example of the predicted classes within regions defined by a tree-based model.

The ultimate goal is to construct a decision tree that generalizes well from the training data and accurately classifies previously unseen samples.

1.2.1 Decision Trees, Entropy

- Entropy is a measure of randomness or uncertainty in a random variable.
  - Higher entropy means more randomness
  - "Information" (about distribution) reduces entropy
  - It is maximized when all outcomes are equally likely.
  - It is minimized when the probability distribution is highly concentrated around a single outcome.
- Idea: Split based on information gain and measure information gain in bits.

Definition: If the prior probabilities of n attribute values are $p_1, \cdots, p_n$, then the entropy of the distribution is:

$$H\left(\langle \mathrm{p}_1, \cdots, \mathrm{pn} \rangle\right) = \sum_{i=1}^{n} -\mathrm{p}_i \log_2 p_i$$

1.2.2 Decision Trees, Minimal Error Pruning

Following Ockham's Razor, prune branches that do not provide much benefit in classifying the items (aids generalization, avoids overfitting).

For a leaf node, all items will assign the majority class at that node. Estimate error rate on the (unseen) test items using the Laplace error :

$$E = 1 - \frac{n+1}{N+k}$$

N = total number of (training) items at the node
n = number of (training) items in the majority class
k = number of classes
If the average Laplace error of the children exceeds that of the parent node, prune off the children.

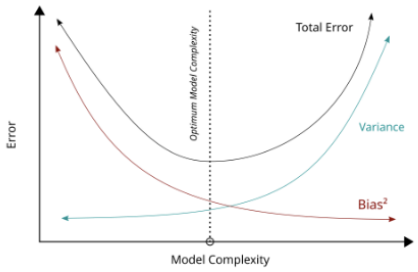
1.2.3 Tree Limitations

- Models based on single trees have particular weaknesses.
  - Model instability: Small changes in the data can drastically alter the structure of the tree or rules, leading to high variance.
  - Suboptimal predictive performance: Single trees often fail to achieve strong generalization.
    - Generalization refers to the ability of a machine learning model to perform well on previously unseen data, not just the data it was trained on.
- When training a decision tree, the dataset is typically divided into three subsets: training, validation, and test.
  - Training set: Used to train the decision tree model.
  - Validation set: Used to evaluate different configurations (hyperparameters) during training.
  - Test set: Provides an unbiased evaluation of the final decision tree model on unseen data.
- Variance measures how much a model's predictions would change if it were trained on different datasets drawn from the same distribution.

- High variance → The model tends to ”memorize” the training data (overfitting) rather than learning the underlying patterns.
- Low variance → The model is more stable and consistent across different datasets.

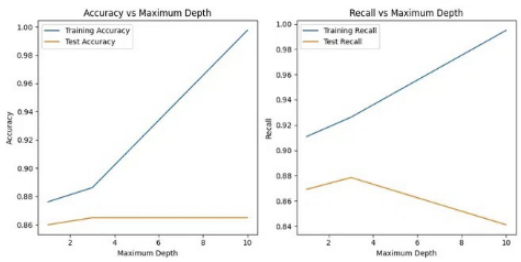
A complex model would help us to avoid bias (underfitting) and limiting the model complexity would help us to avoid overfitting (high variance).

There is a trade-off here as it is known as bias-variance tradeoff.



1.2.4 Tree Depth

- The depth of a tree is the length of the longest path from the root node to any leaf node.
  - Shallow trees suffer from high bias (underfitting) because the tree does not have enough nodes to capture the complexity of the data.
  - Deep trees suffer from high variance (overfitting) because the tree may assign a unique path to each data sample rather than learning general patterns.
- The optimal depth should be determined based on factors such as dataset size, data complexity, and available hardware resources.



2 W6B: Ensemble Learning 集成学习

Models based on single trees always have a less-than-optimal predictive performance. To address these issues, researchers developed ensemble methods, which combine many trees (or other machine learning models) into a single, more robust model.

While ensemble learning can refer to the combination of various machine learning approaches, in this lecture we will focus specifically on tree-based ensemble methods.

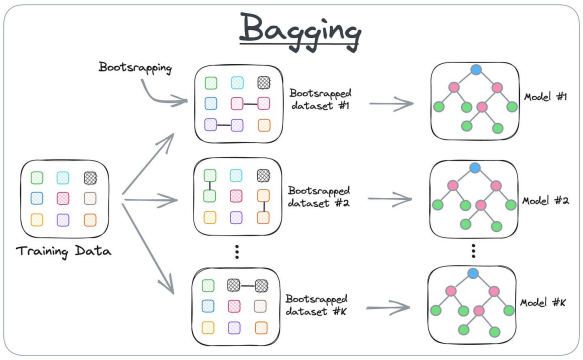
Ensembles generally provide much better predictive performance than individual models.

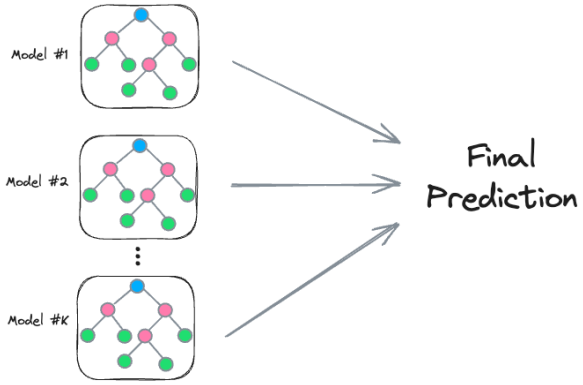
Ensembles generally achieve much better predictive performance than single trees, and this is often true for other types of models as well.

Decision Trees (prone to high variance) -> Bagged trees (improve high variance of decision tree) -> Random forests (improve variance of bagged trees) -> Boosting (improve bias of random forests)

2.1 Ensemble Learning - Bagged Trees

- Bagging (short for bootstrap aggregation) is a general ensemble method that can be applied to any regression or classification model and relies on bootstrapping to create multiple training datasets.
  - A bootstrap sample is a random sample of the data taken with replacement. Once a data point is selected for the sample, it remains available for subsequent selections. Typically, a bootstrap sample is the same size as the original dataset.
  - As a result, some data points may appear multiple times in a sample, while others may not be selected at all.
  - Data points that are not included in a bootstrap sample are commonly referred to as out-of-bag (OOB) samples.





Each model in the ensemble generates a prediction for a new sample, and the final prediction is made using the majority vote rule (for classification) or by averaging the predictions (for regression).

2.1.1 Advantages of Bagging

- No separate test set required: Bagging allows us to estimate model performance without needing a separate dataset, saving both time and data, as all training data can still be used for building the model.
- Each tree is trained on a bootstrap sample, which typically contains about 63% of the data.
- The remaining $\sim 37\%$ of the data, known as out-of-bag (OOB) samples, act as "unseen test data."
- Testing each tree on its OOB samples provides an unbiased estimate of performance.
- Bagging effectively reduces the variance of predictions through its aggregation process.
- When predictions for a sample are averaged, the average prediction has lower variance than the variance of the individual predictions.
- By aggregating many such predictions, these errors partially cancel out, resulting in a more stable and less variable overall prediction.

2.1.2 Variance

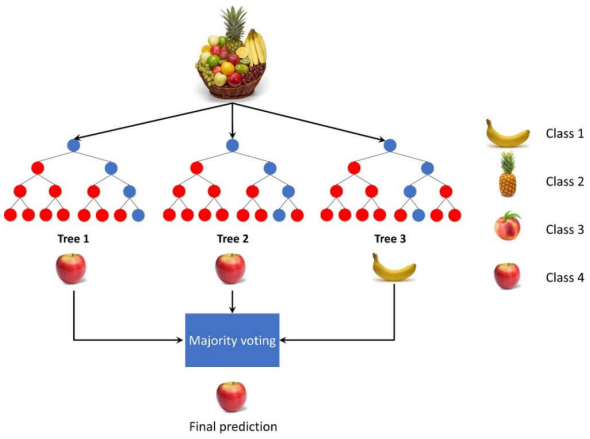
$$\text{Variance}\left(\hat{f}_{\text{final}}(x)\right) = \rho\sigma^2 + \frac{1-\rho}{M}\sigma^2$$

- If $\rho = 0$ (independent models): variance shrinks as 1/M.
- If $\rho = 1$ (perfectly correlated learners): variance stays at σ^2, no gain from averaging.
- For bagging, the variance of the ensemble is reduced by a factor of M (the number of models).

2.1.3 Conclusion

- In bagging, each tree is ultimately unique - no two trees are exactly the same.
- However, the trees are still somewhat correlated(相关) with each other (they are not independent trees).
- As a result, the variance reduction achieved by bagging can be improved further if we can produce more decorrelated trees (decrease ρ).
- Reducing correlation among predictors can be accomplished by introducing additional randomness into the tree construction process.

2.2 Ensemble Learning - Random Forests



- Each model in the ensemble is then used to generate a prediction for a new sample, and the final prediction is made using the majority vote rule (for classification) or by averaging the predictions (for regression).

- At each split in a tree, the algorithm randomly selects **k** predictors (features). Typically, $k = \sqrt{p}$, where p is the total number of features.
- The tree then chooses the best split only from those k features.
- This randomness reduces correlation between trees (the problem with bagged trees).

Compared to bagging, random forests is more computationally efficient on a tree-by-tree basis since the tree building process only needs to evaluate a fraction of the original features at each split, although more trees are usually required by random forests.

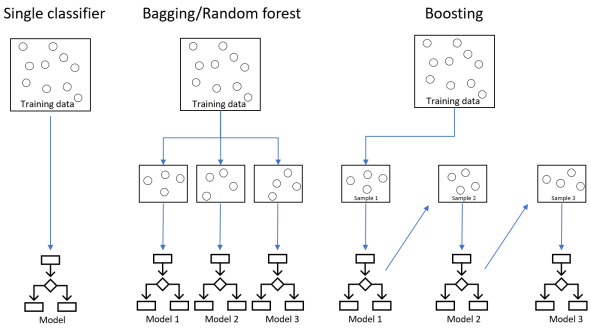
2.2.1 Conclusion

Choosing the number of trees (m) in Random Forests and Bagging:

- **Variance reduction**: Adding more trees decreases variance, making predictions more stable.
- **Constraints**: Training time, memory usage, and overfitting are the main limiting factors when increasing m .
- **Test error behavior**: The test error typically decreases monotonically as m increases - rapidly at first and then levelling off, becoming nearly constant after a sufficient number of trees.

2.3 Ensemble Learning - Boosting Trees

AdaBoost方法是一种迭代算法，在每一轮中加入一个新的弱分类器，直到达到某个预定的足够小的错误率。每一个训练样本都被赋予一个权重，表明它被某个分类器选入训练集的概率。如果某个样本点已经被准确地分类，那么在构造下一个训练集中，它被选中的概率就被降低；相反，如果某个样本点没有被准确地分类，那么它的权重就得到提高。通过这样的方式，AdaBoost方法能"聚焦于"那些较难分（更富信息）的样本上。在具体实现上，最初令每个样本的权重都相等，对于第K次迭代操作，我们就根据这些权重来选取样本点，进而训练分类器C_k。然后就根据这个分类器，来提高被它分错的样本的权重，并降低被正确分类的样本权重。然后，权重更新过的样本集被用于训练下一个分类器C_k[2]。整个训练过程如此迭代地进行下去。



There are many types of boosting algorithms, but here we focus on one of the most influential: **AdaBoost**.

- At each iteration, AdaBoost selects the best classifier based on the current weighted data points.
- Data points that are misclassified in the k-th iteration receive higher weights in the $(k+1)$-st iteration.
- As a result, samples that are difficult to classify gradually receive increasingly larger weights.
- This process ensures that each iteration focuses on learning a different aspect of the data.
- Finally, the sequence of weighted classifiers is combined into an ensemble, producing a strong overall model.

1. Initialize the observation weights $w_i=1/N$, i=1,2,...,N.
2. For m=1 to M:
   (a) Fit a classifier $G_m(x)$ to the training data using weights w_i.
   (b) Compute
   $$err_m = \frac{\sum_1^N w_i\, I(y_i != G_m(x_i))}{\sum_1^N w_i}$$
   (c) Compute the influence α_m
   $$\alpha_m = ½\ \ln\frac{1-errm}{err_m}$$
   (d) Set $w_i^{new\_iteration} = w_i^{old\_iteration}\, e^{\pm\alpha}$, i=1,2,...,N.
3. Output $G(x)=sign[\sum_{m=1}^M \alpha_m G_m(x)]$.

Algorithm 3: AdaBoost algorithm with *M* decision trees in ensemble for classification problems.

- Boosting builds an ensemble of learners sequentially:
  - The first predictions contain some bias (errors).
  - Boosting focuses on the mistakes by assigning greater weight to misclassified predicted samples.
  - The next model is trained to correct those errors.
  - Each new model becoming "specialized" in fixing the weaknesses of the previous ones.
- By building trees sequentially - where each new tree corrects the errors of the previous ones - boosting reduces bias and improves predictive performance.

2.4 Ensemble Learning, Summary

Aspect	Bagging	Random Forests	Boosting
Training	Parallel	Parallel	Sequential
Sampling	Bootstrapping	Bootstrapping + feature randomness	Weighted sampling (focus on errors)
Bias	Doesn't reduce much	Doesn't reduce much	Reduces bias significantly
Variance	Reduces variance	Strong variance reduction	Reduces variance + bias

- Trees are sets of if-then rules used for classification or prediction.
  - They can be built using entropy (or other criteria) but typically suffer from high variance.
- Bagging mitigates high variance by training multiple trees on bootstrap resamples and aggregating their predictions.
- Random Forests improve variance reduction further by adding randomness to training (e.g., selecting random subsets of features at each split).
  - Trees are trained in parallel and weighted equally, which reduces variance well but does not significantly reduce bias.
- Boosting reduces both bias and variance by training trees sequentially, where each new tree focuses on correcting the mistakes of the previous ones, and weighting trees according to their performance.

参考文献

[1] UNSW COMP9414 Lecture slides –Prof. W.Wobcke

[2] Other online resources (eg.graph)