Machine Learning
(IT3190E)

Quang Nhat NGUYEN

quang.nguyennhat@hust.edu.vn

Hanoi University of Science and Technology

School of Information and Communication Technology

Academic year 2020-2021

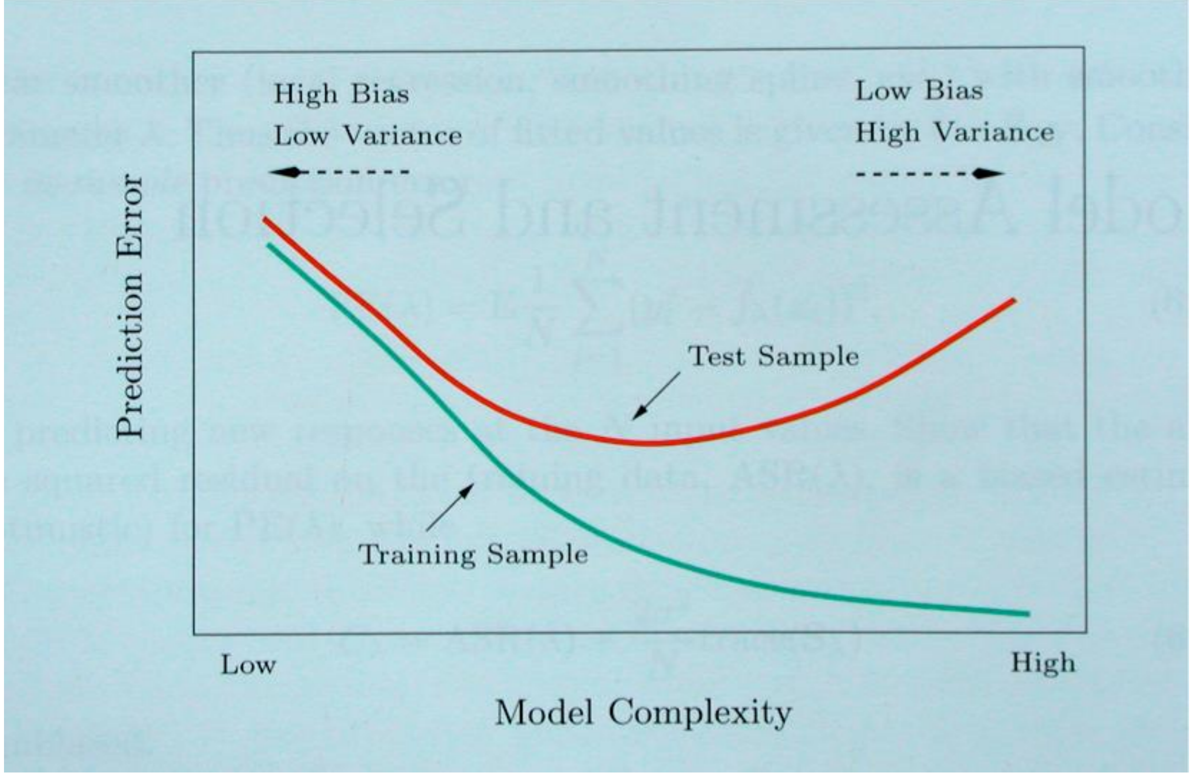
The course's content:

- Introduction
- Performance evaluation of ML system
- Supervised learning
- Unsupervised learning
- **Ensemble learning**
  - **Problem of bias vs. variance trade-off**
  - **Strategies for combining classifiers**
  - **Ensemble learning based on data sampling**
  - **Ensemble learning based on classifiers stacking**
- Reinforcement learning

Problem of bias vs. variance trade-off

Trade-off of bias vs. variance

- High bias typically occurs when under-fitting the data
- High variance typically occurs when over-fitting the data



[Hastie, Tibshirani, Friedman "Elements of Statistical Learning" 2001]

Unstable learners

- Beyond the problems linked to under-fitting/over-fitting, some predictive models are inherently **unstable**
  - Small differences in the learning dataset might lead to very different predictions (regression/classification)
    - Sensitivity to outliers
    - Sensitivity to irrelevant variables
  - Resulting in a large variance in the prediction

Unstable classifiers

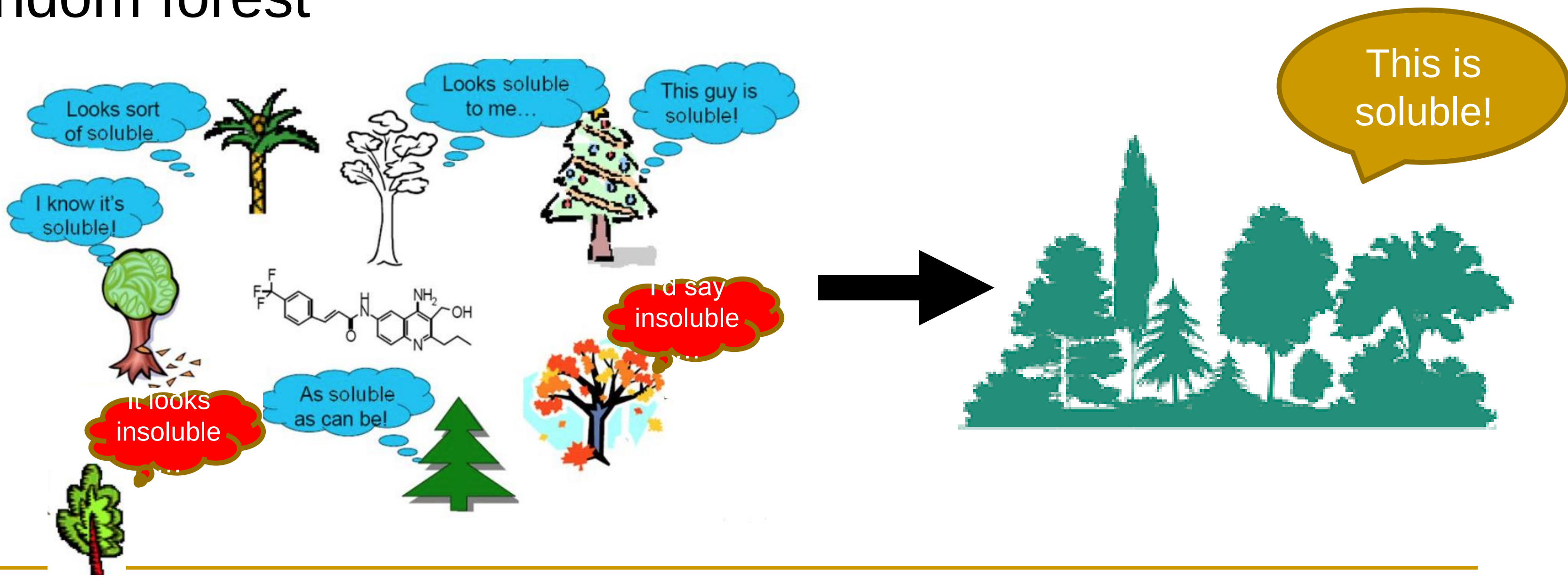
- Decision trees are unstable
  - Example from https://towardsdatascience.com/the-indecisive-decision-tree-story-of-an-emotional-algorithm-1-2-8611eea7e397:

I trained 10 decision trees having exact same hyper-parameters with 10 different seed values, which ensures that each tree trains with a slightly different sample. Let's look at the top three most important variables, which represent structure of the tree, obtained by the trees for both the data sets.

	seed	train_acc	valid_acc	top_feature	second_feature	third_feature
0	0	0.903439	0.785185	Title	Fare	Age
1	288	0.888889	0.748148	Sex	Age	Fare
2	576	0.879630	0.740741	Title	Fare	Age
3	864	0.892857	0.792593	Sex	Age	Fare
4	1152	0.883598	0.748148	Sex	Age	Fare
5	1440	0.886243	0.725926	Title	Age	Sex
6	1728	0.895503	0.703704	Fare	Sex	Age
7	2016	0.894180	0.770370	Title	Fare	Age
8	2304	0.902116	0.792593	Sex	Pclass	Fare
9	2592	0.871693	0.777778	Title	Age	Fare
10	2880	0.898148	0.785185	Title	Age	Pclass

Solution: Ensemble learning

- Idea of ensemble learning methods:
  - “United we stand, divided we fall”
- Objective of ensemble learning methods
  - Reducing variance/unstability of single predictive models by combining multiple models
- Random forest



Solution: Ensemble learning

- Ensemble learning can be applied for both supervised and unsupervised learning
  - Supervised learning: regression or classification
    - *E.g.*, Random forest
  - Unsupervised learning: clustering ensembles
    - *E.g.*, Consensus clustering
- Ensemble learning can be seen as a special kind of **meta-learning**
- In this lecture, we will focus mostly on supervised learning (classification) task

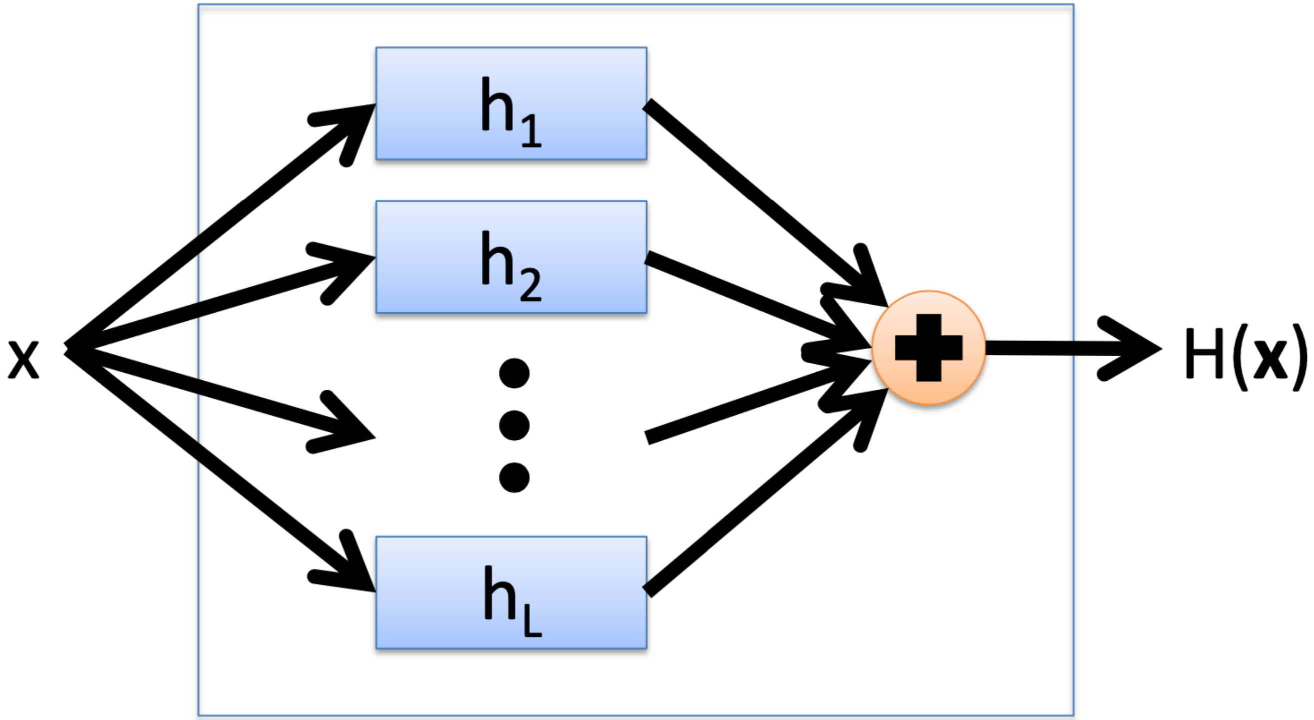
Strategies for combining classifiers

Combining classifiers

- Consider a set of classifiers $h_1, ..., h_L$ being **diverse**
  - Maybe from the same model, but making different mistakes (*e.g.*, trees in the random forest)
  - Maybe based on different models (*e.g.*, SVM + Naive Bayes)
- **Idea:** Construct a classifier $H(\mathbf{x})$ that combines the individual predictions of $h_1, ..., h_L$
  - $h_1, ..., h_L$ might return different predictions, or
  - $h_1, ..., h_L$ might focus on different regions of the representation space
- $H(\mathbf{x})$ is sometimes called a "meta-classifier"

Major voting (Averaging)

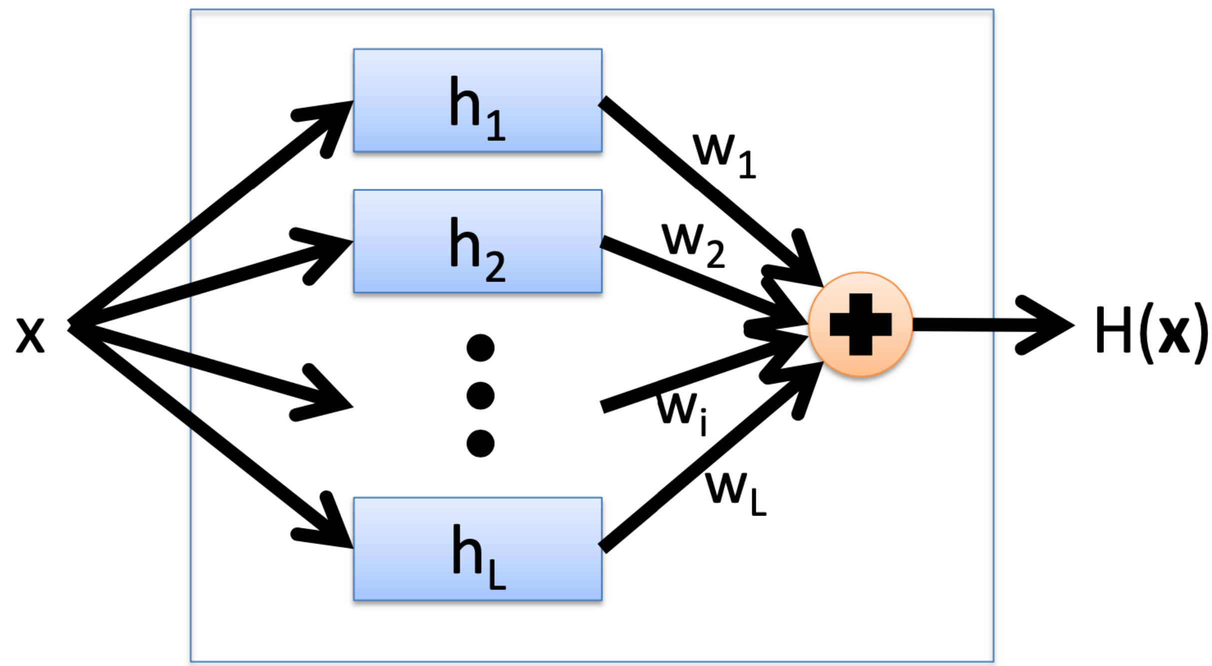
- *H*(***x***) is simply the majority vote
  - Or, the average output for regression
  - This is the most usual strategy for random forest



[https://courses.cs.washington.edu/courses/cse446/20wi/Lecture16/16_EnsembleMethods.pdf]

Weighted average

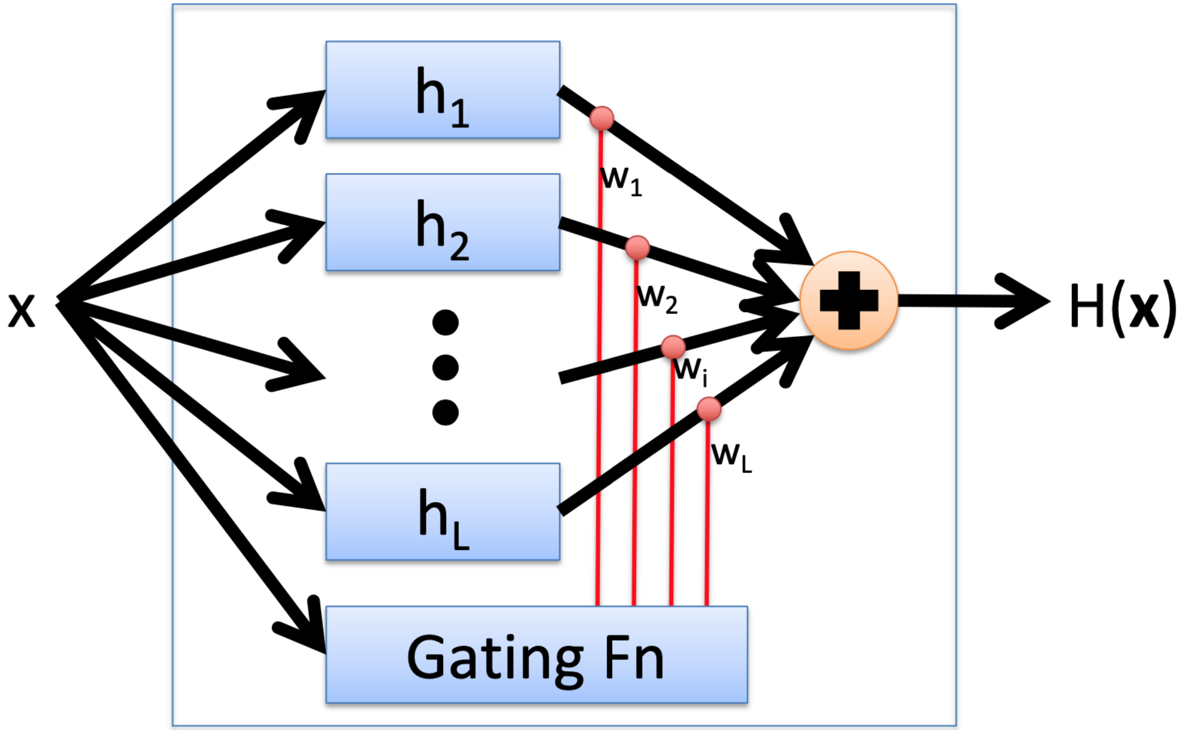
- *H*(***x***) is simply a weighted majority vote
  - Or, the weighted average output for regression
  - The weights $w_1, \ldots, w_L$ are learned from a validation set



[https://courses.cs.washington.edu/courses/cse446/20wi/Lecture16/16_EnsembleMethods.pdf]

Gating

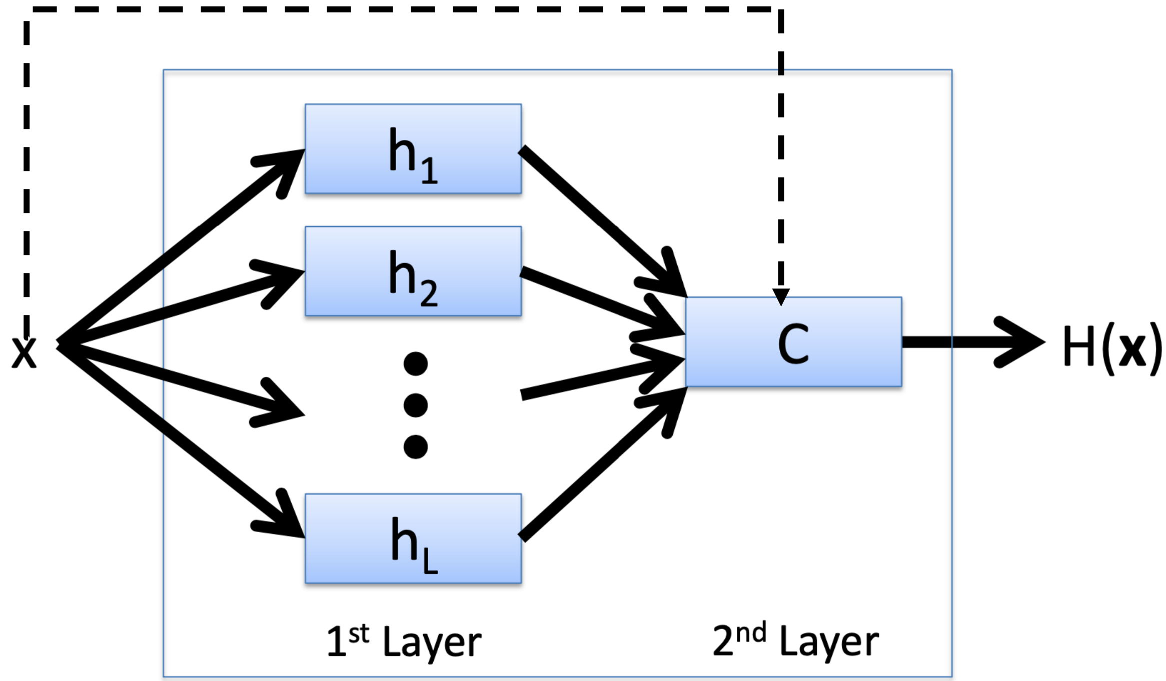
- This is a special case of weighted average where the weights $w_1, ..., w_L$ depend on the input(s)
  - The gating function selects/weights the best models for each problem
  - The gating function is learned from a validation set



[https://courses.cs.washington.edu/courses/cse446/20wi/Lecture16/16_EnsembleMethods.pdf]

Stacking

- Consists in stacking layers of classifiers
  - The predictions of the 1st layer are used as an input for the 2nd layer
  - The second layer is trained on a validation set
  - One might stack more than 2 layers of classifiers



[https://courses.cs.washington.edu/courses/cse446/20wi/Lecture16/16_EnsembleMethods.pdf]

Combining classifiers

- Consider a set of classifiers $h_1, \ldots, h_L$ being **diverse**
  - Maybe based on different models (*e.g.,* SVM + Naive Bayes)
    - In this case, the classifiers can be learned from the same training set (diversity is achieved by the diversity of the models)
  - Maybe from the same model, but making different mistakes (*e.g.,* trees in the random forest): The models can be
    - Learned from different **samplings** of the training set
      - Random strategies
      - *E.g.,* Bagging, Random Subspaces
    - **Stacked**
      - Adaptive strategies
      - *E.g.,* Boosting

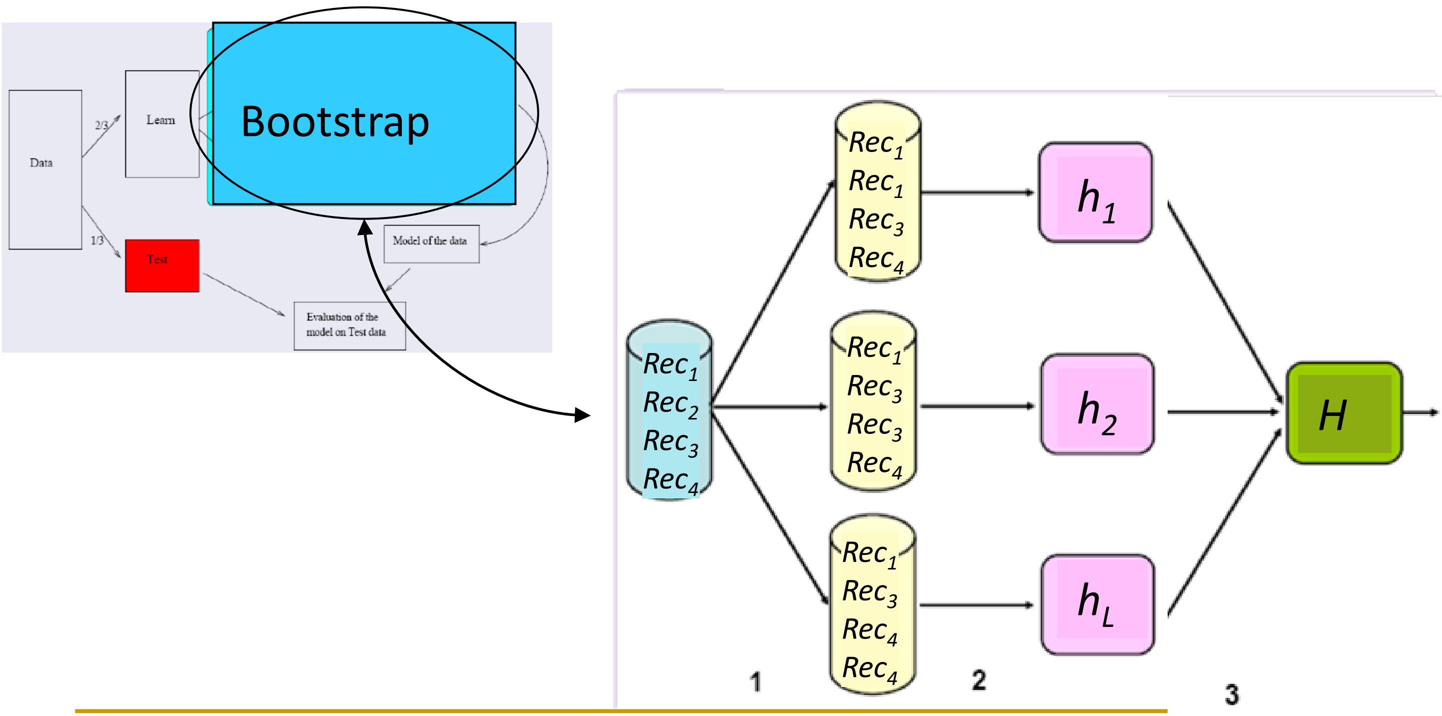
Ensemble learning methods based on data sampling

Bootstrap sampling

- A bootstrap sample is built from an initial dataset
  - By sampling *n* instances with replacement
  - Excludes ~35% of the instances
- A bootstrap sample contains as many objects as the original sample, but...
  - Some objects are replicated several times
  - Others are deleted
- The selection of the *n* objects to be replicated/deleted is ALMOST random
  - Bootstrap samples might be further **stratified** (mostly for classification)
    - Keeps the class distribution of the initial dataset

Bagging

- Bagging [Breiman, 1996]
  1. Creating L bootstrap samples
  2. Learning L classifers h_i
  3. Combining the L classifers h_1, ..., $h_L \rightarrow H$ (final classifier)



Bagging

- Objective of bagging:
  - Reduce the effects of outliers / records that are ‘’too influential’’
- The performance of the individual classifiers h_i is evaluated using out-of-bootstrap data
- For Step 3 (combination of classifiers), one can use:
  - Averaging
  - Weighted average
  - Gating

Bagging

- Advantages of Bagging:
  - Can reduce the detrimental effects of outliers
  - Can enhance performances
- Disadvantages of Bagging:
  - Higher computational complexity
    - E.g., We are essentially multiplying the work of growing a single tree by *L* (especially if we are using the more involved implementation that prunes and validates on the original training data) (https://www.stat.cmu.edu/~ryantibs/datamining/lectures/24-bag.pdf)
  - Loss of interpretability
    - E.g., The final bagged classifier is not a tree, and so we forfeit the clear interpretative ability of a classification tree (https://www.stat.cmu.edu/~ryantibs/datamining/lectures/24-bag.pdf)

Random subspaces

- Objectives of random subspaces [Ho, 1998] :
  - Reduce the effects of irrelevant attributes on the final model
  - Reduce the effects of attributes (features) correlation on the final model
  - Reduce the problems that arise when the number of examples in the training set is insufficient compared to the number of explanatory variables
- The L classifiers $h_1, ..., h_L$ are built from all examples
- Each classifier h_i is built from q attributes
  - Randomly selected from the whole set of explanatory attributes
  - In general, $q << p$ (where p is the initial number of explanatory attributes)

Random subspaces

- For the combination of classifiers, one can use:
  - Averaging
  - Weighted average
  - Gating

Random subspaces

- Advantages of Random subspaces:
  - Can effectively reduce the effects of irrelevant attributes on the final model
  - Can effectively reduce the effects of attributes (features) correlation on the final model
  - Can effectively reduce the problems that arise when the number of examples in the training set is insufficient compared to the number of explanatory variables
- Disadvantages of Random subspaces:
  - Higher computational complexity
  - Loss of readability/interpretability

Example of Ensemble learning based on data sampling:

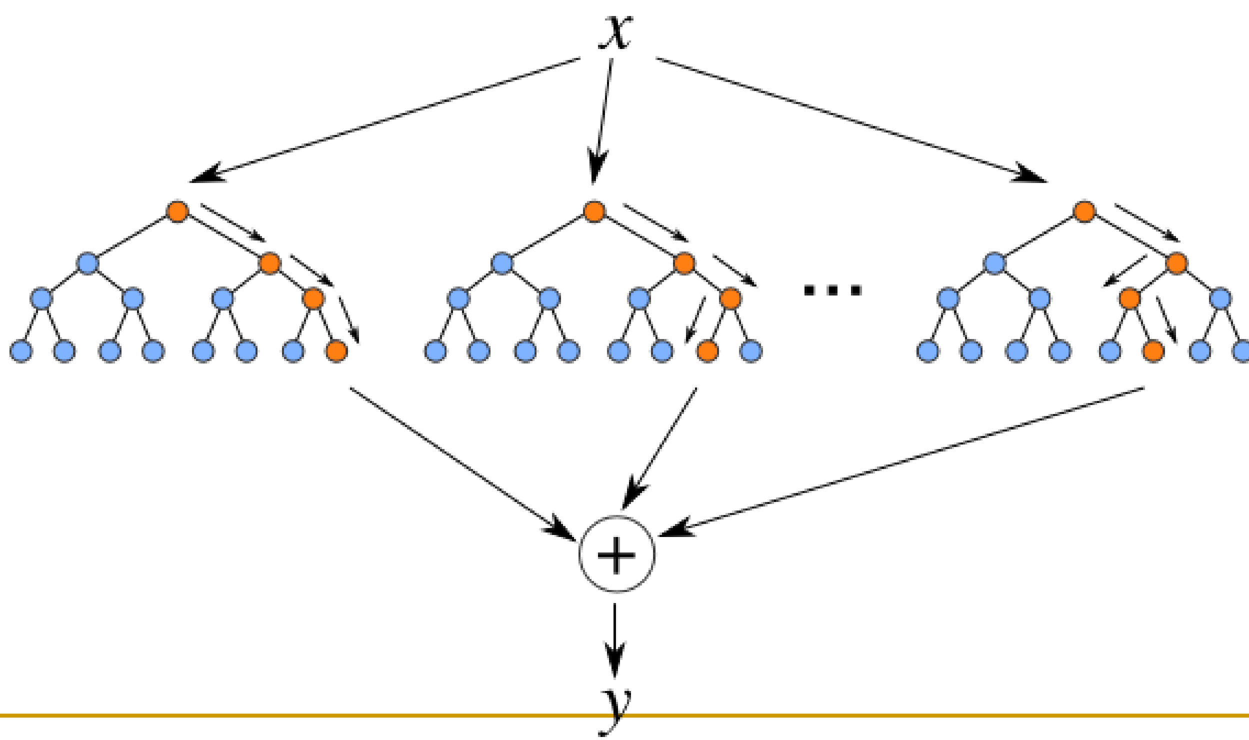
Random Forest

Random forest

- Random forest might be used for classification or regression
- Somehow combines Bagging and Random Subspaces: Each tree in the forest is built by:
  - Using Bootstrap replicas
  - Restrict the node decisions to a small subset of features picked randomly for each node (*node-level* Random Subspaces)

Random forest

- In its original version [Ho 95]:
  - The trees in the forest are not prunned
  - The outputs of the trees are averaged
  - Might be used for classification or regression



Random forest

- Advantages:
  - Better stability (i.e., effectively reduces variance)
  - On average, H (i.e., random forest) achieves better performance than any h_i (i.e., any individual tree)
- Disadvantages:
  - More computationally expensive
    - A lot more for the training phase
    - A bit more for the classification phase
  - Loose the readability/interpretability of the individual trees h_i

Ensemble learning based on classifiers stacking: Boosting

Motivation

- **Ensemble Learning models based on data sampling can effectively reduce the variance of the model**
- But, they might still suffer from a large bias
  - At least, in some regions of the representational space (*i.e.*, for some examples)
- **Boosting aims at reducing the bias**

Boosting

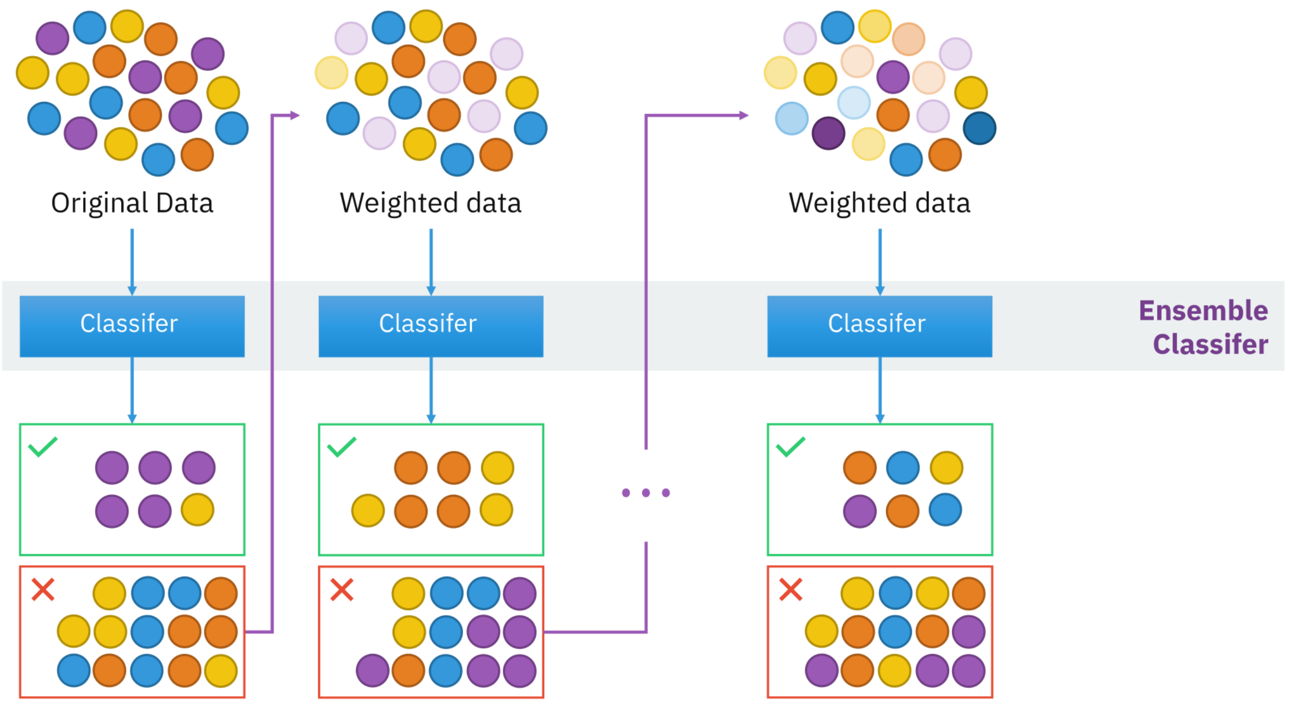
- Idea: sequentially derives weak classifiers (can be rules of thumb)
  - Apply procedure to subset of examples
  - Obtain the first weak classifier
  - Apply to 2nd subset of examples
  - Obtain 2nd weak classifier
  - Repeat *L* times
- How to choose examples on each round?
  - Concentrate on the “hardest” examples
    - The most often misclassified by previous classifiers

Boosting

- Technically:
  - Assume given "weak" classifiers h_t that perform slightly better than random (*i.e.,* accuracy ≥ 55% for 2-class problems)
  - We will stack them in a cascade, such that h_{t+1} focuses on samples that are misclassified by h_t (i.e., difficult examples)
- Given a sufficient number/variety of training examples, a Boosting algorithm can construct a meta-classifier H with much better accuracy

Boosting

- Principle: building a cascade of "weak" classifiers h_t …
- … Each individual classifier h_t aims at focusing on the "hard samples" that were incorrectly classified by its predecessor in the cascade h_{t-1}

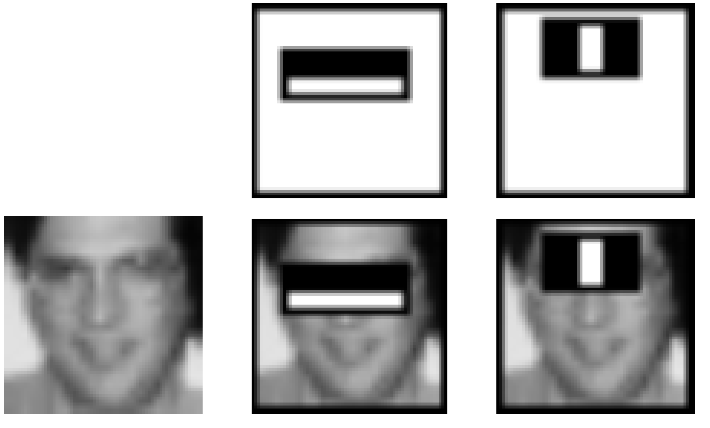


Boosting

- In practice, there are two main strategies:
  - **Arcing:** Increase the probability of selecting these "hard samples" in the training dataset of classifier h_t
  - **Ada-boost:** Increase the weight of these "hard samples" when learning the classifier h_t
- In general, the outputs of the classifiers are combined using (weighted) averaging

Boosting

- Example of practical application of Boosting
  - **Face detection** from numeric images [Viola&Jones, 2001]
    - Find faces in photograph or movie using Adaboost algorithm
    - Weak classifiers: detect light/dark rectangles in image
    - Many clever tricks to make it fast and accurate



Boosting

- Advantages
  - Reduces the effect of "difficult examples" (at the frontier between classes)
  - Often gives very good results in practice
    - If there is a large number of examples in the training set, Boosting often gives better results than Bagging
- Disadvantages
  - More computationally expensive for the individual classifiers
  - If there are not enough of training examples:
    - Either the training set is perfectly classified by the first classifier, and the Boosting algorithm is useless
    - Either the algorithm focuses on a small number of examples, potentially unrepresentative of the classes, and the "boosted" classifier may perform worse than the original classifier…