Ensemble Learning
Model selection
Statistical validation

Ensemble Learning

Definition

- Ensemble learning is a process that uses a set of models, each of them obtained by applying a learning process to a given problem. This set of models (ensemble) is integrated in some way to obtain the final prediction.
- Aggregation of multiple learned models with the goal of improving accuracy.
  - Intuition: simulate what we do when we combine an expert panel in a human decision-making process

Types of ensembles

- There are ensemble methods for:
  - Classification
  - Regression
  - Clustering (also known as consensual clustering)
- We will only discuss ensemble methods for supervised learning (classification and regression)
- Ensembles can also be classified as:
  - Homogeneous: It uses only one induction algorithm
  - Heterogeneous: It uses different induction algorithms

Types of ensembles

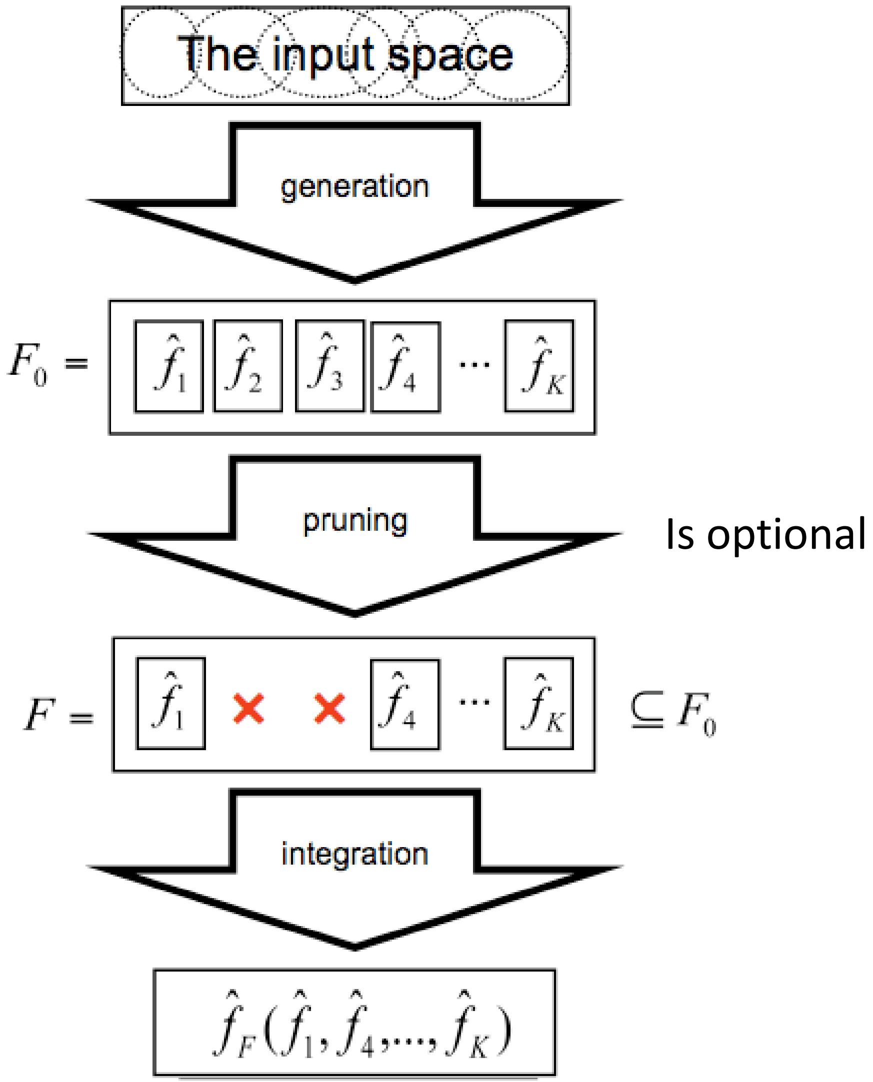
- There are ensemble methods for:
  - Classification
  - Regression
  - Clustering (also known as consensual clustering)
- We will only discuss ensemble methods for supervised learning (classification and regression)
- Ensembles can also be classified as:
  - Homogeneous: It uses only one induction algorithm
  - Heterogeneous: It uses different induction algorithms

Some Comments

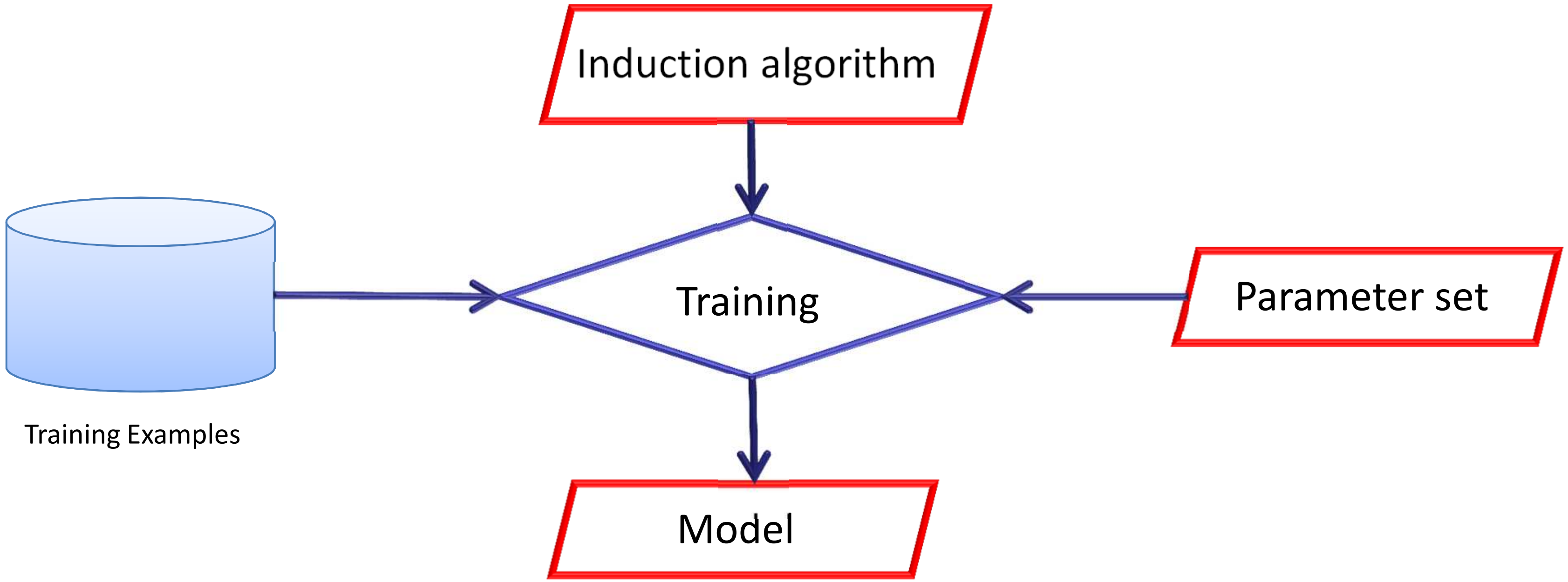
- **Combining models adds complexity**
  - It is, in general, more difficult to characterize and explain predictions
  - The accuracy may increase

- **Violation of Ockham's Razor**
  - "simplicity leads to greater accuracy"
  - Identifying the best model requires identifying the proper "model complexity"

The ensemble learning process



Methods to generate homogeneous ensembles

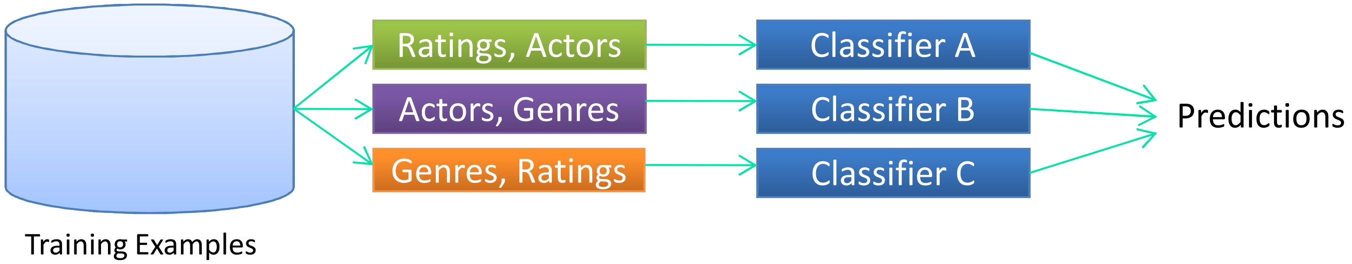


Data manipulation: it changes the training set in order to obtain different models

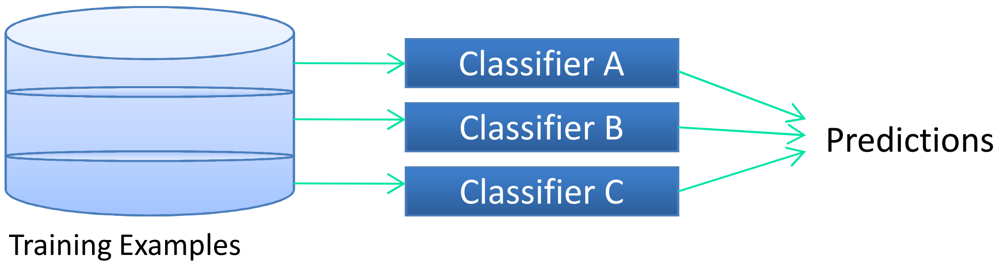
Modeling process manipulation: it changes the induction algorithm, the parameter set or the model (the last one is uncommon) in order to obtain different models

Data manipulation

Manipulating the input features

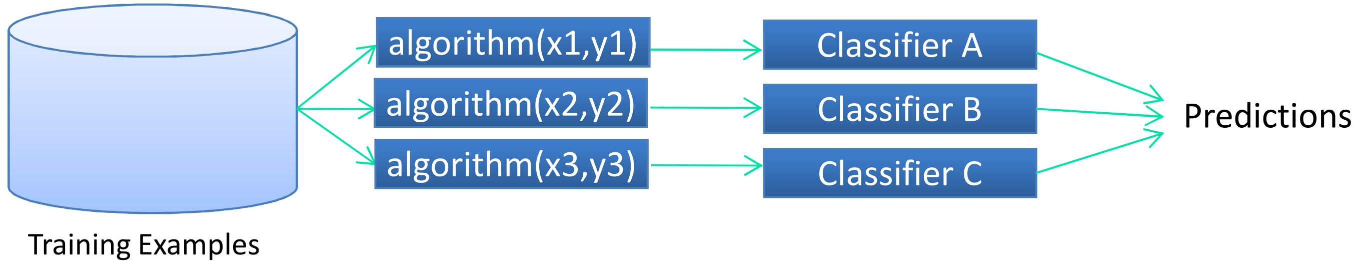


Sub-sampling from the training set

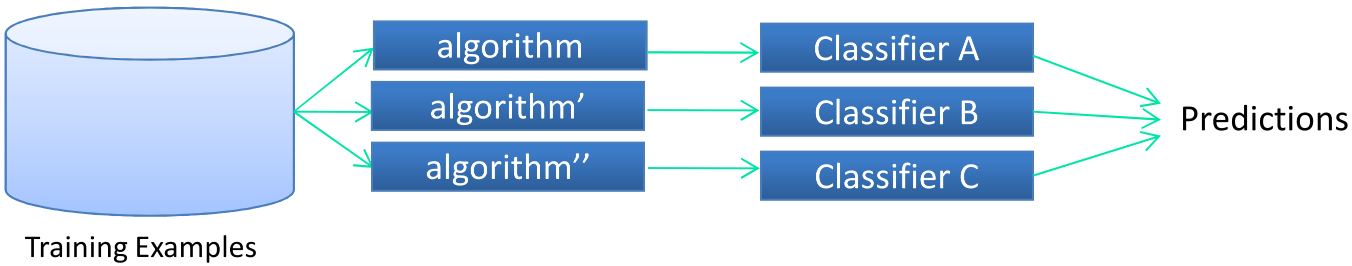


Modeling process manipulation

Manipulating the parameter sets



Manipulating the induction algorithm



where algorithm, algorithm’ and algorithm’’ are variations of the same induction algorithm

How to combine models (the integration phase)

- Algebraic methods
  - **Average**
  - **Weighted average**
  - Sum
  - Weighted sum
  - Product
  - Maximum
  - Minimum
  - Median

- Voting methods
  - **Majority voting**
  - **Weighted majority voting**
  - Borda count
    - *(rank candidates in order of preference)*

In bold: the most frequent

Characteristics of the base models

- For classification:
  - The base classifiers should be as accurate as possible and having diverse errors as much the true class is the majority class (see Brown, G. & Kuncheva, L., “Good” and “Bad” Diversity in Majority Vote Ensembles, *Multiple Classifier Systems, Springer,* **2010**, *5997*, 124-133)
  - It is not possible to obtain the optimum ensemble of classifiers based on the knowledge of the base learners

Characteristics of the base models

- For regression:
  - It is possible to express the error of the ensemble in function of the error of the base learners
  - Assuming the average as the combination method,

$$\boxed{E[(\hat{f}_{\mathcal{F}} - f)^2] \models \overline{bias}^2 + \frac{1}{K} \times \overline{var} + (1 - \frac{1}{K}) \times \overline{covar}}$$

  - The goal is to minimize $E[(\hat{f}_F - f)^2]$, so:
    - The average error of the base learners ($\overline{bias}$) should be as small as possible, i.e., the base learners should be as accurate (in average) as possible;
    - The average variance of the base learners ($\overline{var}$) should be as small as possible;
    - The average covariance of the base learners ($\overline{covar}$) should be as small as possible, i.e., the base learners should have negative correlation.

Popular ensemble methods

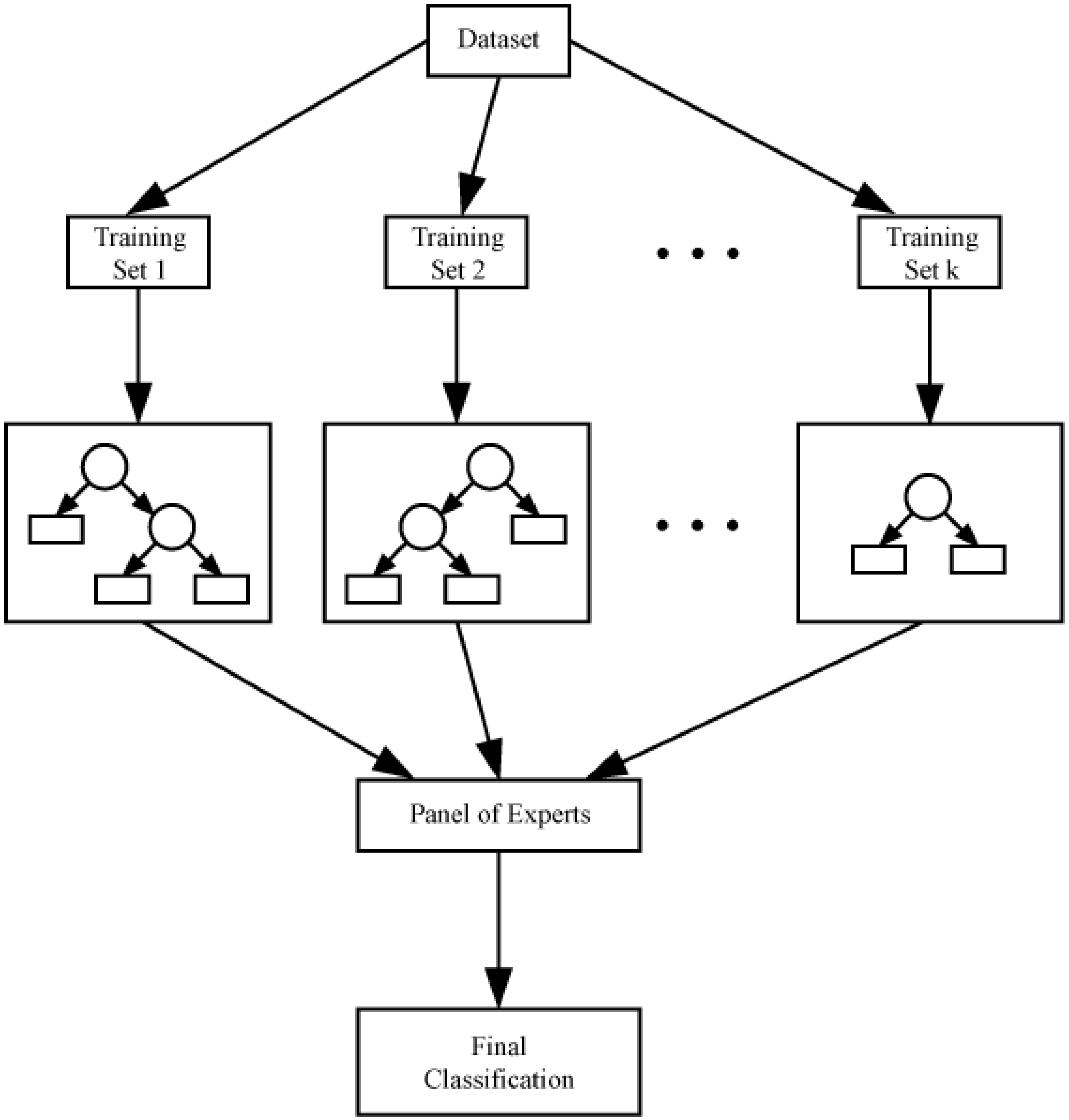
- Bagging:
  - averaging the prediction over a collection of unstable predictors generated from bootstrap samples (both classification and regression)
- Boosting:
  - weighted vote with a collection of classifiers that were trained sequentially from training sets given priority to instances wrongly classified (classification)
- Random Forest:
  - averaging the prediction over a collection of trees splited using a randomly selected subset of features (both classification and regression)
- Ensemble learning via negative correlation learning:
  - generating sequentially new predictors negatively correlated with the existing ones (regression)
- Heterogeneous ensembles:
  - combining a set of heterogeneous predictors (both classification and regression)

Bagging: Bootstrap AGGregatING

- **Analogy:** Diagnosis based on multiple doctors' majority vote
- **Training**
  - Given a set D of *d* tuples, at each iteration *i*, a training set D_i of *d* tuples is **sampled with replacement** from D (i.e., bootstrap)
  - A classifier model M_i is learned for each training set D_i
- **Classification**: classify an unknown sample **X**
  - Each classifier M_i returns its class prediction
  - The bagged classifier M* counts the votes and assigns the class with the most votes to **X**
- **Prediction:** can be applied to the prediction of continuous values by taking the average value of each prediction for a given test tuple

Bagging (Breiman 1996)

- Accuracy
  - Often significant better than a single classifier derived from D
  - For noise data: not considerably worse, more robust
  - Proved improved accuracy in prediction
- Requirement: Need unstable classifier types
  - Unstable means a small change to the training data may lead to major decision changes.
- Stability in Training
  - Training: construct classifier f from D
  - Stability: small changes on D results in small changes on f
  - Decision trees are a typical unstable classifier



http://en.wikibooks.org/wiki/File:DTE_Bagging.png

Boosting

- **Analogy:** Consult several doctors, based on a combination of weighted diagnoses—weight assigned based on the previous diagnosis accuracy
- Incrementally create models selectively using training examples based on some distribution.
- How boosting works?
  - Weights are assigned to each training example
  - A series of k classifiers is iteratively learned
  - After a classifier Mi is learned, the weights are updated to allow the subsequent classifier, Mi+1, to pay more attention to the training examples that were misclassified by Mi
  - The final M* combines the votes of each individual classifier, where the weight of each classifier's vote is a function of its accuracy

Boosting: Construct Weak Classifiers

- Using Different Data Distribution
  - Start with uniform weighting
  - During each step of learning
    - Increase weights of the examples which are not correctly learned by the weak learner
    - Decrease weights of the examples which are correctly learned by the weak learner
- Idea
  - Focus on difficult examples which are not correctly classified in the previous steps

Boosting: Combine Weak Classifiers

- Weighted Voting
  - Construct strong classifier by weighted voting of the weak classifiers
- Idea
  - Better weak classifier gets a larger weight
  - Iteratively add weak classifiers
    - Increase accuracy of the combined classifier through minimization of a cost function

Boosting

- Differences with Bagging:
  - Models are built sequentially on modified versions of the data
  - The predictions of the models are combined through a weighted sum/vote

- Boosting algorithm can be extended for numeric prediction
- Comparing with bagging: Boosting tends to achieve greater accuracy, but it also risks overfitting the model to misclassified data

AdaBoost: a popular boosting algorithm

(Freund and Schapire, 1996)

- Given a set of d class-labeled examples, (X1, y1), …, (Xd, yd)
- Initially, all the weights of examples are set the same (1/d)
- Generate k classifiers in k rounds. At round i,
  - Tuples from D are sampled (with replacement) to form a training set Di of the same size
  - Each example's chance of being selected is based on its weight
  - A classification model Mi is derived from Di and its error rate calculated using Di as a test set
  - If a tuple is misclassified, its weight is increased, otherwise it is decreased
- Error rate: err(Xj) is the misclassification error of example Xj. Classifier Mi error rate is the sum of the weights of the misclassified examples.

Adaboost comments

- This distribution update ensures that instances misclassified by the previous classifier are more likely to be included in the training data of the next classifier.
- Hence, consecutive classifiers' training data are geared towards increasingly hard-to-classify instances.
- Unlike bagging, AdaBoost uses a rather undemocratic voting scheme, called the *weighted majority voting. The idea is an intuitive one: those classifiers* that have shown good performance during training are rewarded with higher voting weights than the others.

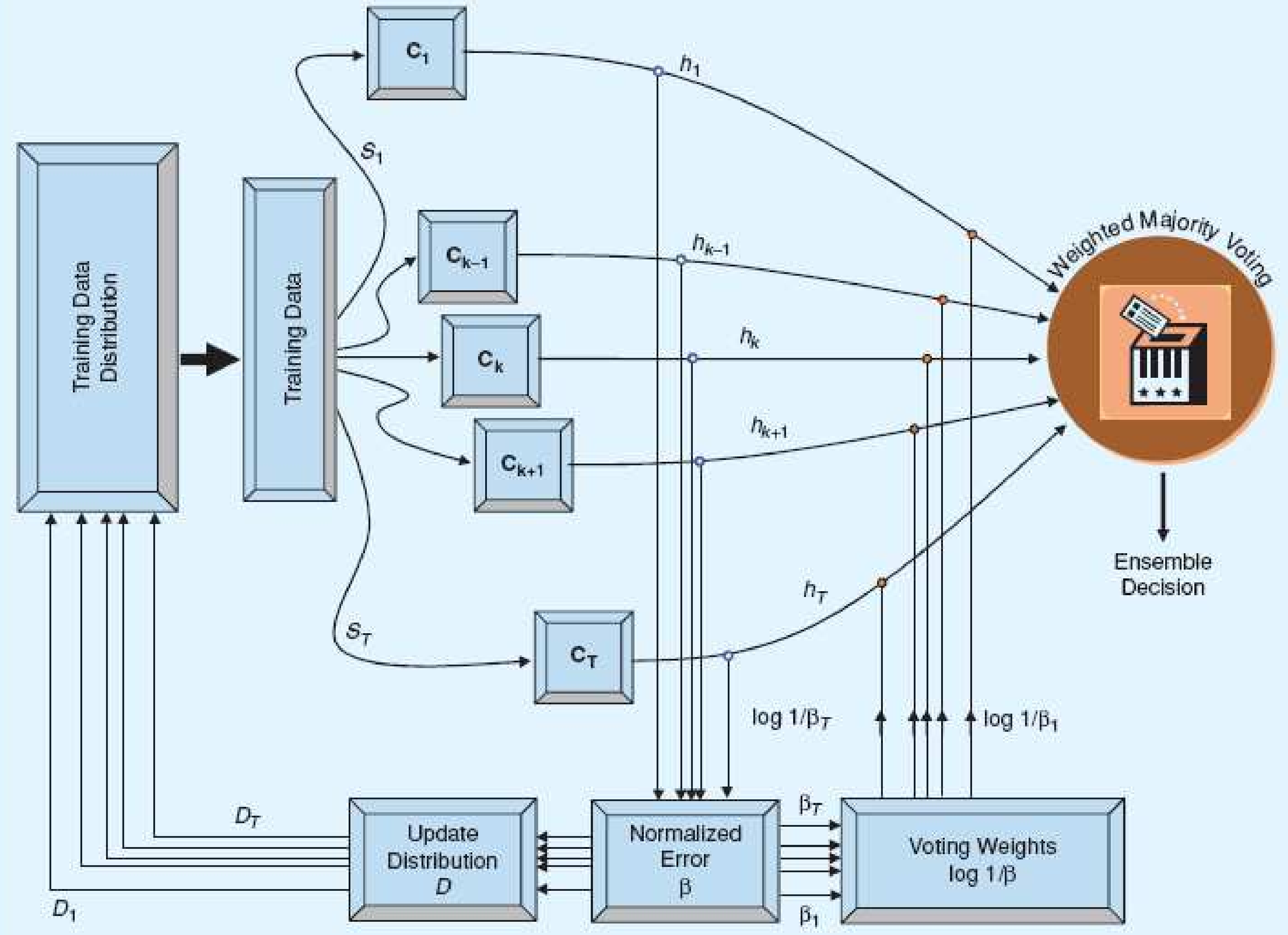


Figure 9. The AdaBoost.M1 algorithm.

The diagram should be interpreted with the understanding that the algorithm is sequential: classifier *CK is created before classifier CK+1, which in turn requires* that *βK and the current distribution DK be available.*

Random Forest (Breiman 2001)

- Random Forest: **A variation of the bagging algorithm**
- Created from individual decision trees.
- Diversity is guaranteed by selecting randomly at each split, a subset of the original features during the process of tree generation.
- **During classification**, each tree votes and the most popular class is returned
- **During regression**, the result is the averaged prediction of all generated trees

Random Forest (Breiman 2001)

- Two Methods to construct Random Forest:
  - Forest-RI (random input selection): Randomly select, at each node, F attributes as candidates for the split at the node. The CART methodology is used to grow the trees to maximum size
  - Forest-RC (random linear combinations): Creates new attributes (or features) that are a linear combination of the existing attributes (reduces the correlation between individual classifiers)
- Comparable in accuracy to Adaboost, but more robust to errors and outliers
- Insensitive to the number of attributes selected for consideration at each split, and faster than bagging or boosting

Ensemble learning via negative correlation learning

- Negative correlation learning can be used only in rnsemble regression algorithms that try to minimize/maximize a given objective function (e.g., neural networks, support vector regression)
- The idea is: a model should be trained in order to minimize the error function of the ensemble, i.e., it adds to the error function a penalty term with the averaged error of the models already trained.
- This approach will produce models negatively correlated with the averaged error of the previously generated models.

Model selection

Model selection

- Given a problem, which algorithms should we use?
- Golden rule: there is no algorithm that is the best one for any given problem
- Typically, two approaches (or both) can be adopted:
  - To choose the algorithm more suitable for the given problem
  - To adapt the given data for the intended algorithm (using pre-processing, for instance)
- Additionally, the concept of “good algorithm” depends on the problem:
  - For a doctor, the interpretation of the model can be a major criterion for the selection of the model (decision trees and Bayesian networks are very appreciated)
  - For logistics, the accuracy of travel time prediction is, typically, the most important selection criterion.

Model selection

- Hastie, T.; Tibshirani, R. & Friedman, J. H., The elements of statistical learning: data mining, inference, and prediction, *Springer,* 2001, pag. 313.

TABLE 10.1. *Some characteristics of different learning methods. Key:* 🟢 *= good,* 🟡 *=fair, and* 🔴 *=poor.*

Characteristic	Neural nets	SVM	Trees	MARS	k-NN, kernels
Natural handling of data of "mixed" type	🔴	🔴	🟢	🟢	🔴
Handling of missing values	🔴	🔴	🟢	🟢	🟢
Robustness to outliers in input space	🔴	🔴	🟢	🔴	🟢
Insensitive to monotone transformations of inputs	🔴	🔴	🟢	🔴	🔴
Computational scalability (large N)	🔴	🔴	🟢	🟢	🔴
Ability to deal with irrelevant inputs	🔴	🔴	🟢	🟢	🔴
Ability to extract linear combinations of features	🟢	🟢	🔴	🔴	🟡
Interpretability	🔴	🔴	🟡	🟢	🔴
Predictive power	🟢	🟢	🔴	🟡	🟢

Statistical validation

Statistical validation

- If model1 has an accuracy of 10 and model2 has an accuracy of 10.1, for a given test set, can we say that model1 is more accurate than model2?
- The answer is: we do not know. Remember that we are using a sample. The test set is a sample. How can we know whether these models would perform equally in a different test set?
- We should take into account with the variability of the results.
- We should validate statistically the results.
- Two recommended references:
  - Salzberg, S. L., On comparing classifiers: pitfalls to avoid and a recommended approach, *Data Mining and Knowledge Discovery,* **1997***, 1*, 317-327
  - Demsar, J., Statistical comparisons of classifiers over multiple data sets, *Journal of Machine Learning Research,* **2006***, 7*, 1-30

Introductory References

- *‘Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations’, Ian H. Witten and Eibe Frank, 1999*
- *‘Data Mining: Practical Machine Learning Tools and Techniques second edition’, Ian H. Witten and Eibe Frank, 2005*
- *Todd Holloway, 2008, “Ensemble Learning Better Predictions Through Diversity”, power point presentation*
- *Leandro M. Almeida, “Sistemas Baseados em Comitês de Classificadores”*
- *Cong Li, 2009, “Machine Learning Basics 3. Ensemble Learning”*
- R. Polikar, “Ensemble based systems in decision making,” *IEEE Circuits and Systems Magazine, vol. 6, no. 3, pp. 21–45, Quarter 2006.*

Top References

- Wolpert, D. H., Stacked generalization, *Neural Networks,* **1992***, 5,* 241-259
- Breiman, L., Bagging predictors, *Machine Learning,* **1996***, 26*, 123-140
- Freund, Y. & Schapire, R., Experiments with a new boosting algorithm, *International Conference on Machine Learning,* **1996**, 148-156
- Breiman, L., Random forests, *Machine Learning,* **2001***, 45*, 5-32
- Liu, Y. & Yao, X., Ensemble learning via negative correlation, *Neural Networks,* **1999***, 12*, 1399-1404
- Rodríguez, J. J.; Kuncheva, L. I. & Alonso, C. J., Rotation forest: a new classifier ensemble, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **2006***, 28*, 1619-1630