Ensemble Learning

Outline

- **Introduction**
- **Negative Correlation Learning**
- **Other Methods for Constructing an Ensemble Classifier**
  - **Bagging**
  - **Boosting & AdaBoost**
  - **Random Forest**
- **Summary**

Introduction

What Is an Ensemble?

- **An ensemble indicates a collection of individual learning machines, also called individual/base learners or, simply, individuals.**
- **For example, we could have an ensemble of neural networks (NNs), decision trees, support vector machines, polynomial regressors, etc.**
- **We could even have heterogeneous base learners in an ensemble.**
- **There are many related topics to ensembles, e.g., committee machines, multiple classifiers, mixture of experts, stacked generalisation, etc.**

[Example] Regression Problems

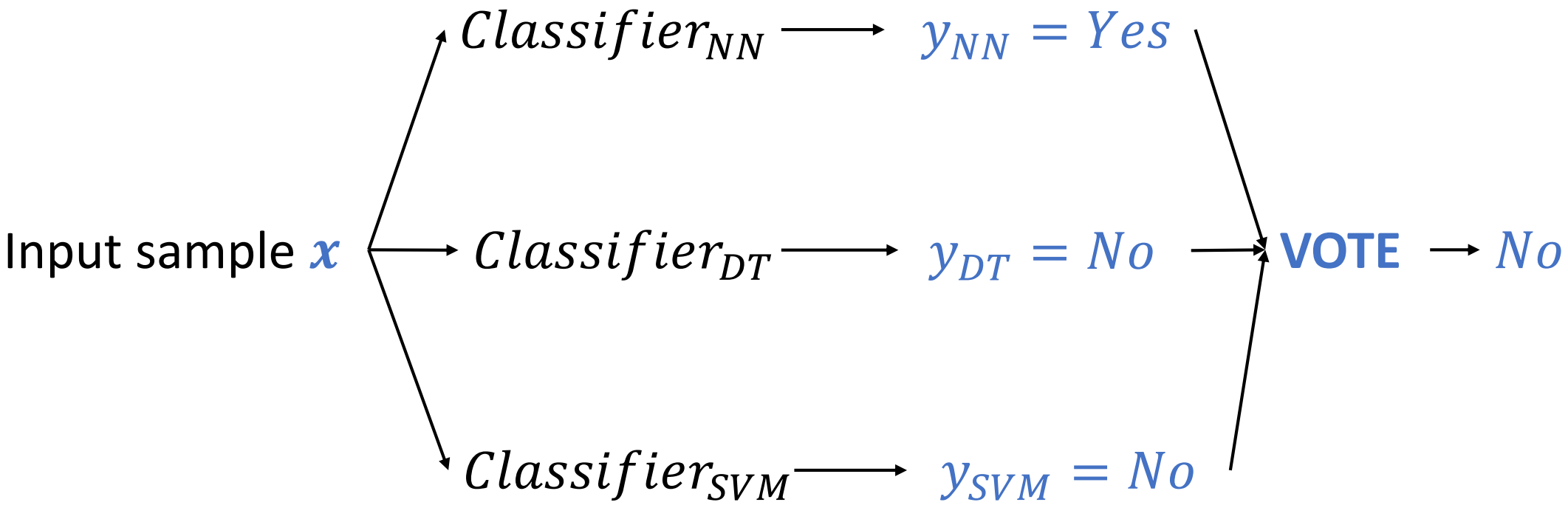
- **Given K base learners, whose outputs are $o_j, \quad j = 1, 2, \ldots, K$, a simple ensemble output could be**

$$O = \sum_{j=1}^{K} w_j o_j,$$

where w_j are the weights.

[Example] Classification Problems

Waiting at a Restaurant: Simple (unweighted) majority voting is used in this example.



Why Ensembles?

- **For a large and complex problem, designing a monolithic system to solve it is often very difficult.**
- **Divide-and-conquer is a common strategy in solving such problems.**
- **Ensemble approaches could be viewed as an automatic approach toward divide-and-conquer.**

- ***X. Yao and Y. Liu, ``Making use of population information in evolutionary artificial neural networks,'' IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics, 28(3):417-425, June 1998.***

A Simple Illustrative Example

- **Consider an ensemble of K base learners and we combine their predictions using simple majority voting. We misclassify a new example x if at least half of the base learners misclassify x, i.e.,**

$$K_{bad} \geq floor\left(\frac{K}{2}\right) + 1.$$

- **[Question 1] Is the ensemble much less likely to misclassify x than a single learner?**
- **[Question 2] Does an ensemble always perform better than a single learner?**

A Simple Illustration Example

- **[Question 1] Is the ensemble much less likely to misclassify x than using a single learner?**
- **[Possible Answer]**
  - **Suppose each $f \in \mathcal{F}$ misclassifies a randomly chosen example with probability ε (called error) and the errors of these K learners are independent of each other.**
  - **Consider the case that there are 2 classes, then the error that the ensemble misclassifies an example is:**

$$\varepsilon_{ens} = \sum_{i=floor\left(\frac{K}{2}\right)+1}^{K} \binom{K}{i} \varepsilon^{i}(1-\varepsilon)^{K-i}$$

A Simple Illustration Example

- **[Question 1] Is the ensemble much less likely to misclassify x than using a single learner?**
- **[Possible Answer]**
  - **Suppose each $f \in \mathcal{F}$ misclassifies a randomly chosen example with probability ε (called error) and the errors of these K learners are independent of each other.**
  - **Then if ε is small, the probability of misclassification with a large K is miniscule.**

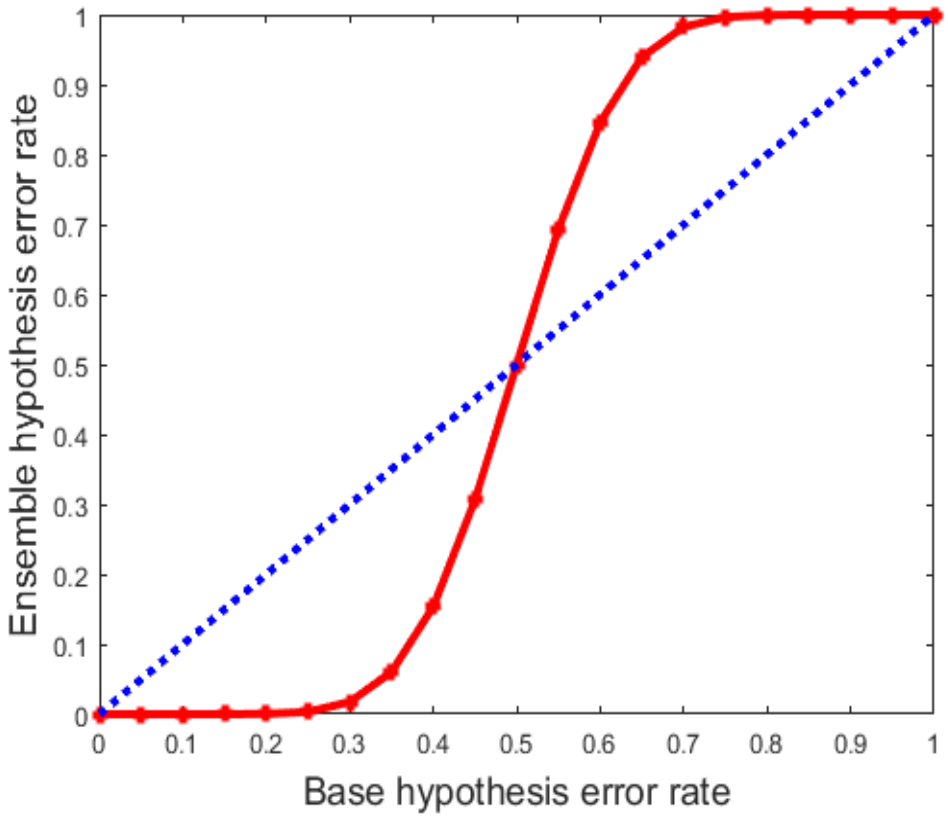
K\Error	ε=0.1	ε=0.2	ε=0.4
5	0.008560	0.057920	0.317440
10	0.000147	0.006369	0.166239
20	0.000001	0.000563	0.127521

Table: Error of misclassification using majority voting, examples with 2 classes.

A Simple Illustration Example

- **[Question 2] Does an ensemble always perform better than a single learner?**
- **[Answer] Illustrated by the figure (an ensemble of 2 base learners):**

  Error rate ε should be below 0.5!
- **[Remark] An ensemble of identical base learners = one base learner**



Identical base hypotheses.
Independent base hypotheses.

When Is an Ensemble Better?

1. **The errors of base learners should be independent of each other.**

2. **The base learners should do better than random guessing (i.e., with $\varepsilon < 0.5$).**

Unreasonable!

Independence Assumption

- **True independence of errors from different base learners is hard to achieve because of, e.g.,**
  - **Same set of training data.**
  - **Similar training bias.**
  - **Similar algorithms.**
  - **Etc.**
- **[Possible Solutions]**
  - **Use different learners to reduce the positive correlation between their errors.**
    - **Different supervised learning methods.**
    - **Different parameters and weights.**
    - **Different base learners.**
    - **Etc.**

Can We Do Better Than Independence?

- **It turns out that we can.**
- **Instead of pursuing mutual independence of errors of base learners, we can go one step further and encourage negative correlation of errors of base learners.**
- **How?**

Negative Correlation Learning

Negative Correlation Learning (NCL) [3]

- **Suppose we have a training set**

$$\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\},$$

where x and y are input features and output.

- **Consider estimating the output by forming an ensemble whose output $F(x)$ is a simple averaging of outputs $F_i(x)$ of K neural networks (could be other base learners as well):**

$$F(x) = \frac{1}{K}\sum_{i=1}^{K} F_i(x) \qquad \textbf{(1)}$$

Negative Correlation Learning (NCL) [3]

- **Negative correlation learning defines a simple error function for each network i as follows (N is the size of training set):**

$$\varepsilon_i = \frac{1}{N}\sum_{n=1}^{N}\left(\frac{1}{2}(F_i(x_n) - y_n)^2 + \lambda p_i(x_n)\right) \quad \textbf{(2)}$$

where $p_i(x_n) = (F_i(x_n) - F(x_n))\sum_{j\neq i}(F_j(x_n) - F(x_n))$.

- $\frac{1}{N}\sum_{n=1}^{N}\left(\frac{1}{2}(F_i(x_n) - y_n)^2\right)$: **empirical risk function.**
- $\frac{1}{N}\sum_{n=1}^{N} p_i(x_n)$: **correlation penalty function.**
- $0 \leq \lambda \leq 1$: **adjust the strength of the penalty.**

What Does $\boldsymbol{p_i}$ Do?

$$\boldsymbol{p_i(x_n) = (F_i(x_n) - F(x_n)) \sum_{j \neq i} (F_j(x_n) - F(x_n))}$$

- **Negatively correlate each NN's error with errors for the rest of the ensemble.**

- **The partial derivative of $\boldsymbol{\varepsilon_i}$ with respect to the output of network $\boldsymbol{i}$ on the $\boldsymbol{n^{th}}$ training example is:**

$$\begin{aligned} \frac{\partial \varepsilon_i(\boldsymbol{x_n})}{\partial F_i(\boldsymbol{x_n})} &= F_i(\boldsymbol{x_n}) - y_n + \lambda \frac{\partial p_i(\boldsymbol{x_n})}{\partial F_i(\boldsymbol{x_n})} \\ &= F_i(\boldsymbol{x_n}) - y_n - \lambda \textstyle\sum_{j \neq i} \big(F_j(\boldsymbol{x_n}) - F(\boldsymbol{x_n})\big) \\ &= F_i(\boldsymbol{x_n}) - y_n - \lambda \big(F_i(\boldsymbol{x_n}) - F(\boldsymbol{x_n})\big) \\ &= (1-\lambda)(F_i(\boldsymbol{x_n}) - y_n) + \lambda(F(\boldsymbol{x_n}) - y_n) \end{aligned} \quad \textbf{(3)}$$

Observations

- **During training, all base NNs interact with each other through their penalty terms in the error function. NCL considers errors that all other NNs have made while training an NN.**

- **Case $\lambda = 0$: Independent training of individual NNs is a special case of NCL.**

- **Case $\lambda = 1$:** $\frac{\partial \varepsilon_i(x_n)}{\partial F_i(x_n)} \propto \frac{\partial \varepsilon_{ens}(x_n)}{\partial F_i(x_n)}$.
  - **The minimization of the empirical risk function of the ensemble is achieved by minimizing the error functions of individual NNs.**

Example: Australian Credit Card Assessment [3]

- **Assess (classify) credit card applications.**
  - **14 attributes:**
    - **6 numeric/continuous;**
    - **8 categorical/discrete.**
  - **690 instances with missing values.**
  - **Two classes:**
    - **307 belong to one class;**
    - **383 belong to another.**

Experimental setting [3]

- **Training set: the first 518 examples.**
- **Test set: the remaining 172 examples.**
- **Input attributes: rescaled to between 0.0 and 1.0 by a linear function.**
- **Output attributes of all the problems:**
  - **Encoded using a 1-of-*m* output representation for m classes.**
  - **The output with the highest activation designated the class.**
- **An ensemble of 4 networks with one hidden layer.**

Comparison Among Learning Algorithms [3]

Comparison among negative correlation learning (NCL), EPNet (Yao & Liu, 1997), an evolutionary ensemble learning algorithm (Evo-En-RLS) (Yao & Liu, 1998), and others (Michie, Spiegelhalter & Taylor, 1994) in terms of the average testing error rate for the Australian credit card assessment problem. TER stands for Testing Error Rate in the table

Algorithm	TER	Algorithm	TER
NCL	0.120	DIPOL92	0.141
EPNet	0.115	Discrim	0.141
Evo-En-RLS	0.095	Logdisc	0.141
Cal5	0.131	Cart	0.145
Itrule	0.137	RBF	0.145
Castle	0.148	NaiveBay	0.151
IndCART	0.152	BP	0.154

NCL: Summary

- **Instead of creating an ensemble of unbiased individual networks whose errors are uncorrelated, NCL can produce individual networks whose errors are negatively correlated.**

- **However, a parameter λ is introduced.**

Other Methods for Constructing an Ensemble Classifier

- Bagging
- Boosting & AdaBoost
- Random Forest

Bootstrap **Agg**regat**ing** (acronym ***Bagging***)

- **Pseudo-code**

1: **Input** the original data set $\mathscr{D}$
2: **Input** the number of bootstrap samples k
3: **for** $i = 1$ to k **do**
4: Create a bootstrap sample $\mathscr{D}_i$ of size N from $\mathscr{D}$ according to uniform probability distribution
5: Train a base classifier C_i on the bootstrap $\mathscr{D}_i$
6: **end for**
7: Train models $C_1, C_2, \ldots, C_k$ using $\mathscr{D}_1, \mathscr{D}_2, \ldots, \mathscr{D}_k$, respectively
8: Use a *voting model* or *averaged model* C composed by $C_1, C_2, \ldots, C_k$ as final ensemble model

- **Remarks**:
  - **Repeatedly sample from a data set, *with replacement*, according to a uniform probability distribution (unbiased sampling).**
  - **size(sampled data set) = size(original data set)**
  - **Data set C_i contains approximately 63% of the original data if N is sufficiently large. [2]**

Bagging: From the View of *Errors*

- **Improves the generalization error by reducing the variance of the base classifiers.**

- **A critical factor of accuracy: the stability of the base classifiers.**

  "Stable" = being robust to minor perturbations in the training set

  - **If a base classifier is *unstable***

    ⇒Bagging helps to reduce the errors associated with the noise in the training set

  - **If a base classifier is *stable***

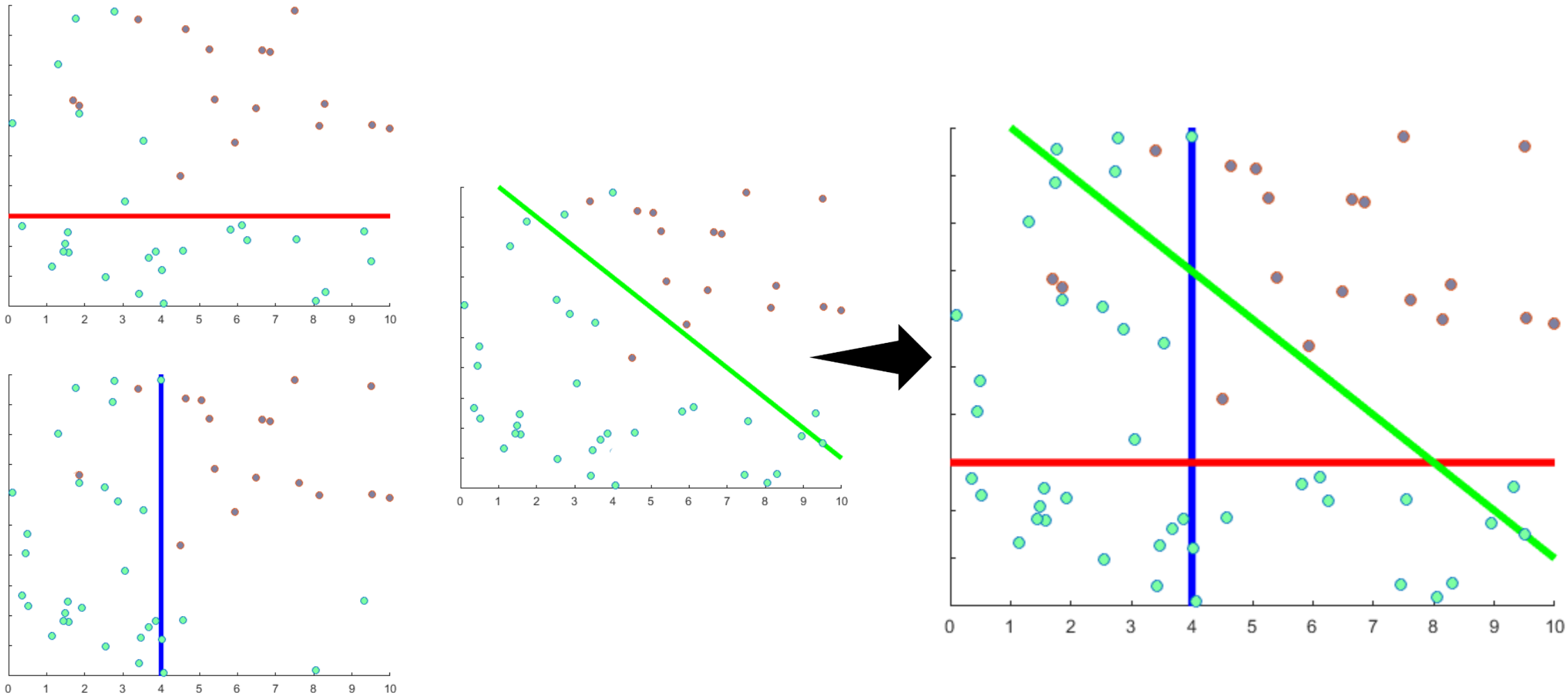
    ⇒The error of the ensemble is mainly due to the bias in the base classifier.

    ⇒Not easy to tackle this by Bagging.

    ⇒[Question] It may degrade the classifier's performance, why?

    [Answer] because the effective size of each training set is about 37% smaller than the original data.

Bagging: Illustration Example



Bagging: A Classification Example [4]

Dataset	#Samples	#Variables	#Classes	#Test set	$\bar{\varepsilon}$ (%)	$\overline{\varepsilon_E}$ (%)	Decrease
Waveform	300	21	3	1500	29.0	19.4	**33%**
Heart	1395	16	2	250	10.0	5.3	**47%**
Breast cancer	699	9	2	100	6.0	4.2	**30%**
Ionosphere	351	34	2	25	11.2	8.6	**23%**
Diabetes	1036	8	2	250	23.4	18.8	**20%**
Glass	214	9	6	20	32.0	24.9	**22%**
Soybean	307	35	19	25	14.5	10.6	**27%**

- **Ensemble consist of classification trees using 10-fold cross-validation.**
- **Tested on 7 datasets. For each dataset:**
  - **Uniformly random division of each data set into learning and test sets.**
  - **50 bootstrap samples from the training set.**
  - **100 trials each.**

Why Bagging works?

- Consider a **regression** problem and the following notations:
  - $(\boldsymbol{x}, y)$: an example independently drawn from training set $\mathcal{D}$ according to the probability distribution $\mathcal{P}$.
  - $h(\boldsymbol{x}, \mathcal{D})$: the predictor.
- Then the aggregated predictor is: $h_{ens}(\boldsymbol{x}, \mathcal{P}) = E_{\mathcal{D}}[h(\boldsymbol{x}, \mathcal{D})]$
- Take $\boldsymbol{X}$, Y as random variables having the distribution $\mathcal{P}$ and independent of $\mathcal{D}$. The average prediction error ε of $h(\boldsymbol{x}, \mathcal{D})$ is

$$\varepsilon = E_{\mathcal{D}} E_{Y,\boldsymbol{X}}\left(Y - h(\boldsymbol{X}, \mathcal{D})\right)^2.$$

- Define the error of the aggregated predictor h_{ens} to be

$$\varepsilon_{ens} = E_{Y,\boldsymbol{X}}\left(Y - h_{ens}(\boldsymbol{X}, \mathcal{P})\right)^2.$$

Instability helps!

- Using $(EZ)^2 \leq EZ^2$ gives:

$$\varepsilon = EY^2 - 2EYh_{ens} + E_{Y,\boldsymbol{X}}E_{\mathcal{D}}h^2(\boldsymbol{X},\mathcal{D})$$
$$\geq E(Y - h_{ens})^2 = \varepsilon_{ens}$$

  - ε_{ens} is lower than ε.
  - How much lower (gap) depends on how unequal the two sides of $[E_{\mathcal{D}}h(\boldsymbol{x},\mathcal{D})]^2 \leq E_{\mathcal{D}}h^2(\boldsymbol{x},\mathcal{D})$ are.
  - The effect of instability is clear.
    - If $h(\boldsymbol{x},\mathcal{D})$ does not change too much with replicate $\mathcal{D}$, the two sides will be nearly equal, and aggregation will not help.
    - The more highly variable the $h(\boldsymbol{x},\mathcal{D})$ are, the more improvement aggregation may produce. But h_{ens} always improves on h.

Other Methods for Constructing an Ensemble Classifier

- Bagging
- Boosting & AdaBoost
- Random Forest

Boosting

- Idea: Combine several weak classifiers to create a stronger one.
- An iterative procedure used to adaptively change the distribution of training samples.
- Thus, instead of sampling uniformly at random, a weight is assigned to each training example -> to be changed adaptively to emphasize more difficult examples.

Boosting: Pseudo-Code

- **Pseudo-code**

1: **Input** the original data set $\mathscr{D}$
2: **Input** the number of bootstrap samples k
3: Number of training samples $N = |\mathscr{D}|$
4: Initialise the weights for samples $\mathbf{w} \leftarrow (\frac{1}{N}, \frac{1}{N}, \ldots, \frac{1}{N})$
5: **for** $i = 1$ to k **do**
6: Create a bootstrap sample $\mathscr{D}_i$ of size N from $\mathscr{D}$ according to $\mathbf{w}$
7: Train a base model on $\mathscr{D}_i$
8: Increase the weights of incorrectly classified examples
9: Reduce the weights of correctly classified examples
10: Normalise $\mathbf{w}$
11: **end for**
12: Aggregating the trained base classifiers and use it as final ensemble model

Boosting: Two Core Questions

- Two core components:
  - **Weights**:
    - Used as a sampling distribution when creating bootstraps.
    - Used by the base classifier to learn a model which is **biased** towards the examples that are hard to classify (ones with higher weights).
  - **Final ensemble** = weighted-majority/weighted-average combination

- [Two Questions]
  1. How the weights are updated at each iteration?
  2. How to combine the trained models to create the final ensemble?

Variants differ in the above 2 points! For instance: AdaBoost.

AdaBoost

function ADABOOST($examples$, L, K) **returns** a weighted-majority hypothesis
  inputs: $examples$, set of N labeled examples $(x_1, y_1), \ldots, (x_N, y_N)$
    L, a learning algorithm
    K, the number of hypotheses in the ensemble
  local variables: **w**, a vector of N example weights, initially $1/N$
    h, a vector of K hypotheses
    z, a vector of K hypothesis weights

  for $k = 1$ **to** K **do**
    $\mathbf{h}[k] \leftarrow L(examples, \mathbf{w})$
    $error \leftarrow 0$
    for $j = 1$ **to** N **do**
      if $\mathbf{h}[k](x_j) \neq y_j$ **then** $error \leftarrow error + \mathbf{w}[j]$
    for $j = 1$ **to** N **do**
      if $\mathbf{h}[k](x_j) = y_j$ **then** $\mathbf{w}[j] \leftarrow \mathbf{w}[j] \cdot error/(1 - error)$
    $\mathbf{w} \leftarrow$ NORMALIZE(**w**)
    $\mathbf{z}[k] \leftarrow \log(1 - error)/error$
  return WEIGHTED-MAJORITY(**h**, **z**)

Image source: Figure 18.34 of [1].

Property of AdaBoost:
If the input base classifier is a weak learner (i.e., prediction accuracy slightly better than guessing randomly), then *AdaBoost* will return a hypothesis that classifies the training data perfectly for K large enough.

Other Methods for Constructing an Ensemble Classifier

- Bagging
- Boosting & AdaBoost
- Random Forest

Random Forest

- **An ensemble consisting of multiple DTs.**
- **Each tree is generated based on the values of an independent set of random vectors.**
- **The random vectors are generated from a fixed probability distribution, unlike the adaptive approach used in AdaBoost, where the probability distribution is varied to focus on examples that are hard to classify.**
- **Bagging using DTs is a special case of random forest.**

Random Forest

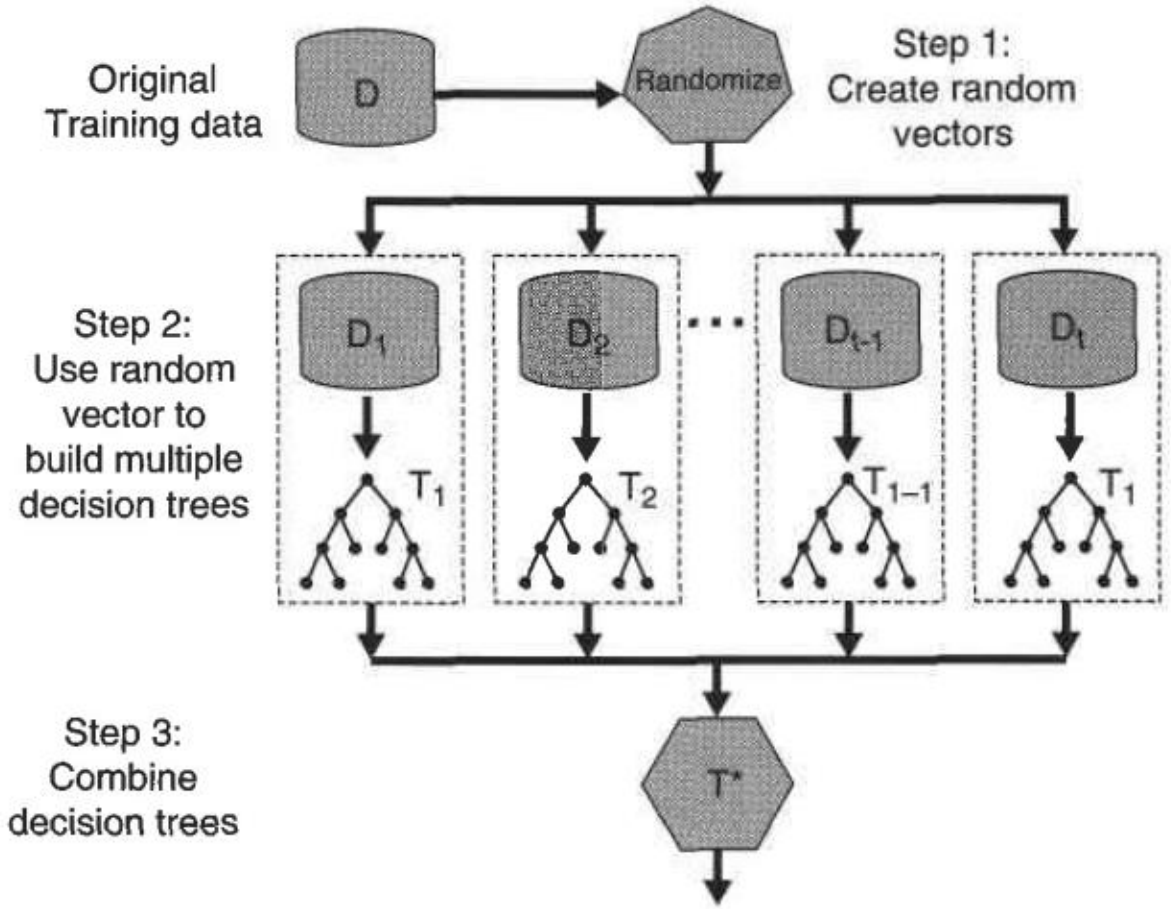


Image source: Figure 5.40 of [2].

Random Forest: From the View of *Error*

- It has been proven that, when *the number of trees is sufficiently large*:

$$Generalization\_Error \leq \frac{\bar{\rho}(1-s^2)}{s^2}$$

  - $\bar{\rho}$ denotes the average correlation among the DTs,
  - s is a quantity that measures the strength of the DT classifiers.

- "**Strength**": average performance of the t classifiers

$$s = \frac{1}{t}\sum_{i=1}^{t} M_i(\boldsymbol{X}, Y)$$

- "**Performance**": the margin of classifier i

$$M_i(\boldsymbol{X}, Y) = P\left(\hat{Y}_\theta = Y\right) - \max_{Z \neq Y} P(\hat{Y}_\theta = Z)$$

  - $\hat{Y}_\theta$ is the predicted class of $\boldsymbol{X}$ according to a classifier built from some random vector θ.

Random Forest: Randomisation helps!

- [Question] What can you observe from the following bound?

$$Generalization\_Error \leq \frac{\bar{\rho}(1-s^2)}{s^2}$$

  - $\bar{\rho}$ denotes the average correlation among the DTs,
  - s is a quantity that measures the strength of the DT classifiers.
- [Observation]
  - **If the DTs are more correlated, the upper bound of the generalization error increases.**
    - ***Randomisation helps to reduce the correlation!***
  - **If the strength of the ensemble decreases, the upper bound of the generalization error increases.**

Input Feature Selection

- **Forest-RI:**
  1. **Randomly select F input features to split at each node of the DT.**
  2. **The decision to split a node is determined from the F features.**
  3. **Grow the entire tree *without any pruning*.**
  4. **Combine the predictions using a *majority voting scheme*.**
  - **Trade-off between the strength and correlation, depending on F:**
    - **F small => less correlated trees.**
    - **F large => strength increases.**
    - **Suggested $F = \log_2 d + 1$ where d is the #features of original data.**

- **Forest-RC: when d is low, increase the feature space using linear combination of features.**

Discussions

Current Status

- **For regression problems, it has been shown that an ensemble performs no worse than any of its individual learners under some mild assumptions/conditions.**
- **For classification problems, there are also ample empirical evidences to show ensemble's advantages over single individuals, although a rigorous proof is still yet to be found.**
- **What is it in an ensemble that makes it better?**

Diversity!

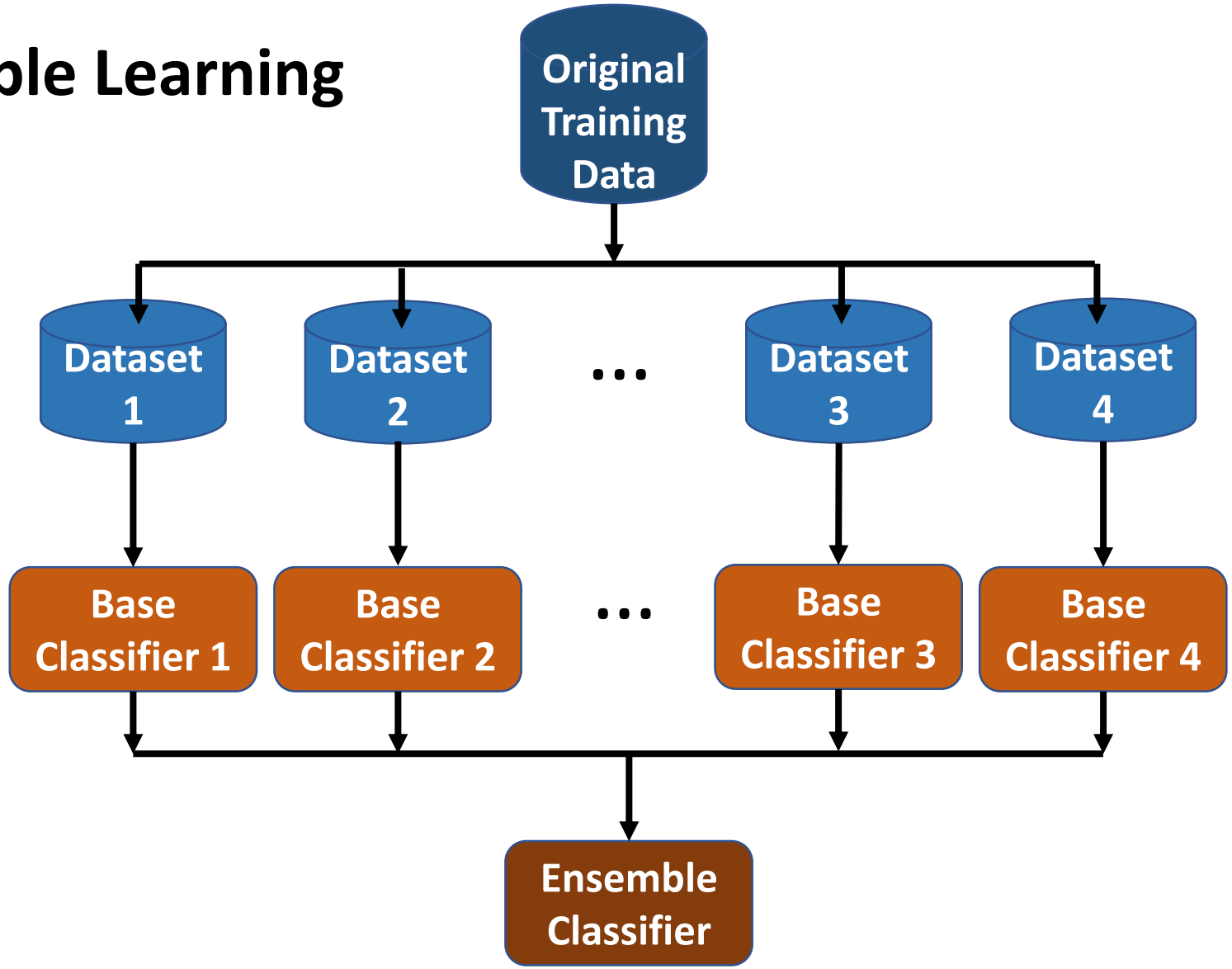
- **An ensemble of positively correlated individuals provide few advantages over single individuals.**
- **There have been many studies demonstrating that a diverse ensemble provide better generalization.**
- **What do you mean by “diverse”? How do you define diversity? How do you generate diversity? …**
- **Still ongoing research as to how diversity can be best defined and used in practice.**

- *G. Brown, J. L. Wyatt, R. Harris and X. Yao, ``Diversity Creation Methods: A Survey and Categorisation,'' Information Fusion, 6(1):5-20, January 2005.*
- *K. Tang, P. N. Suganthan and X. Yao, ``An Analysis of Diversity Measures,'' Machine Learning, 65:247-271, 2006.*

A Logical View of Ensemble Learning

[Question] How to

1. Create datasets
2. Build classifiers
3. Combine classifiers



Simple Ensemble Approaches

- **Train different models on the same data set, then *vote* (classification) or *average* (regression) of predictions of multiple trained models.**

- **Train the same model multiple times on different data sets generated from the original data set.**

- **Train different models on multiple datasets.**

Constructing an Ensemble Classifier

- **Dataset level:**
  - **By manipulating the training set:**
    - **Create multiples data sets by sampling the original data.**
    - **Methods differ in the sampling distribution.**
    - **Examples: *Boosting*, *Bagging*.**
  - **By manipulating the input features:**
    - **Methods differ in the feature selection.**
    - **Example: *Random Forest*.**
  - **By manipulating the class labels:**
    - **Methods differ in the class partition.**
    - **Example: *error-correcting output coding*.**
- **Classifier level:**
  - **By manipulating the learning algorithms: depends on the classifier used.**

How to Combine

- **Majority voting**
- **Weighted voting**
- **Averaging**
- **Weighted averaging**
- **Stacked generalization**
- **…**

Summary

Summary

	Bagging	Boosting	Random Forest
Sampling distribution	Uniform random distribution, with replacement	Adaptive weighted distribution	Non-adaptive weighted distribution
#Feature	Same as original data set	Same as original data set	Fewer/More
Parallel?	Yes	No	No
Suitable problems /classifiers	Regression Classification	Classification Regression	Only applied to decision trees

Reading Materials for This Lecture

[1] AI textbook (pages 748-752).

[2] P. Tan, M. Steinbach and V. Kumar. *Introduction to Data Mining* (pages 276-293).

[3] Liu, Y., & Yao, X. (1999). Ensemble learning via negative correlation. *Neural networks*, *12*(10), 1399-1404.

[4] Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2), 123-140.

[5] Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32.

[6] Wang, S., & Yao, X. (2013). Relationships between diversity of classification ensembles and single-class performance measures. *IEEE Transactions on Knowledge and Data Engineering*, *25*(1), 206-219.