Chapter ML:II (continued)

II. Machine Learning Basics

Measuring Performance
True Misclassification Rate

Definition 8 (True Misclassification Rate)

Let X be a feature space with a finite number of elements. Moreover, let C be a set of classes, let $y : X \to C$ be a classifier, and let c be the target concept to be learned. Then the true misclassification rate, denoted as $Err^*(y)$, is defined as follows:

$$Err^*(y) = \frac{|\{\mathbf{x} \in X : y(\mathbf{x}) \neq c(\mathbf{x})\}|}{|X|}$$

Measuring Performance
True Misclassification Rate

Definition 8 (True Misclassification Rate)

Let X be a feature space with a finite number of elements. Moreover, let C be a set of classes, let $y : X \to C$ be a classifier, and let c be the target concept to be learned. Then the true misclassification rate, denoted as $Err^*(y)$, is defined as follows:

$$Err^*(y) = \frac{|\{\mathbf{x} \in X : y(\mathbf{x}) \neq c(\mathbf{x})\}|}{|X|}$$

Problem:

- ❑ Usually the *total function* c is unknown.

Solution:

- ❑ Estimation of $Err^*(y)$ with $Err(y, D_{ts})$, i.e., evaluating y on a subset $D_{ts} \subseteq D$ of carefully chosen examples D. Recall that for the feature vectors in D the target concept c is known.
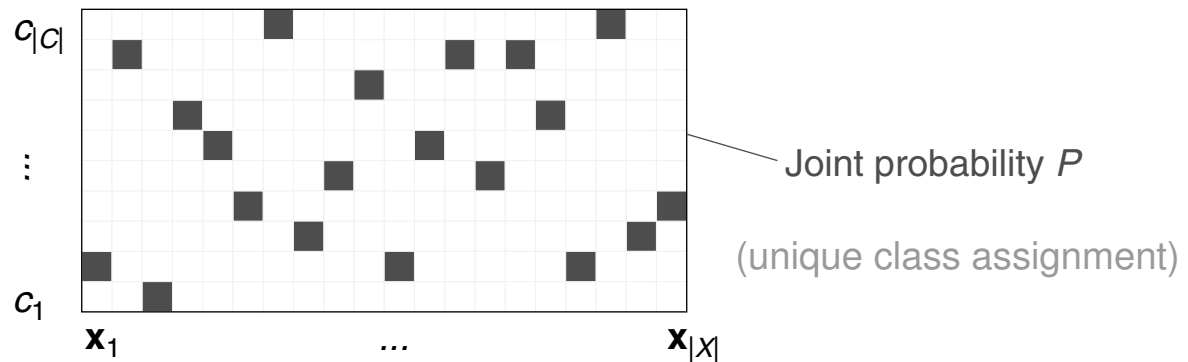
Remarks:

- Instead of the term "true misclassification rate" we may also use the term "true misclassification error" or simply "true error".

- The English word "rate" can be used to denote both the mathematical concept of a flow quantity (a change of a quantity per time unit) as well as the mathematical concept of a *portion*, a *percentage*, or a *ratio*, which has a stationary (= time-independent) semantics. Note that the latter semantics is meant here when talking about the misclassification rate.

- Unfortunately, the German word „Rate" is often (mis)used to denote the mathematical concept of a portion, a percentage, or a ratio. Taking a precise mathematical standpoint, the correct German words are „Anteil" or „Quote". I.e., a semantically correct translation of misclassification rate is „Missklassifikationsanteil", and not „Missklassifikationsrate".
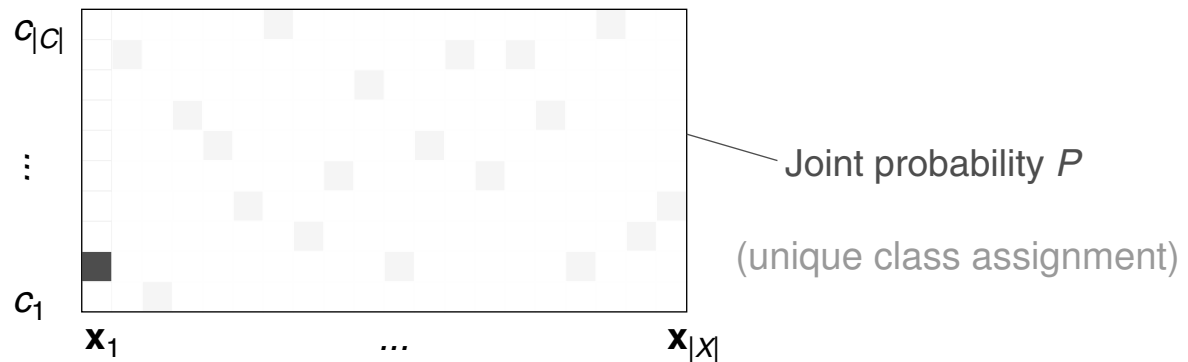
Measuring Performance

Let X and C be defined as before. Moreover, let P be a probability measure on $X \times C$. Then $P(\mathbf{x}, c)$ (precisely: $P(\mathcal{H} = \mathbf{x}, \mathcal{C} = c)$) denotes the probability that feature vector $\mathbf{x} \in X$ belongs to class $c \in C$. Illustration:

Joint probability P

(unique class assignment)

Let X and C be defined as before. Moreover, let P be a probability measure on $X \times C$. Then $P(\mathbf{x}, c)$ (precisely: $P(\mathcal{H} = \mathbf{x}, \mathcal{C} = c)$) denotes the probability that feature vector $\mathbf{x} \in X$ belongs to class $c \in C$. Illustration:

Joint probability P
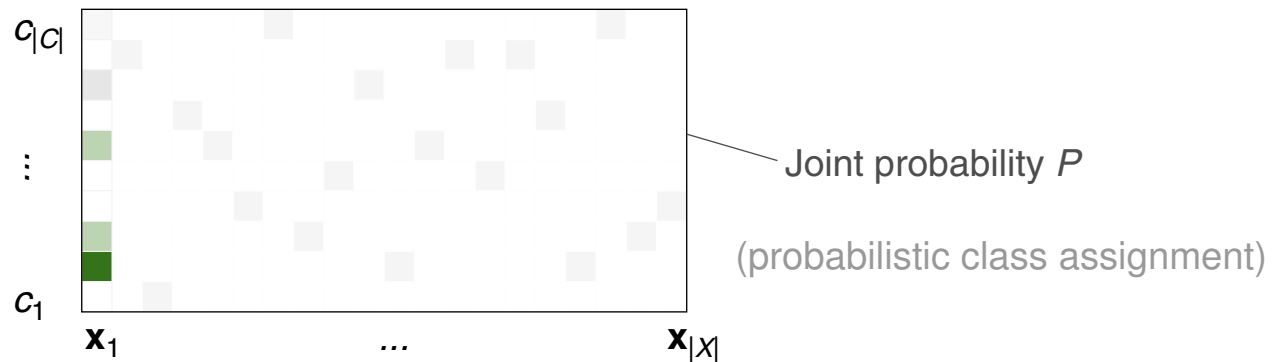
(unique class assignment)

Measuring Performance

Let X and C be defined as before. Moreover, let P be a probability measure on $X \times C$. Then $P(\mathbf{x}, c)$ (precisely: $P(\mathcal{H} = \mathbf{x}, \mathcal{C} = c)$) denotes the probability that feature vector $\mathbf{x} \in X$ belongs to class $c \in C$. Illustration:

Joint probability P

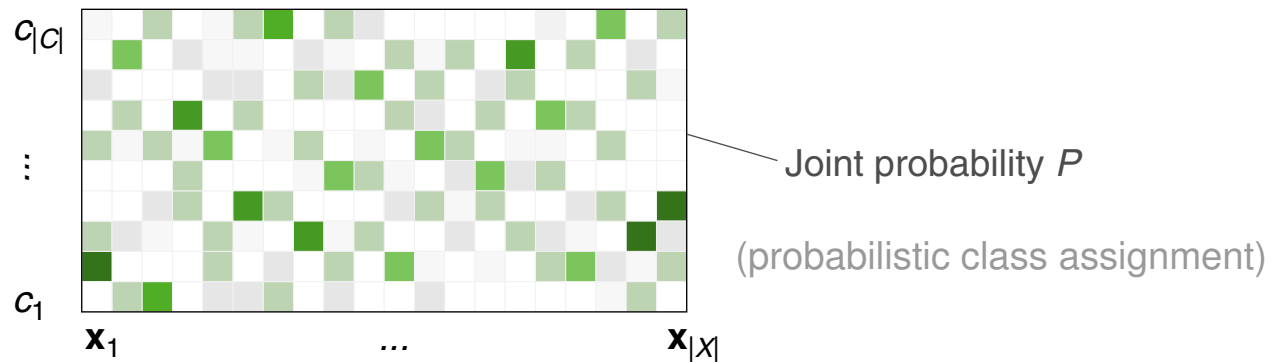(probabilistic class assignment)

Measuring Performance

Let X and C be defined as before. Moreover, let P be a probability measure on $X \times C$. Then $P(\mathbf{x}, c)$ (precisely: $P(\mathcal{H} = \mathbf{x}, \mathcal{C} = c)$) denotes the probability that feature vector $\mathbf{x} \in X$ belongs to class $c \in C$. Illustration:

Joint probability P

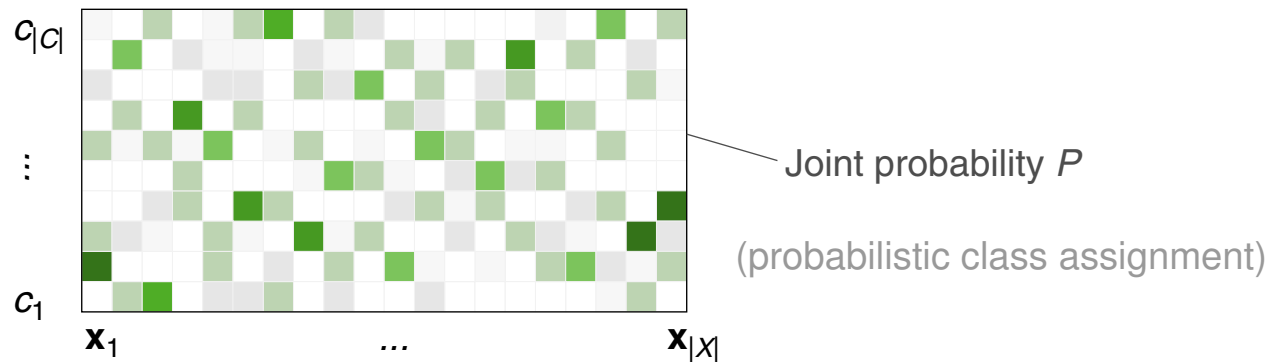(probabilistic class assignment)

Measuring Performance

Let X and C be defined as before. Moreover, let P be a probability measure on $X \times C$. Then $P(\mathbf{x}, c)$ (precisely: $P(\mathcal{H} = \mathbf{x}, \mathcal{C} = c)$) denotes the probability that feature vector $\mathbf{x} \in X$ belongs to class $c \in C$. Illustration:

Joint probability P

(probabilistic class assignment)

$$Err^*(y) = \sum_{\mathbf{x} \in X} \sum_{c \in C} P(\mathbf{x}, c) \cdot I(y(\mathbf{x}), c), \quad \text{with } I(y(\mathbf{x}), c) = \begin{cases} 0 & \text{if } y(\mathbf{x}) = c \\ 1 & \text{otherwise} \end{cases}$$

Measuring Performance

True Misclassification Rate: Probabilistic Foundation (continued)

Let X and C be defined as before. Moreover, let P be a probability measure on $X \times C$. Then $P(\mathbf{x}, c)$ (precisely: $P(\mathcal{H} = \mathbf{x}, \mathcal{C} = c)$) denotes the probability that feature vector $\mathbf{x} \in X$ belongs to class $c \in C$. Illustration:
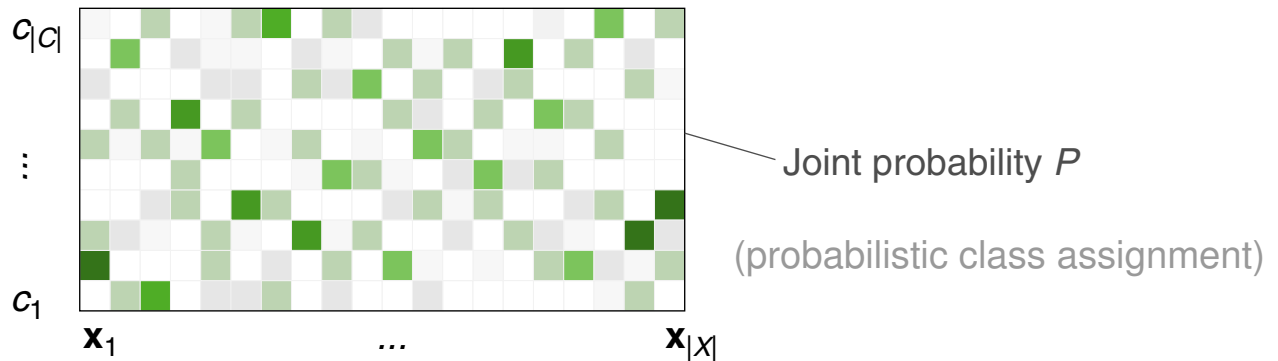

Joint probability P

(probabilistic class assignment)

$$Err^*(y) = \sum_{\mathbf{x} \in X} \sum_{c \in C} P(\mathbf{x}, c) \cdot I(y(\mathbf{x}), c), \quad \text{with } I(y(\mathbf{x}), c) = \begin{cases} 0 & \text{if } y(\mathbf{x}) = c \\ 1 & \text{otherwise} \end{cases}$$

$D = \{(\mathbf{x}_1, c_1)), \ldots, (\mathbf{x}_n, c_n))\} \subseteq X \times C$ is a set of examples whose elements are drawn independently and according to the same P.
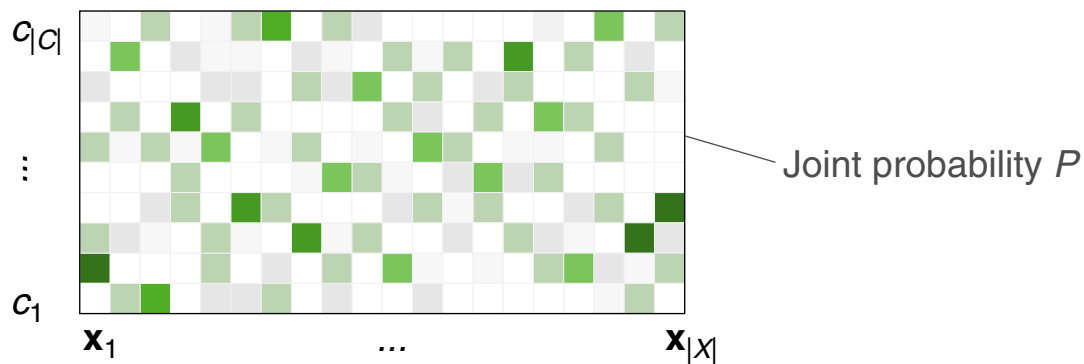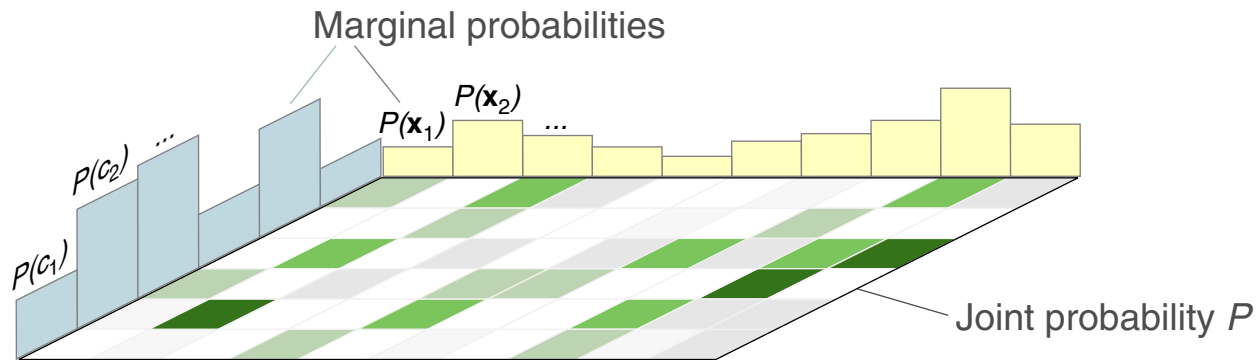
Remarks:

❑ $\mathcal{H}$ and $\mathcal{C}$ are random variables with domains X and C respectively.

❑ The function $c(\mathbf{x})$ is modeled as random variable, $\mathcal{C}$, since in the real world the classification of a feature vector $\mathbf{x}$ may not be deterministic but the result of a random (measuring) process. Keyword: label noise.

❑ Let A and B denote two events, e.g., $A =$ "$\mathcal{H} = \mathbf{x}$" and $B =$ "$\mathcal{C} = c$". Then the following expressions are syntactic variants, i.e., they are semantically equivalent: $P(A, B)$, $P(A$ and $B)$, $P(A \wedge B)$.

❑ Also the sampling process is a stochastic process: The elements in D and D_{ts} are considered as random variables that are both independent of each other and identically distributed. This property of a set of random variables is abbreviated with "i.i.d."
If the elements in D or D_{ts} are not chosen according to P, then $Err(y, D_{ts})$ cannot be used as an estimation of $Err^*(y)$. Keyword: sample selection bias

True Misclassification Rate: Probabilistic Foundation (continued)

Illustration of the marginal probabilities $P(c_i)$ and $P(\mathbf{x}_j)$:

Remarks:

❑ $P(\mathbf{x} \mid c_i)$ is the probability distribution of $\mathcal{H}$ under class $\mathcal{C} = c_i$.
$P(\mathbf{x} \mid c_i)$ is also called class-conditional   probability [density] function.

In the illustration: the distribution of $\mathbf{x}$ (consider a row) for a certain class c.
Summation [integration] over the $\mathbf{x} \in X$ yields the marginal probability $P(c_i)$.

❑ $P(c \mid \mathbf{x}_j)$ is the probability distribution of $\mathcal{C}$ under feature vector $\mathcal{H} = \mathbf{x}_j$.
$P(c \mid \mathbf{x}_j)$ is also called conditional   class probability function.

In the illustration: the distribution of c (consider a column) for a certain feature vector $\mathbf{x}$.
Summation over the $c \in C$ yields the marginal probability $P(\mathbf{x}_j)$.

❑ $P(\mathbf{x}_j, c_i) = P(\mathbf{x}_j \mid c_i) \cdot P(c_i)$, where $P(c_i)$ is the a-priori probability for observing c_i, and
$P(\mathbf{x}_j \mid c_i)$ is the probability for the event of $\mathbf{x}_j$ given the event of c_i.

Likewise, $P(\mathbf{x}_j, c_i) = P(c_i, \mathbf{x}_j) = P(c_i \mid \mathbf{x}_j) \cdot P(\mathbf{x}_j)$, where $P(\mathbf{x}_j)$ is the a-priori probability for
observing $\mathbf{x}_j$, and $P(c_i \mid \mathbf{x}_j)$ is the probability for the event of c_i given the event of $\mathbf{x}_j$.

❑ Let both events $\mathcal{H} = \mathbf{x}_j$ and $\mathcal{C} = c_i$ have occurred already, and, let $\mathbf{x}_j$ be known and c_i be
unknown. Then, $P(\mathbf{x}_j \mid c_i)$ is called *likelihood* (for the event of $\mathbf{x}_j$ given the event of c_i).

Measuring Performance

❑ $D = \{(\mathbf{x}_1, c(\mathbf{x}_1)), \ldots, (\mathbf{x}_n, c(\mathbf{x}_n))\} \subseteq X \times C$ is a set of examples.

❑ $D_{tr} = D$ is the training set.

❑ $y : X \to C$ is a classifier learned on the basis of D_{tr}.

Training error = misclassification rate with respect to D_{tr} :

$$Err(y, D_{tr}) = \frac{|\{(\mathbf{x}, c(\mathbf{x})) \in D_{tr} : y(\mathbf{x}) \neq c(\mathbf{x})\}|}{|D_{tr}|}$$

Measuring Performance

□ $D = \{(\mathbf{x}_1, c(\mathbf{x}_1)), \ldots, (\mathbf{x}_n, c(\mathbf{x}_n))\} \subseteq X \times C$ is a set of examples.

□ $D_{tr} = D$ is the training set.

□ $y : X \rightarrow C$ is a classifier learned on the basis of D_{tr}.

Training error = misclassification rate with respect to D_{tr} :

$$Err(y, D_{tr}) = \frac{|\{(\mathbf{x}, c(\mathbf{x})) \in D_{tr} : y(\mathbf{x}) \neq c(\mathbf{x})\}|}{|D_{tr}|}$$

Problems:

□ $Err(y, D_{tr})$ is based on examples that are also exploited to learn y.

→ $Err(y, D_{tr})$ quantifies memorization but not the generalization capability of y.

→ $Err(y, D_{tr})$ is an optimistic estimation, i.e., it is constantly lower compared to the error incurred when applying y in the wild.

Measuring Performance

2-Fold Cross-Validation (Holdout Estimation) [True Misclassification Rate]

- $D = \{(\mathbf{x}_1, c(\mathbf{x}_1)), \ldots, (\mathbf{x}_n, c(\mathbf{x}_n))\} \subseteq X \times C$ is a set of examples.

- $D_{tr} \subset D$ is the training set.

- $y : X \to C$ is a classifier learned on the basis of D_{tr}.

- $D_{ts} \subset D$ with $D_{ts} \cap D_{tr} = \emptyset$ is a test set.

Holdout estimation = misclassification rate with respect to D_{ts} :

$$Err(y, D_{ts}) = \frac{|\{(\mathbf{x}, c(\mathbf{x})) \in D_{ts} : y(\mathbf{x}) \neq c(\mathbf{x})\}|}{|D_{ts}|}$$

Measuring Performance

2-Fold Cross-Validation (Holdout Estimation) [True Misclassification Rate]

- $D = \{(\mathbf{x}_1, c(\mathbf{x}_1)), \ldots, (\mathbf{x}_n, c(\mathbf{x}_n))\} \subseteq X \times C$ is a set of examples.

- $D_{tr} \subset D$ is the training set.

- $y : X \to C$ is a classifier learned on the basis of D_{tr}.

- $D_{ts} \subset D$ with $D_{ts} \cap D_{tr} = \emptyset$ is a test set.

Holdout estimation = misclassification rate with respect to D_{ts} :

$$Err(y, D_{ts}) = \frac{|\{(\mathbf{x}, c(\mathbf{x})) \in D_{ts} : y(\mathbf{x}) \neq c(\mathbf{x})\}|}{|D_{ts}|}$$

Requirements:

- D_{tr} and D_{ts} must be governed by the same distribution.

- D_{tr} and D_{ts} should have similar sizes.

Remarks:

❏ A typical value for splitting D into training set D_{tr} and test set D_{ts} is 2:1.

❏ When splitting D into D_{tr} and D_{ts} one has to ensure that the underlying distribution is maintained. Keywords: stratification, sample selection bias

Measuring Performance

k-Fold Cross-Validation [Holdout Estimation]

- Form k test sets by splitting D into disjoint sets $D_1, \ldots, D_k$ of similar size.

- For $i = 1, \ldots, k$ do:

  1. $y_i : X \to C$ is a classifier learned on the basis of $D \setminus D_i$

  2. $Err(y_i, D_i) = \dfrac{|\{(\mathbf{x}, c(\mathbf{x})) \in D_i : y_i(\mathbf{x}) \neq c(\mathbf{x})\}|}{|D_i|}$

Cross-validated misclassification rate:

$$Err_{cv}(y, D) = \frac{1}{k} \sum_{i=1}^{k} Err(y_i, D_i)$$

Measuring Performance

Special case with $k = n$:

❑ Determine the cross-validated misclassification rate for $D \setminus D_i$ where $D_i = \{(\mathbf{x}_i, c(\mathbf{x}_i))\}, \ i \in \{1, \ldots, n\}$.

Measuring Performance

n-Fold Cross-Validation (Leave One Out)

Special case with $k = n$:

❑ Determine the cross-validated misclassification rate for $D \setminus D_i$ where $D_i = \{(\mathbf{x}_i, c(\mathbf{x}_i))\}, \ i \in \{1, \ldots, n\}$.

Problems:

❑ High computational effort if D is large.

❑ Singleton test sets ($|D_i| = 1$) are never stratified since they contain a single class only.

Remarks:

❏ For large k the set $D \setminus D_i$ is of similar size as D. Hence *Err*(y_i, D_i) is close to *Err*(y, D), where y is the classifier learned on the basis of the entire set D.

❏ n-fold cross-validation is a special case of exhaustive cross-validation methods, which learn and test on all possible ways to divide the original sample into a training and a validation set.
[Wikipedia]

Measuring Performance

Bootstrapping [Holdout Estimation]

Resampling the example set D :

- For $j = 1, \ldots, l$ do:

    1. Form training set D_j by drawing m examples from D with replacement.

    2. $y_j : X \to C$ is a classifier learned on the basis of D_j

    3. $Err(y_j, D \setminus D_j) = \dfrac{|\{(\mathbf{x}, c(\mathbf{x})) \in D \setminus D_j : y_j(\mathbf{x}) \neq c(\mathbf{x})\}|}{|D \setminus D_j|}$

Bootstrapped misclassification rate:

$$Err_{bt}(y, D) = \frac{1}{l} \sum_{j=1}^{l} Err(y_j, D \setminus D_j)$$

Remarks:

- ❑ Let $|D| = n$. The probability that an example is not considered is $(1 - 1/n)^m$. Similarly, the probability that an example is considered at least once is $1 - (1 - 1/n)^m$.

- ❑ If m gets closer to n, then $1 - (1 - 1/n)^m \approx 1 - 1/e \approx 0.632$. I.e., each training set contains about 63.2% of the examples in D.

- ❑ The classifiers $y_1, \ldots, y_l$ can be used in a combined fashion, called *ensemble*, where the class is determined by means of a majority decision:

$$y(\mathbf{x}) = \underset{c \in C}{\text{argmax}} \, |\{j \in \{1, \ldots, l\} : y_j(\mathbf{x}) = c\}|$$

Measuring Performance

Use of a cost measure for the misclassification of a feature vector $\mathbf{x}$ in class c' instead of in class c:

$$cost(c' \mid c) \begin{cases} \geq 0 & \text{if } c' \neq c \\ = 0 & \text{otherwise} \end{cases}$$

Estimation of $Err^*_{cost}(y)$ based on a sample $D_{ts} \subseteq D$:

$$Err_{cost}(y, D_{ts}) = \frac{1}{|D_{ts}|} \cdot \sum_{(\mathbf{x}, c(\mathbf{x})) \in D_{ts}} cost(y(\mathbf{x}) \mid c(\mathbf{x}))$$

Remarks:

❑ The misclassification rate *Err* is a special case of Err_{cost} with $cost(c' \mid c) = 1$ for $c' \neq c$.