

# Time Series Classification and Ensemble Methods

Pietro Sala

Data Mining 24/25 - Exercise 4

## 1 Base Classifier: Sequential Boost Classifier (SBC)

In the analysis of time-series data, we often encounter the need for specialized classification methods that can effectively capture temporal patterns. The Sequential Boost Classifier (SBC) addresses this need by combining multiple weak learners through a boosting approach, where each weak learner focuses on a specific time point in the sequence.

Given an input sequence  $x = [x_0, \dots, x_n]$ , we construct our classifier as an ordered collection of weighted weak learners:

$$H = [(\alpha_0, h_0), \dots, (\alpha_m, h_m)]$$

Each weak learner pair  $(\alpha_i, h_i)$  plays a specific role in the classification:

- $\alpha_i$  represents the weight assigned to the learner, reflecting its importance in the final decision
- $h_i = (t_i, d_i)$  is our weak learner, characterized by:
  - A threshold value  $t_i$
  - A direction indicator  $d_i \in \{-1, 1\}$

When applied to an input value  $x$ , each weak learner produces a prediction through the function:

$$h_i(x) = d_i \cdot \text{sgn}(t_i - x)$$

Here, the sign function  $\text{sgn} : \mathbb{R} \rightarrow \{-1, 1\}$  is defined as:

$$\text{sgn}(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

The final classification of a sequence combines these weak learners' predictions in a weighted sum:

$$H(x) = \text{sgn} \left( \sum_{i=0}^m \alpha_i \cdot h_i(x_i) \right)$$

### 1.1 Training Process

The SBC learning process requires careful consideration of several parameters:

- **Required Parameters:**

- A training dataset  $(X, Y)$  where the labels  $Y$  are binary values from  $\{-1, 1\}$

- **Optional Parameters:**

- The maximum number of weak learners (*n\_weak\_learners*)
- A validation set  $(X, Y)$  for monitoring the classifier's performance during training

### 1.2 Making Predictions

During the prediction phase, our classifier produces a normalized score computed as:

$$\frac{\sum_{i: h_i(x_i)=1} \alpha_i}{\sum_i \alpha_i}$$

To enhance prediction accuracy, we offer an optional truncation mechanism:

- When *truncate* is enabled, the classifier uses only the most effective prefix of weak learners, as determined by validation set performance

## 2 Bagging Ensemble of SBC (BE-SBC)

Building upon the base SBC classifier, we introduce a more robust approach through ensemble

learning. The BE-SBC leverages bagging techniques to create a diverse collection of SBC classifiers, each bringing its unique perspective to the classification task.

## 2.1 Ensemble Construction

The creation of a BE-SBC requires several key parameters:

- **Required Parameters:**

- Training data  $(X, Y)$  supporting both binary and multiclass scenarios
- The desired number of SBC classifiers ( $n\_estimators$ )

- **Optional Parameters:**

- Configuration options for individual SBCs, such as the number of weak learners

## 2.2 Prediction Mechanisms

For binary classification, standard majority voting applies. For multiclass scenarios with classes  $\{c_1, \dots, c_l\}$ , each SBC  $H_i$  in the ensemble is associated with a class  $H_i.class \in \{c_1, \dots, c_l\}$  selected during training for its one-versus-all classification.

The ensemble provides three specialized voting schemes for multiclass prediction:

- **Majority Voting:** Returns the distribution

$$P_{maj}(c_k|x) = \frac{|\{H_i : H_i(x) = 1 \wedge H_i.class = c_k\}|}{|\{H_i : H_i(x) = 1\}|}$$

considering only classifiers that output 1 on input  $x$

- **Weighted Voting:** Returns the weighted distribution

$$P_w(c_k|x) = \frac{\sum_{H_i: H_i(x)=1 \wedge H_i.class=c_k} acc_i}{\sum_{H_i: H_i(x)=1} acc_i}$$

where  $acc_i$  is the accuracy of  $H_i$  on the validation set

- **Track Record:** Combines positive and negative predictions via

$$P_{tr}(c_k|x) = \frac{\sum_{H_i: H_i.class=c_k} w_i}{\sum_{i=1}^n w_i}$$

where

$$w_i = \begin{cases} acc_i & \text{if } H_i(x) = 1 \\ fnr_i & \text{if } H_i(x) = -1 \end{cases}$$

with  $fnr_i$  being the false negative rate of  $H_i$  on the validation set

For each SBC  $H_i$  in the ensemble:

- During training, a class  $c_k$  is randomly selected and assigned to  $H_i.class$
- Training labels are created as:

$$y_i[j] = \begin{cases} 1 & \text{if } y[j] = H_i.class \\ -1 & \text{otherwise} \end{cases}$$

- This creates a one-versus-all classifier for the selected class

## 3 SBC-based Isolation-like Ensemble (I-SBC)

The I-SBC extends our framework into the realm of unsupervised learning, creating an ensemble of SBC classifiers specifically designed for anomaly detection and distance-based analysis.

### 3.1 Learning Process

The I-SBC construction requires:

- **Core Parameters:**

- An unlabeled dataset  $X$
- The desired ensemble size ( $n\_estimators$ )

- **Optional Parameters:**

- Configuration settings for individual SBCs

### 3.2 Distance Measures

Given two input sequences  $x_1, x_2$  and an I-SBC ensemble  $E$  with  $n$  estimators, where each estimator  $H_i$  consists of  $m$  weak learners, we define three similarity measures  $s$  and their corresponding distances  $d$ :

- **Breiman-like:** Measures similarity through behavioral consistency

$$s_{breiman}(x_1, x_2) = \frac{\left| \left\{ H_i \in E : \bigwedge_{j=1}^{|H_i|} h_j^i(x_1) = h_j^i(x_2) \right\} \right|}{|E|}$$

$$d_{breiman}(x_1, x_2) = 1 - s_{breiman}(x_1, x_2)$$

- **Zhu-like:** Evaluates similarity based on matching decision sequences

$$s_{zhu}(x_1, x_2) = \frac{\sum_{i=1}^n \text{maxPrefix}(H_i, x_1, x_2)}{\sum_{i=1}^n |H_i|}$$

$$d_{zhu}(x_1, x_2) = 1 - s_{zhu}(x_1, x_2)$$

where  $\text{maxPrefix}(H_i, x_1, x_2)$  returns the length of the longest prefix of weak learners in  $H_i$  that produce identical outputs for both  $x_1$  and  $x_2$ .

$$\text{maxPrefix}(H_i, x_1, x_2) = \max_{1 \leq j \leq |H_i|, h_{j'}^i(x_1) = h_{j'}^i(x_2) \text{ for each } 1 \leq j' \leq j} j$$

- **Ratio-like:** Considers the proportion of matching individual decisions

$$s_{ratio}(x_1, x_2) = \frac{\sum_{i=1}^n \sum_{j=1}^{|H_i|} \llbracket h_j^i(x_1) = h_j^i(x_2) \rrbracket}{\sum_{i=1}^n |H_i|}$$

$$d_{ratio}(x_1, x_2) = 1 - s_{ratio}(x_1, x_2)$$

where:

- $h_j^i$  denotes the  $j$ -th weak learner of the  $i$ -th SBC
- $\llbracket P \rrbracket$  is the Iverson bracket: 1 if predicate  $P$  is true, 0 otherwise

These distance measures satisfy several important properties:

- Symmetry:  $s(x_1, x_2) = s(x_2, x_1)$  and  $d(x_1, x_2) = d(x_2, x_1)$
- Range:  $s, d : X \times X \rightarrow [0, 1]$
- Complementary:  $s(x_1, x_2) + d(x_1, x_2) = 1$
- Identity:  $s(x_1, x_2) = 1$  (and thus  $d(x_1, x_2) = 0$ )  $\iff$  identical behavior across all relevant components
- Maximum dissimilarity:  $s(x_1, x_2) = 0$  (and thus  $d(x_1, x_2) = 1$ )  $\iff$  completely opposite behavior

## 4 I-SBC Nearest Neighbor Classifier

Given a trained I-SBC ensemble and its associated distance measure  $d$ , we define the I-SBC Nearest Neighbor classifier (I-SBC-NN) as follows.

For an input  $x$  and classes  $\{c_1, \dots, c_l\}$ , let:

$$x^{c_k} = \arg \min_{x' \in X: y(x') = c_k} d(x, x')$$

be the nearest neighbor of  $x$  among training samples of class  $c_k$ , where  $y(x')$  denotes the true class of sample  $x'$ .

The classifier produces a probability distribution over classes defined as:

$$P(c_k|x) = \frac{1 - d(x, x^{c_k})}{\sum_{j=1}^l (1 - d(x, x^{c_j}))}$$

This assigns higher probabilities to classes whose nearest representatives are closer to the input according to the chosen I-SBC distance measure.

## 5 Assignment Tasks

Students should use at least two supervised time-series datasets for their analysis:

- One binary classification dataset
- One multiclass classification dataset (with at least 3 classes)

The implementation and analysis involves three main challenges:

### 1) Clustering Analysis

- Train I-SBC on  $X$  (unsupervised)
- Compare the three distance measures using one internal metric and one external metric

### 2) BE-SBC Conformal Prediction

- Train a BE-SBC classifier on  $(X, Y)$  (supervised)
- Calibrate a conformal classifier using the trained BE-SBC as the underlying predictor
- Verify the calibration of the conformal classifier and analyze its efficiency

### 3) Distance-based Classification

- Train an I-SBC-NN classifier on  $(X, Y)$  (supervised)
- Calibrate a conformal classifier using the trained I-SBC-NN as the underlying predictor
- Verify the calibration of the conformal classifier and analyze its efficiency

The analysis must be performed on both datasets (binary and multiclass). Implement all functions, scripts, and objects needed to perform the above tasks.