



**HAL**  
open science

## Learning temporal matchings for time series discrimination

Cédric Frambourg, Ahlame Douzal-Chouakria, Éric Gaussier

► **To cite this version:**

Cédric Frambourg, Ahlame Douzal-Chouakria, Éric Gaussier. Learning temporal matchings for time series discrimination. 2014. hal-00996951

**HAL Id: hal-00996951**

**<https://hal.science/hal-00996951v1>**

Submitted on 27 May 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



21 (as clustering or classification), weakening its efficiency on complex applications.

22  
23 To partly overcome these problems, several variants of DTW have been pro-  
24 posed to improve performance in classification or clustering. They mostly aim  
25 to more finely estimate the DTW parameters, namely, warping constraints, the  
26 time weighting, or the underlying cost function. In Yu et al. [20], without being  
27 exhaustive, a multiple bands global path constraint, extending the Sakoe-Chiba  
28 band [16], is learned through a brute-force search to maximize the marginal  
29 nearest neighbor classification. In Ratanamahatana and Keogh [15], a multiple  
30 bands global path is estimated for each class through a forward or backward hill-  
31 climbing search to maximize the  $k$ NN accuracy. These methods assume equally  
32 weighted times and a constant values-based cost function. In Gaudin and Ni-  
33 coloyannis [6], a genetic algorithm approach is used to approximate a global  
34 time weighting matrix. In Jeong et al. [9], a global logistic weight function  
35 is evaluated by searching empirically through the entire data set, and aligned  
36 values are penalized according to the induced warping. In both works, the time  
37 weighting estimation is similarly formulated as a  $k$ NN accuracy maximization  
38 problem; as in classical DTW, they involve a values-based cost function without  
39 warping constraints. In Douzal-Chouakria and Amblard [4], a unified formal-  
40 ism for an adaptive DTW cost function is proposed, specifically to cover both  
41 the behavior and value components; the cost function parameters are estimated  
42 using a classification tree for time series. In Xie and Wiltgen [19], DTW based  
43 on a linear cost function is proposed, involving local (i.e. derivative based) and  
44 global (i.e. values-based) features. The latter methods focus on estimating an  
45 adaptive cost function while considering equally weighted times without warp-  
46 ing constraints.

47  
48 Using an approach different from DTW alignment, Gaffney et al. [5] propose  
49 a probabilistic model that allows for the derivation of an EM learning algorithm  
50 that handles clustering and matching processes jointly. For this purpose, a B-  
51 spline regression mixture models are proposed for clustering time series data,  
52 augmented for the alignment process, with affine transformations for scaling and  
53 translation in time. In the same spirit, Ramsay et al. [14] propose a method  
54 that learns an alignment function for each time series, parameterized with order  
55 one B-spline coefficients. Both of the above proposed alignments remain limited  
56 to time series of the same class, which limits the discriminative power of these  
57 methods. In Listgarten et al. [11] a hierarchical Bayesian model is proposed,  
58 which aligns time series simultaneously across all classes, while detecting and  
59 characterizing class-specific differences. The proposed model assumes that the  
60 time series from all clusters originate from the same source, i.e. that they share  
61 a common global structure with rare differences between clusters. Although  
62 these approaches yield more accurate alignments between time series, they as-  
63 sume that time series from the same class share a single global structure.

64  
65 However, it is indisputable that time series peculiarities may be more compli-  
66 cated in real applications. In particular, it is not unusual that time series from a

67 same class exhibit differences in overall behaviors, or that time series behaviors  
 68 exhibit similarities across classes. Consequently, for time series discrimination, it  
 69 appears important that the time series matching relies on the commonly shared  
 70 features within the classes and the most differential ones between classes. Such  
 71 challenging linkages can be achieved by training time series matching within and  
 72 between classes to localize discriminative features. However, these temporally  
 73 unconstrained linkages are difficult to reach with alignments that are mainly  
 74 founded on monotone functions that preserve the temporal order constraints.

75  
 76 With this in mind, we propose a new approach for training time series  
 77 discriminative matching that highlights class-specific characteristics and dif-  
 78 ferences. The main idea consists of using a discriminant criterion based on  
 79 variance/covariance to strengthen or weaken links according to their contribu-  
 80 tions to the variances within and between classes. The variance/covariance  
 81 measure is used in many approaches, including discriminant analysis, clustering  
 82 and classification [7, 2, 13, 18]. However, to the best of our knowledge, it has  
 83 never been investigated for learning temporal matching to discriminate classes  
 84 of time series. To this end, we propose a new formalization of the classical vari-  
 85 ance/covariance for a set of time series, as well as for a partition of time series  
 86 (Section 2). In Section 3, we present a method for training the intra and inter  
 87 class time series matching, driven by within-class variance minimization and  
 88 between-class variance maximization. The learned discriminative matching is  
 89 then used to define a locally weighted time series metric that restricts the time  
 90 series comparison to discriminative features (Section 4). The relevance of the  
 91 proposed approach is studied by  $k$ -nearest neighbor time series classification on  
 92 real datasets. In Section 5, the experiments carried out reveal that the proposed  
 93 approach is able to capture fine-grained distinctions between time series, all the  
 94 more so that time series of a same class exhibit dissimilar behaviors.

95  
 96 Let us underline the main characteristics of the proposed method: 1) It  
 97 enlarges time series alignments to a general temporal matching that localizes  
 98 the common features within classes and the distinctive ones between classes, 2)  
 99 It takes into account time series of possibly dissimilar behaviors within classes;  
 100 3) It is trained according to the temporal dynamics of all time series within and  
 101 between classes.

## 102 2. Variance/covariance for time series

103 We first recall the definition of the conventional variance/covariance ma-  
 104 trix, prior to introducing its formalization for time series data. Let  $X$  be the  
 105  $(n \times p)$  data matrix containing  $n$  observations of  $p$  numerical variables. The  
 106 conventional  $(p \times p)$  variance/covariance matrix expression is:

$$V = X^t(I - UP)^tP(I - UP)X \quad (1)$$

107

108 where,  $I$  is the diagonal identity matrix,  $U$  the unit matrix, and  $P$  a diagonal  
 109 weight matrix of general term  $p_i = \frac{1}{n}$  for equally weighted observations.

110 In the following, we provide a generalization of the variance/covariance ex-  
 111 pression Eq.(1) to multivariate time series observations.

112 *2.1. Variance induced by a set of time series*

113 For a set of time series, let  $X$  Eq.(2) be the  $(nT \times p)$  matrix providing the  
 114 description of  $n$  multivariate time series  $S_1, \dots, S_n$  by  $p$  numerical variables at  $T$   
 115 time stamps.

116

$$X = \begin{matrix} & X_1 & \dots & X_p \\ \dots & & & \\ S_l & \begin{pmatrix} \dots & & \\ x_{11}^l & \dots & x_{1p}^l \\ \dots & \dots & \dots \\ x_{T1}^l & \dots & x_{Tp}^l \\ \dots & & \dots \end{pmatrix} & \\ \dots & & & \end{matrix} \quad (2)$$

117 The matching between  $n$  time series can be described by a matrix  $M$  Eq.(3)  
 118 of positive terms composed of  $n^2$  block matrices  $M^{ll'}$  ( $l = 1, \dots, n; l' = 1, \dots, n$ ).  
 119 A block  $M^{ll'}$  Eq.(4) is a  $(T \times T)$  matrix that specifies the matching between  $S_l$   
 120 and  $S_{l'}$ , of general term  $m_{ii'}^{ll'} \in [0, 1]$  giving the weight of the link between the  
 121 observation  $i$  of  $S_l$  and  $i'$  of  $S_{l'}$ .

122

$$M = \begin{matrix} & S_1 & \dots & S_n \\ S_1 & \begin{bmatrix} M^{11} & \dots & M^{1n} \\ \dots & M^{ll'} & \dots \\ M^{n1} & \dots & M^{nn} \end{bmatrix} & \\ \dots & & & \\ S_n & & & \end{matrix} \quad (3)$$

124

$$M^{ll'} = S_l \begin{matrix} & S_{l'} \\ \begin{bmatrix} m_{11}^{ll'} & \dots & m_{1T}^{ll'} \\ \dots & m_{ii'}^{ll'} & \dots \\ m_{T1}^{ll'} & \dots & m_{TT}^{ll'} \end{bmatrix} & \end{matrix} \quad (4)$$

125

126 In particular, three basic matchings can be considered:

- 127
- 128 • A complete linkage connecting all observations of  $S_l$  and  $S_{l'}$ , whatever their  
 129 time stamps, is obtained by setting  $\forall i, i' \in \{1, \dots, T\}$ ,  $m_{ii'}^{ll'} = \frac{1}{T}$ , defined by  
 130  $M^{ll'} = \frac{1}{T}U_T$ ,  $U_T$  being the  $(T \times T)$  unit matrix
  - 131 • The Euclidean alignment connecting observations that occur at the same  
 132 time is obtained by setting  $\forall i, i' \in \{1, \dots, T\}$ ,  $m_{ii'}^{ll'} = 1$  if  $i = i'$  and 0  
 133 otherwise, described by  $M^{ll'} = I$ ;

134 • A dynamic time warping alignment is obtained by setting  $\forall i, i' \in \{1, \dots, T\}$ ,  
 135  $m_{ii'}^{l'l'} = 1$  if  $i$  is aligned with  $i'$  by the standard DTW, and 0 otherwise.

136 Then, the  $(p \times p)$  variance/covariance matrix  $V_M$  induced by a set of time series  
 137  $S_1, \dots, S_n$  connected to one another according to the matching matrix  $M$  can be  
 138 defined on the basis of Eq.(1), as:

$$V_M = X^t(I - M)^t P(I - M)X \quad (5)$$

139 where  $P$  is a  $(nT \times nT)$  diagonal matrix of weights, with  $p_i = \frac{1}{nT}$  for equally  
 140 weighted observations. Note that for a complete linkage matching,  $M$  is equal  
 141 to  $UP$  and  $V_M$  leads to a conventional variance covariance  $V$  Eq.(1).  
 142

143 For clarity and to simplify notation, we focus for the theoretical develop-  
 144 ments on univariate time series. The extension to the multivariate case is direct  
 145 and will be used in the experiments.  
 146

147 Thus, let  $x_i^l$  be the value of the variable  $X$  taken by  $S_l$  ( $l = 1, \dots, n$ ) at the  
 148  $i$ th time stamp ( $i = 1, \dots, T$ ).  
 149

150 **Definition 1.** *The variance  $V_M$  of the variable  $X$  is given by:*

$$V_M = \sum_{l=1}^n \sum_{i=1}^T p_i (x_i^l - \sum_{l'=1}^n \sum_{i'=1}^T m_{ii'}^{l'l'} x_{i'}^{l'})^2 \quad (6)$$

151

152 Note that each value  $x_i^l$  is centered relative to the term  $\sum_{l'=1}^n \sum_{i'=1}^T m_{ii'}^{l'l'} x_{i'}^{l'}$   
 153 estimating the average of  $X$  in the neighborhood of the time  $i$  of  $S_l$ . The  
 154 neighborhood of  $i$  is the set of instants  $i'$  of  $S_{l'}$  ( $l' = 1 \dots n$ ) connected to  $i$  with  
 155  $m_{ii'}^{l'l'} \neq 0$ . We now proceed to define the variance within and between classes  
 156 when the set of time series is partitioned into classes.

## 157 2.2. Variance induced by a partition of time series

158 Let us now consider a set of time series  $S_1, \dots, S_n$  partitioned into  $K$  classes,  
 159 with  $y_i \in \{1, \dots, K\}$  the class label of  $S_i$  and  $n_k$  the number of time series  
 160 belonging to class  $C_k$ . The definition of the *within variance* (i.e. the variance  
 161 within classes) and the *between variance* (i.e. the variance between classes)  
 162 induced by  $K$  classes is obtained by using the expression given in Eq.(5) based  
 163 on a matching  $M$  specified below.

164 **Definition 2.** *The within variance with an intra-class matching matrix  $M$*   
 165 *is given by:*

$$WV_M = \frac{1}{nT} \sum_{k=1}^K \sum_{l=1}^{n_k} \sum_{i=1}^T (x_i^l - \sum_{l'=1}^{n_k} \sum_{i'=1}^T m_{ii'}^{l'l'} x_{i'}^{l'})^2$$

166 with

$$M^{ll'} = \begin{cases} \mathbf{I} & \text{if } l = l' \\ \neq \mathbf{0} & \text{if } y_l = y_{l'} \text{ and } l \neq l' \\ \mathbf{0} & \text{if } y_l \neq y_{l'} \end{cases} \quad (7)$$

167 where  $\mathbf{I}$  and  $\mathbf{0}$  are the  $(T \times T)$  identity and zero matrices, respectively.

168 The general setting for the blocks  $M^{ll'}$  of the intra-class matching  $M$  is  
 169 based on three considerations: (a) the Euclidean alignment ( $M^{ll} = \mathbf{I}$ ) linking  
 170 each time series to itself ensures a variance of zero when comparing a time series  
 171 with itself, (b) time series within the same class should be connected, while (c)  
 172 time series of different classes are not connected, as they do not contribute to  
 173 the within variance.

174 Similarly, we have:

175 **Definition 3.** The **between variance** with an inter-class matching matrix  $M$   
 176 is given by:

$$BV_M = \frac{1}{nT} \sum_{k=1}^K \sum_{l=1}^{n_k} \sum_{i=1}^T (x_i^l - (m_{ii}^l x_i^l + \sum_{k' \neq k} \sum_{l'=1}^{n_{k'}} \sum_{i'=1}^T m_{ii'}^{ll'} x_{i'}^{l'}))^2$$

177 with

$$M^{ll'} = \begin{cases} \mathbf{I} & \text{if } l = l' \\ \mathbf{0} & \text{if } y_l = y_{l'} \text{ and } l \neq l' \\ \neq \mathbf{0} & \text{if } y_l \neq y_{l'} \end{cases} \quad (8)$$

178 where  $\mathbf{I}$  and  $\mathbf{0}$  are the  $(T \times T)$  identity and zero matrices, respectively.

179 The setting of the inter-class matching  $M$  is symmetric with respect to the  
 180 preceding one, matching between time series of the same class being forbidden,  
 181 while matching between time series of different classes is taken into account.

182  
 183 As one can note, the matching matrix  $M$  plays a crucial role in the definition  
 184 of the within and between variances. The main issue for time series classification  
 185 is therefore to learn a discriminative matching that highlights shared features  
 186 within classes and distinctive ones between classes. To do so, we look for the  
 187 matching matrix  $M$ , under the general settings given in Eqs. (7) and (8), that  
 188 minimizes the within variance and maximizes the between variance. We present  
 189 an efficient way to do this in the following section.

### 190 3. Learning discriminative matchings

191 We present here an efficient method to learn the matching matrix  $M$ , so  
 192 as to connect time series based on their discriminative features. The proposed  
 193 approach consists of two successive phases. In the first phase, the intra-class  
 194 matching is learned to minimize the within variance. The learned intra-class  
 195 matching reveals time series connections based on class-specific characteristics.  
 196 In the second phase, the learned intra-class matching is refined to maximize the  
 197 between variance.

198 *3.1. Learning the intra-class matching*

199 We are interested in inferring commonly shared structure within classes,  
 200 that is in identifying the set of time stamps  $i'$  connected to each time stamp  $i$   
 201 regardless of their weights.

202  
 203 Thus, the problem of learning the intra-class matching matrix  $M$  to minimize  
 204 the within variance, i.e. the quantity  $WV_M$  of Definition 2, can be formulated  
 205 as the following constrained optimization problem:

$$\left\{ \begin{array}{l} \arg \min_M \quad WV_M \\ \text{subject to: } \forall k \in \{1, \dots, K\}, \forall (l, l') \in C_k, \forall (i, i') \in [1, T]^2 : \\ \quad m_{ii}^l > 0 \text{ and } m_{ii'}^l = 0 \text{ for } i \neq i' \\ \quad \sum_{i'=1}^T m_{ii'}^{l'} > 0, \\ \quad \sum_{l'=1}^{n_k} \sum_{i'=1}^T m_{ii'}^{l'} = 1 \\ \quad \text{if } m_{ii'}^{l'} \neq 0, m_{ii'}^{l'} = m_{ii}^l \end{array} \right. \quad (9)$$

206  
 207 The first three constraints are dictated by the variance/covariance definition.  
 208 More precisely, the first constraint ensues from the first setting ( $M^l = \mathbf{I}$ ) given  
 209 in Eq. (7). The second constraint guarantees the second setting ( $M^{l'} \neq \mathbf{0}$ ) by  
 210 imposing that each time stamp  $i$  of  $S_l$  be connected to at least one time  $i'$  of  $S_{l'}$ .  
 211 The third one corresponds to a row normalization of  $M$  involved in the centering  
 212 process of  $WV_M$  (Definition 2). The last constraint determines the linkage struc-  
 213 ture to be extracted, namely, equally weighted links in the neighborhood of  $i$ , as  
 214 the interest is to reveal time stamps  $i'$  connected to  $i$ , regardless of their weights.

215  
 216 The fourth constraint renders the problem discrete and standard gradient  
 217 approaches are inappropriate. Furthermore, an exhaustive search is in prac-  
 218 tice unfeasible, because the number of configurations is  $(2^T - 1)^{\sum_k (n_k - 1)n_k T} \simeq$   
 219  $2^{\sum_k n_k^2 T^2}$ . We introduce here an efficient approach that iteratively evaluates the  
 220 contribution of each linked observation  $(i, i')$  to the within variance; the weights  
 221  $m_{ii'}^{l'}$  are then penalized for all links  $(i, i')$  that significantly increase the within  
 222 variance. This process, called *TrainIntraMatch*, is described in Algorithm 1 and  
 223 involves the following steps.

224 *Step1: Initialization* A complete linkage is used to initialize the intra-class  
 225 matching matrix  $M$ , to ensure that all possible matchings are considered  
 226 and that no *a priori* constraints on the type of matching one should look for  
 227 are introduced. Furthermore, it satisfies the constraints given in Eq.(9).

$$M^{l'} = \begin{cases} \mathbf{I} & \text{if } l = l' \\ \frac{1}{T} \mathbf{U} & \text{if } y_l = y_{l'} \text{ and } l \neq l' \\ \mathbf{0} & \text{if } y_l \neq y_{l'} \end{cases} \quad (10)$$

228



---

**Algorithm 1** *TrainIntraMatch*( $X, \alpha$ )

---

```

M = complete intra-class matching Step 1
repeat
  LinkRemoved = false
  for all (l, l') with yl = yl' and l ≠ l' do
    for all (i, i') ∈ [1, T] × [1, T] do
      Cii'll' evaluation with Eq. (11) Step 2
    end for
  end for
  for all (i, l) ∈ [1, T] × [1, n] do
    Link = arg maxi', l' (Cii'll') satisfying Eq. (13) Step 3
    if Link ≠ ∅ then
      Remove Link (mi, i'l, l' = 0) and
      Update weights with Eq. (12)
      LinkRemoved = true
    end if
  end for
until ¬LinkRemoved Step 4
return (MIntra = M)

```

---

229 *Step 2: Computing link contributions* We define the contribution  $C_{i_1 i_2}^{l_1 l_2}$  of the  
230 link  $(i_1, i_2)$  between  $S_{l_1}$  and  $S_{l_2}$  ( $y_{l_1} = y_{l_2}$ ) as the induced variation on  
231 the within variance after the link  $(i_1, i_2)$  has been removed:

$$C_{i_1 i_2}^{l_1 l_2} = WV_M - WV_{M \setminus (i_1, i_2, l_1, l_2)} \quad (11)$$

232 where  $M \setminus (i_1, i_2, l_1, l_2)$  denotes the matrix obtained from  $M$  by setting  
233  $m_{i_1 i_2}^{l_1 l_2}$  to 0 and re-normalizing its  $i_1^{th}$  row:

$$m_{i_1 i'}^{l_1 l'} \leftarrow \frac{m_{i_1 i'}^{l_1 l'}}{1 - m_{i_1 i_2}^{l_1 l_2}} \quad (12)$$

234 The evaluated contributions reveal two types of links: the links of positive  
235 contribution  $C_{ii'}^{ll'} > 0$  that decrease the within variance if removed, and the  
236 links of negative contribution  $C_{ii'}^{ll'} < 0$  that increase the within variance if  
237 removed.

238 *Step 3: Link deletion* The deletion of a link with positive contribution ensures  
239 that the within variance will decrease. Because of the renormalization  
240 given in Eq.(12), the second and third constraints given in Eq.(9) are  
241 satisfied. However, one should not remove a link if its deletion violates  
242 the fourth constraint. In addition, if all links within a row have a negligible  
243 contribution to the variance, one can dispense with removing them in order  
244 to (a) avoid overtraining and (b) speed up the process. Thus, a link  $(i, i')$   
245 between  $S_l$  and  $S_{l'}$  is deleted if it satisfies:

$$C_{ii'}^{ll'} > \alpha \cdot WV_{M_1} \quad \text{and} \quad \sum_{i''=1, (i'' \neq i')}^T m_{ii''}^{ll'} > 0 \quad (13)$$

246

247

248

249

where  $\alpha \in [0, 1]$  and  $WV_{M_1}$  is the initial within variance. When  $\alpha = 0$ , all links with positive contributions are deleted as long as this deletion does not violate the constraints.

250

251

252

253

254

Because the normalization in Eq.(12) performed after the deletion of  $(i_1, i_2)$  impacts only the weights of the  $i_1^{th}$  row, deleting a single link per row at each iteration of the process guarantees that the global within variance will decrease. Thus, at each iteration one can simply delete the link on each row of maximal contribution compliant with Eq.(13).

255

256

*Step 4: Stopping the learning process.* The algorithm iterates steps 2, 3 and 4 until there are no more links satisfying the conditions specified in Eq.(13).

257

258

259

260

From the learned intra-class matching obtained at step 4, noted  $M_{Intra}$ , one may induce for each time series  $S_l$  one intra-block  $M_{Intra}^l$  to indicate the characteristic linkage between  $S_l$  and time series of the same class. This intra-block is obtained by summing the block matrices learned for  $S_l$ , as follows:

$$M_{Intra}^l = \sum_{l' \in \{1, \dots, n_k\}} M_{Intra}^{l'} \quad (14)$$

261

262

263

264

265

Note that the row normalization of  $M_{Intra}$  assures the normalization of  $M_{Intra}^l$ . Furthermore, post-pruning can be carried out on  $M_{Intra}^l$  by setting to 0 all the weights lower than the initial uniform weighting (weight  $< \frac{1}{T}$ ), assumed not significant for classes characterization.

266

### 3.2. Learning the inter-class matching

267

268

269

270

271

272

273

274

The goal of this second phase is to refine the highlighted connections in  $M_{Intra}$  (i.e., that connects shared features within classes) to capture the links that are additionally differentiating classes. For this, we refer to a similar algorithm called *TrainInterMatch*, where the inter-class matching is initialized with  $M_{Intra}$ , then trained to maximize the between variance. Similarly, the problem of learning the inter-class matching matrix  $M$  to maximize the between variance, i.e. the quantity  $BV_M$  of Definition 3, can be formulated as the following constrained optimization problem:

$$\left\{ \begin{array}{l} \arg \max_M \quad BV_M \\ \text{subject to:} \quad \forall k \in \{1, \dots, K\}, \forall l \in C_k, \forall l' \notin C_k, \forall (i, i') \in [1, T]^2 : \\ \quad m_{ii}^{ll} > 0 \text{ and } m_{ii'}^{ll} = 0 \text{ for } i \neq i' \\ \quad \sum_{i'=1}^T m_{ii'}^{ll'} > 0, \\ \quad m_{ii}^{ll} + \sum_{k' \neq k} \sum_{l'=1}^{n_k} \sum_{i'=1}^T m_{ii'}^{ll'} = 1 \\ \quad \text{if } m_{ii'}^{ll'} \neq 0, m_{ii'}^{ll'} = m_{ii}^{ll} \end{array} \right. \quad (15)$$

275

276

277

The first constraint derives from the first setting ( $M^{ll} = \mathbf{I}$ ) given in Eq. (8). The second constraint guarantees the third setting ( $M^{ll'} \neq \mathbf{0}$ ) of Eq. (8) by

278 imposing that each time stamp  $i$  of  $S_l$  should be connected to at least one time  
 279  $i'$  of  $S_{l'}$ . The third one corresponds to row normalization of  $M$  involved in the  
 280 centering process of  $BV_M$  (Definition 3), and the last condition ensures equally  
 281 weighted links in the neighborhood of  $i$ .

282  
 283 As for the within variance minimization problem, the fourth constraint  
 284 makes the problem discrete and standard gradient approaches are inappropriate.  
 285 In addition, an exhaustive search is unfeasible as the number of configurations  
 286 is  $2^{n^2 T^2}$ . Thus, we adopt the same approach, which consists in iteratively evalu-  
 287 ating the contribution of each linked observations  $(i, i')$  to the between variance;  
 288 the weights  $m_{ii'}^{ll'}$  are then penalized for all links  $(i, i')$  significantly decreasing  
 289 the between variance. We now briefly describe the main steps of the *TrainIn-*  
 290 *terMatch* algorithm 2.

---

**Algorithm 2** *TrainInterMatch*( $X, \alpha$ )

---

```

M = defined from  $M_{Intra}$  with Eq. (16) Step 1
repeat
   $LinkRemoved = false$ 
  for all  $(l, l')$  with  $y_l \neq y_{l'}$  do
    for all  $(i, i') \in [1, T] \times [1, T]$  do
       $C_{ii'}^{ll'}$  evaluation with Eq. (17) Step 2
    end for
  end for
  for all  $(i, l) \in [1, T] \times [1, n]$  do
     $Link = \arg \min_{i', l'} (C_{ii'}^{ll'})$  satisfying Eq. (19) Step 3
    if  $Link \neq \emptyset$  then
      Remove  $Link$  ( $m_{i, i'}^{l, l'} = 0$ ) and
      Update weights with Eq. (18)
       $LinkRemoved = true$ 
    end if
  end for
until  $\neg LinkRemoved$  Step 4
return ( $M_{Intra} = M$ )

```

---

291 *Step1: Initialization* The inter-class matching matrix  $M$  is initialized as follows:

$$M^{ll'} = \begin{cases} \mathbf{I} & \text{if } l = l' \\ \mathbf{0} & \text{if } y_l = y_{l'} \text{ and } l \neq l' \\ M_{Intra}^l & \text{if } y_l = k \neq y_{l'} \end{cases} \quad (16)$$

292  
 293 where  $M^{ll'}$  is initialized with the aggregated intra block  $M_{Intra}^l$  of  $S_l$   
 294 according to Eq.(14). The row normalized initial matrix is in agreement  
 295 with the constraints given in Eq.(15).

296 *Step 2: Computing link contributions* We define the contribution  $C_{i_1 i_2}^{l_1 l_2}$  of the  
 297 link  $(i_1, i_2)$  between  $S_{l_1}$  and  $S_{l_2}$  ( $y_{l_1} \neq y_{l_2}$ ) to be the variation induced on  
 298 the between variance after the link  $(i_1, i_2)$  has been removed:

$$C_{i_1 i_2}^{l_1 l_2} = BV_M - BV_{M \setminus (i_1, i_2, l_1, l_2)} \quad (17)$$

299 where  $M \setminus (i_1, i_2, l_1, l_2)$  denotes the matrix obtained from  $M$  by setting  
 300  $m_{i_1 i_2}^{l_1 l_2}$  to 0 and re-normalizing its  $i_1^{th}$  row:

$$m_{i_1 i'}^{l_1 l'} \leftarrow \frac{m_{i_1 i'}^{l_1 l'}}{1 - m_{i_1 i_2}^{l_1 l_2}} \quad (18)$$

301 The evaluated contributions reveal two types of links: those making a posi-  
 302 tive contribution  $C_{ii'}^{ll'} > 0$  that induce a decrease of the between variance  
 303 if removed and those making a negative contribution  $C_{ii'}^{ll'} < 0$  that cause  
 304 an increase of the between variance if removed.

305 *Step 3: Link deletion* The deletion of a link with a negative contribution ensures  
 306 that the between variance will increase. The second and third constraints  
 307 given in Eq.(15) are preserved by the row normalization given in Eq.(18).  
 308 However, one should not remove a link if its deletion would violate the  
 309 fourth constraint. In addition, if all one can dispense with removing links  
 310 with negligible contributions in order to (a) avoid overtraining and (b)  
 311 speed up the process. Thus, a link  $(i, i')$  between  $S_l$  and  $S_{l'}$  is deleted if  
 312 it satisfies:

$$C_{ii'}^{ll'} < -\alpha \cdot BV_{M_1} \quad \text{and} \quad \sum_{i''=1, (i'' \neq i')}^T m_{ii''}^{ll'} > 0 \quad (19)$$

313

314 where  $\alpha \in [0, 1]$  and  $BV_{M_1}$  is the initial between variance. When  $\alpha = 0$ ,  
 315 all links with negative contributions are deleted as long as this deletion  
 316 does not violate the constraints. As argued before, the convergence of the  
 317 learning process is ensured by removing, at each iteration, one link at most  
 318 for each row of  $M$ , the one exhibiting the minimal negative contribution  
 319 compliant with Eq.(19).

320 *Step 4: Stopping the learning process.* The algorithm iterates steps 2, 3 and 4  
 321 until there are no more links satisfying the conditions specified in Eq.(19).  
 322 Let  $M_*$  be the learned intra-class matching.

### 323 3.3. Convergence and complexity of the learning process

324 As noted above, the process retained guarantees that each time one deletes  
 325 a link compliant with Eq.(13) in *TrainIntraMatch* (respectively to Eq.(19) in  
 326 *TrainInterMatch*) and renormalizes  $M$  according to Eq.(12) (respectively Eq.(18)),  
 327 the variance decreases (respectively increases) while the constraints are still sat-  
 328 isfied. This process thus converges towards a matching matrix  $M$ , yielding a  
 329 lower within class (respectively higher between class) variance than the original

330 one. Furthermore, if  $\alpha$  equals 0, the set of links obtained is minimal in the sense  
 331 that any deletion of a link from this set will lead to an increase (respectively  
 332 decrease) in the variance.

333 In terms of complexity, the dominating factor in the above process is the  
 334 computation of the contributions of each link. This contribution  $C_{i_1 i_2}^{l_1 l_2}$  can be  
 335 expressed as the difference between the centered values of  $x_{i_1}^{l_1}$  before and after  
 336 the deletion of  $(i_1, i_2)$ :

$$\begin{aligned}
 C_{i_1 i_2}^{l_1 l_2} &= \frac{1}{nT} (x_{i_1}^{l_1} - \sum_{l'=1}^{n_k} \sum_{i'=1}^T m_{i_1 i'}^{l_1 l'} x_{i'}^{l'})^2 - \\
 &\quad \frac{1}{nT} (x_{i_1}^{l_1} - \sum_{l'=1}^{n_k} \sum_{i'=1}^T \frac{m_{i_1 i'}^{l_1 l'}}{1 - m_{i_1 i_2}^{l_1 l_2}} x_{i'}^{l'})^2
 \end{aligned} \tag{20}$$

337 for the *TrainIntraMatch*, and:

$$\begin{aligned}
 C_{i_1 i_2}^{l_1 l_2} &= \frac{1}{nT} (x_{i_1}^{l_1} - (m_{i_1 i}^{l_1 l} + \sum_{k' \neq k} \sum_{l'=1}^{n_{k'}} \sum_{i'=1}^T m_{i_1 i'}^{l_1 l'} x_{i'}^{l'}))^2 - \\
 &\quad \frac{1}{nT} (x_{i_1}^{l_1} - (\frac{m_{i_1 i_1}^{l_1 l_1}}{1 - m_{i_1 i_2}^{l_1 l_2}} x_i^l + \sum_{k' \neq k} \sum_{l'=1}^{n_{k'}} \sum_{i'=1}^T \frac{m_{i_1 i'}^{l_1 l'}}{1 - m_{i_1 i_2}^{l_1 l_2}} x_{i'}^{l'}))^2
 \end{aligned} \tag{21}$$

338

339 for the *TrainInterMatch*.

340

341 The complexity of the *TrainIntraMatch* algorithm thus amounts to  $O(\sum_{k=1}^K I_k n_k^2 T^2)$   
 342 where  $I_k$  corresponds to the number of times the process has affected elements  
 343 of class  $k$ . In the worst case, for each class  $k$ ,  $I_k = (n_k - 1)(T - 1)$ , and the  
 344 overall complexity amounts to  $O(\sum_{k=1}^K n_k^3 T^3)$ . Similarly, the complexity of the  
 345 *TrainInterMatch* algorithm is about  $O(\sum_{k=1}^K I_k (n - n_k) n_k T^2)$  with, in the worst  
 346 case,  $I_k = (n - n_k)(T - 1)$ , and an overall complexity of  $O(\sum_{k=1}^K (n - n_k)^2 n_k T^3) \simeq$   
 347  $O(n^3 T^3)$ .

348

349 We now turn to the application of the learned matching matrix to time series  
 350 classification.

#### 351 4. Time series classification based on the learned matching

352 Our aim here is to present a way of using learned discriminative matching  
 353 to locally weight time series for  $k$ -nearest neighbor classification. The purpose  
 354 of the proposed weighting is to restrict the time series comparison to the dis-  
 355 criminant (characteristic and differential) features. Let  $M_*$  be the discrimina-  
 356 tive matching learned by the *TrainIntraMatch* and *TrainInterMatch* algorithms,  
 357 where discriminant linkages are highly weighted. For each  $S_l$  of the training  
 358 sample, we define its discriminative matching  $M_*^l$  as the average of the learned  
 359 matrices  $M_*^{l'}$  ( $y_{l'} \neq y_l = k$ ):

$$M_*^{l\cdot} = \frac{1}{(n - n_k)T} \sum_{l'} M_*^{l'}$$

360

361 In  $k$ -nearest neighbor classification, one can compare a new time series  $S_{test}$   
 362 to a sample series  $S_l$  of  $C_k$  based on its learned discriminative matching. How-  
 363 ever, as discussed above, discriminative features of time series of  $C_k$  may appear  
 364 at different time stamps, according to a delay  $r$ . Thus, to evaluate the proximity  
 365 to  $S_l$  one should consider, in addition, the delay inherent in  $S_{test}$ . This can be  
 366 achieved by looking for the delay  $r$  that leads to the minimal distance between  
 367  $S_{test}$  and  $S_l$ , as proposed in the following locally weighted proximity measure:

$$D_l(S_l, S_{test}) = \min_{r \in \{0, \dots, T-1\}} \left( \sum_{|i-i'| \leq r; (i, i') \in [1, T]^2} \frac{m_{ii'}^{l\cdot}}{\sum_{|i-i'| \leq r} m_{ii'}^{l\cdot}} (x_i^l - x_{i'}^{test})^2 \right) \quad (22)$$

368

369 where  $r$  corresponds to the Sakoe-Chiba band width [16]. Note that for  $r = 0$ ,  $D_l$   
 370 defines a locally weighted Euclidean distance involving the diagonal weights  $m_{ii}^{l\cdot}$ ,  
 371 whereas for  $r = T - 1$  the time series are compared according to the complete  
 372 learned discriminative matching. The proposed  $D_l$  defines a dissimilarity index  
 373 satisfying the positivity, symmetry and coincidence axioms. The  $D_l$  is simply  
 374 noted  $D$  in the following.

## 375 5. Experiments

### 376 5.1. Description of the datasets

377 To motivate our approach we first considered BME and UMD, two synthetic  
 378 challenging datasets composed of time series that are dissimilar within classes  
 379 and slightly similar between classes. BME consists of three classes *Begin*, *Mid-*  
 380 *dle*, and *End* (Figure 1). In the *Begin* (respectively the *End*) class, the time  
 381 series share a common signature defined by a small bell arising at the initial  
 382 (respectively final) period. The overall behavior may be distinctive within these  
 383 classes depending on whether the large bell is up or down positioned. Further-  
 384 more, time series of the *Begin* and the *End* classes composed of an up-positioned  
 385 large bell are quite similar to the *Middle* class time series.

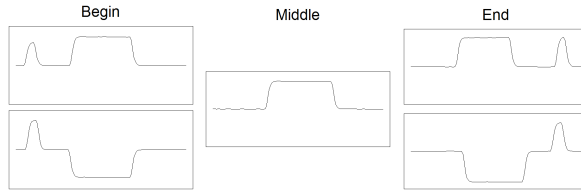


Figure 1: Distinctive behaviors within BME classes: *Begin*, *Middle*, and *End*

386 The second dataset UMD, composed similarly of three classes *Up*, *Middle*,  
 387 and *Down*, introduces more complexity, with a local shared signature (i.e. a

388 small bell) occurring at different time stamps, as illustrated in Figure 2.  
 389

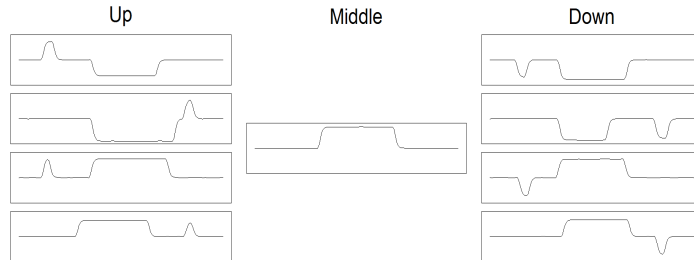


Figure 2: Distinctive behaviors within UMD classes: UP, MIDDLE, and DOWN

390 The reason for considering these synthetic datasets is to show, through a  
 391 challenging but easily identifiable discriminant features, which connections are  
 392 selected by the *TrainIntraMatch* algorithm and how they are refined after the  
 393 *TrainInterMatch* algorithm captures the discriminative linkage.

394  
 395 The proposed approach is thus motivated by a real application that aims to  
 396 analyze the electrical power consumption of customers, to adequately meet con-  
 397 sumer demands. We use two datasets *CONSLEVEL* and *CONSSEASON* obtained  
 398 from a public database<sup>1</sup> providing the electric power consumption recorded in a  
 399 personal home over almost one year (349 days). Each time series consists of 144  
 400 measurements that give the power consumption of one day with a 10 minute  
 401 sampling rate.

402  
 403 *CONSLEVEL* divides the 349 time series into two classes (*Low* and *High*) de-  
 404 pending on whether the average electric power during the peak demand period  
 405 (6:00pm-8:00pm) is lower or greater than the annual average consumption dur-  
 406 ing that period. Figure 3 shows the pattern of electric consumption within the  
 407 *CONSLEVEL* classes; the red frames delineate the time interval [108,120], which  
 408 corresponds to the peak period (6:00pm-8:00pm).

409  
 410 On the other hand, *CONSSEASON* splits the 349 time series into two sea-  
 411 son classes (*Warm* and *Cold*) depending on whether the power consumption is  
 412 recorded during the warm (from April to September) or cold (from October to  
 413 March) seasons (Figure 4). Note that the electric power consumption profiles  
 414 differ markedly within classes in both datasets.

415  
 416 The goal of the proposed approach applied to the *CONSLEVEL* and *CON-*  
 417 *SEASON* datasets is: to localize the periods that characterize the daily power

<sup>1</sup>These data are available at <http://bilab.enst.fr/wakka.php?wiki=HomeLoadCurve>, and analyzed in [8]

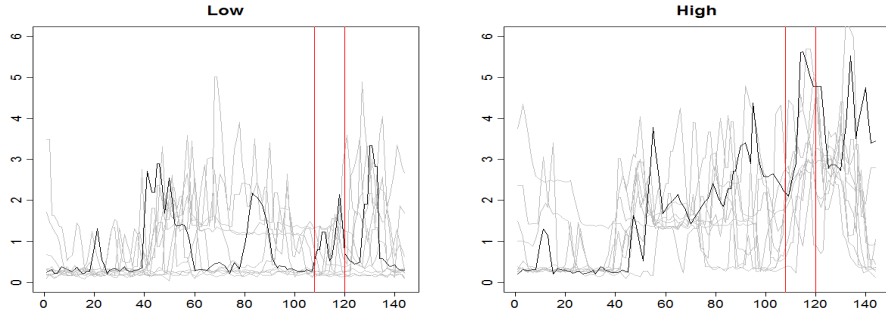


Figure 3: The electrical power consumption within the *Low* and *High* classes of CONSLEVEL dataset.

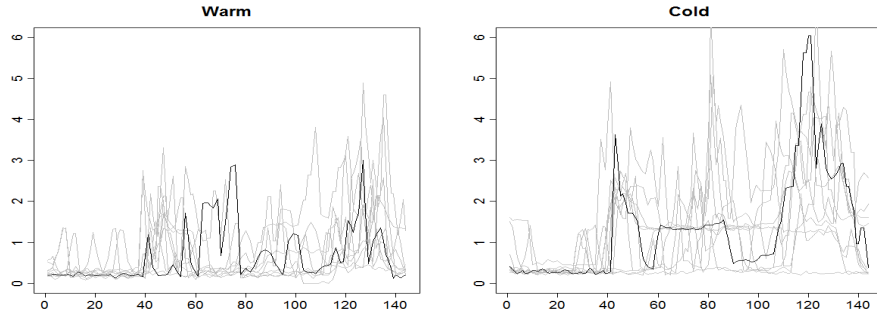


Figure 4: The electrical power consumption within the *Warm* and *Cold* classes of CONSEASON dataset.

418 consumption of each class, to highlight periods that differentiate the power consumption of different classes, and to classify new power consumption based on the learned discriminative features. For instance, for the CONSLEVEL dataset, a classification based on the discriminative periods prior to the time interval 6:00pm-8:00pm can help forecast the consumer demands during the peak period.

424

425 In addition to the above mentioned datasets, we have used a standard dataset on character trajectories TRAJ [1], where time series share a quite similar global behavior within classes (20 classes of 50 time series each). The goal of this latter dataset is to verify whether the proposed approach can recover standard time series structures within classes or not.

430

## 431 5.2. Results and discussion

432 The algorithms *TrainIntraMatch* and *TrainInterMatch* are applied to the above datasets with  $\alpha = 0.5\%$ . As an example, let us first illustrate, for the

433



434 BME dataset, the progression of the within and between variances during the  
 435 learning processes (Figure 5). The clearly monotonically decreasing (respec-  
 436 tively increasing) behavior of the within (respectively between) class variance,  
 437 which ends at a plateau, assesses: a) the pertinence of the conducted links pe-  
 438 nalization to minimize the within variance and maximize the between variance,  
 439 b) the convergence of the proposed algorithms.  
 440

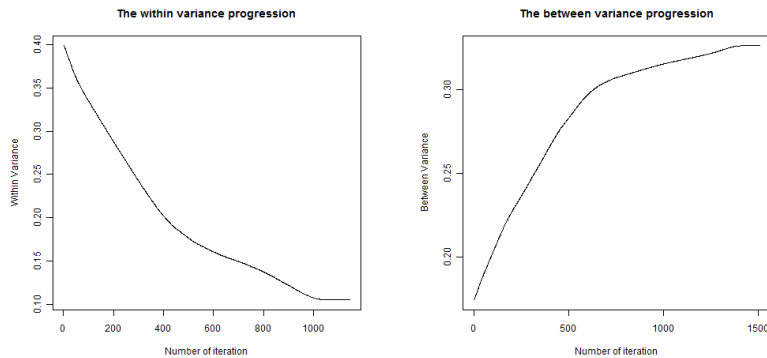


Figure 5: The within and between variance progression for BME dataset.

441 Figure 6 displays, for a time series from the *Middle* class (UMD dataset),  
 442 its learned intra-class and inter-class blocks; the bright cells indicate highly  
 443 weighted links. The intra-class block (Figure 6 left) reveals the characteris-  
 444 tic matching between the given time series (in a row) and time series of the  
 445 same class (in a column). The determined structure, from the intra-class block,  
 446 shows a strong linkage between, on the one hand, the central large bells rep-  
 447 resented by the central light square, and on the other hand, the initial and  
 448 final plateaus. From the corresponding inter-block (Figure 6 right), we can see  
 449 that connections that are characteristic (captured in the intra-class block) but  
 450 not differential have been removed, while those that are both characteristic and  
 451 differential are retained and reinforced. In particular, note that the weights of  
 452 the central square region are lowered but not set to zero, because this region  
 453 discriminates time series of the *Up* and *Down* classes that are composed of a  
 454 down positioned large bell. However, the linkage between the initial and final  
 455 plateaus has been drastically reduced to small discriminative regions (i.e., highly  
 456 weighted), corresponding to the periods where the small bells may arise within  
 457 the *Up* and *Down* classes.  
 458

459 For CONSLEVEL, similarly, we can see in Figure 7 the learned intra-class and  
 460 inter-class blocks for a given time series of the *Low* class. The intra-class block  
 461 reveals a checkerboard structure, indicating that the electric power consump-  
 462 tion within the *Low* class alternates, in a daily period, between a low and a  
 463 moderately high consumption. The corresponding inter-class block shows the

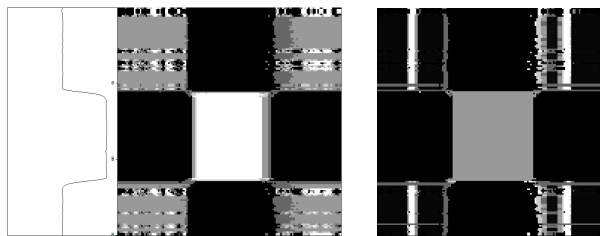


Figure 6: The learned intra (left) and inter (right) class matching for UMD dataset.

464 discriminative matching between the considered *Low* class time series and time  
 465 series of the *High* class (on column). This block displays many discriminative  
 466 regions; for example, it shows that the power consumption within the *High* class  
 467 within the period underlined in red (prior to 6:00pm-8:00pm) is especially im-  
 468 portant in predicting the consumption during the peak period.

469

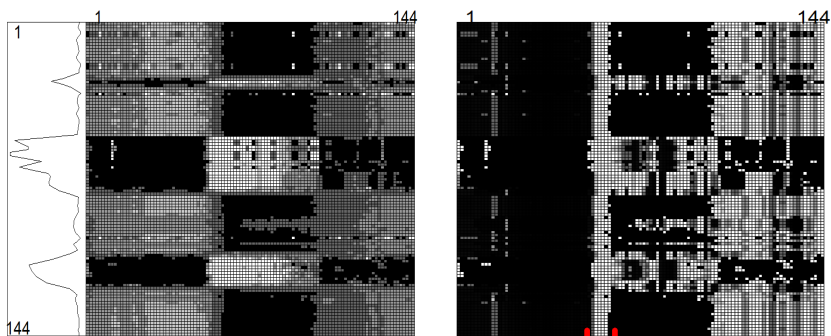


Figure 7: The learned intra and inter class matching for a *Low* class time series.

470 The learned discriminative matching is then used for the locally weighted  
 471 time series metric  $D$  given in Eq.(22). The relevance of the proposed approach  
 472 and of the induced metric are then studied through a  $k$ -nearest neighbor clas-  
 473 sification for  $k = 1, 3, 5, 7$  and through a leave-one-out protocol. The results  
 474 obtained are compared to two baselines: the Euclidean  $DE$  and dynamic time  
 475 warping  $DTW$  distances (Table 1).

476

477 The misclassification error rates obtained in Table 1 show the efficiency of  
 478 the proposed locally weighted metric  $D$  in discriminating between complex time  
 479 series classes, compared to standard metrics for time series. In particular, one  
 480 can note that for all datasets but TRAJ, the best results (in bold) are obtained  
 481 with  $D$ . For TRAJ, the three metrics lead to comparable results suggesting that  
 482 the Euclidean alignment is an appropriate matching for this dataset. In Figure  
 483 8, we can see that the learned discriminative matching, for example, for "c",  
 484 "o", "i", "e", "u" and "a" characters is close to the Euclidean one, which shows

Table 1:  $k$ -Nearest Neighbor classification error rates

	$k$	D	DE	DTW
BME	1	<b>0.032</b>	0.165	0.130
	3	0.034	0.208	0.132
	5	0.062	0.234	0.136
	7	0.079	0.297	0.191
UMD	1	<b>0.055</b>	0.173	0.121
	3	0.111	0.333	0.177
	5	0.173	0.343	0.225
	7	0.222	0.378	0.274
CONSLEVEL	1	0.056	0.306	0.289
	3	0.044	0.267	0.261
	5	0.028	0.233	0.239
	7	<b>0.017</b>	0.233	0.233
CONSSEASON	1	<b>0.094</b>	0.239	0.283
	3	0.128	0.228	0.311
	5	0.205	0.200	0.300
	7	0.111	0.222	0.306
TRAJ	1	0.014	<b>0.012</b>	0.019
	3	0.018	0.017	0.022
	5	0.022	0.021	0.028
	7	0.019	0.021	0.026

485 the ability of the proposed approach to recover standard time series alignments.  
486 In addition, one can see that for nearly all datasets the best performances are  
487 obtained for  $k = 1$ . For CONSLEVEL, a slight improvement is reached for  $k = 7$ ,  
488 indicating a great clusters overlap for this dataset.

489

490 Finally, to complete the above observed results and to assess the discrimi-  
491 native strength of the proposed metric with respect to the baselines retained,  
492 classical multidimensional-scaling [3] is used to visualize time series proximities  
493 induced by the studied metrics for both CONSLEVEL and CONSSEASON datasets  
494 (Figure 9). These figures corroborate the discriminative power of the proposed  
495 metric, because they display compact and well isolated classes, whereas baselines  
496 present a great overlap between the different classes.

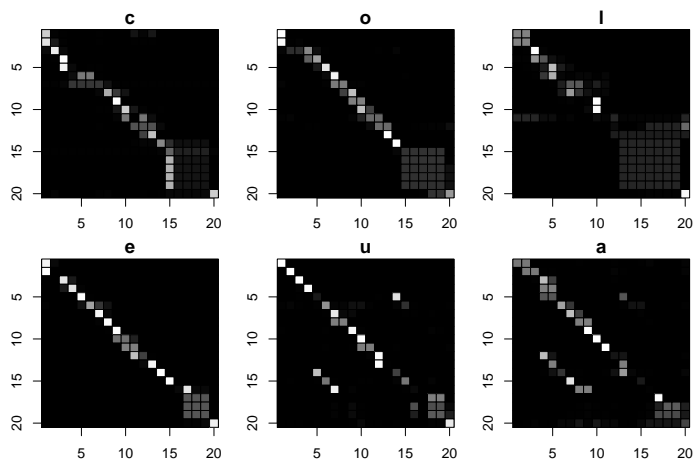


Figure 8: The learned discriminative matching for the characters "c", "o", "l", "e", "u" and "a" of TRAJ dataset.

497 **6. Conclusion**

498 Motivated by discriminating time series that are dissimilar within classes or  
 499 nearly similar across classes, we have presented a new approach for training dis-  
 500 criminative matchings that connects time series with respect to the commonly  
 501 shared features within classes, and the most differential ones across classes. To  
 502 do so, we have first introduced a generalization of the variance to sets of time  
 503 series. The definition we have provided generalizes the standard definition of  
 504 the variance and is the first one proposed, to our knowledge, for sets of time  
 505 series. We have then introduced efficient approaches to learn matching matrices  
 506 between time series that minimize (respectively maximize) the within (respec-  
 507 tively between) class variance. Based on the learned matching, we have finally  
 508 introduced a new locally weighted metric that restricts the time series compar-  
 509 ison to discriminative features. The experiments we have conducted show the  
 510 ability of the learned matching to capture fine-grained distinctions between time  
 511 series; they also show that the metric we have introduced outperforms metrics  
 512 commonly used on time series, as the Euclidean and Dynamic Time Warping  
 513 distances, on two real datasets.

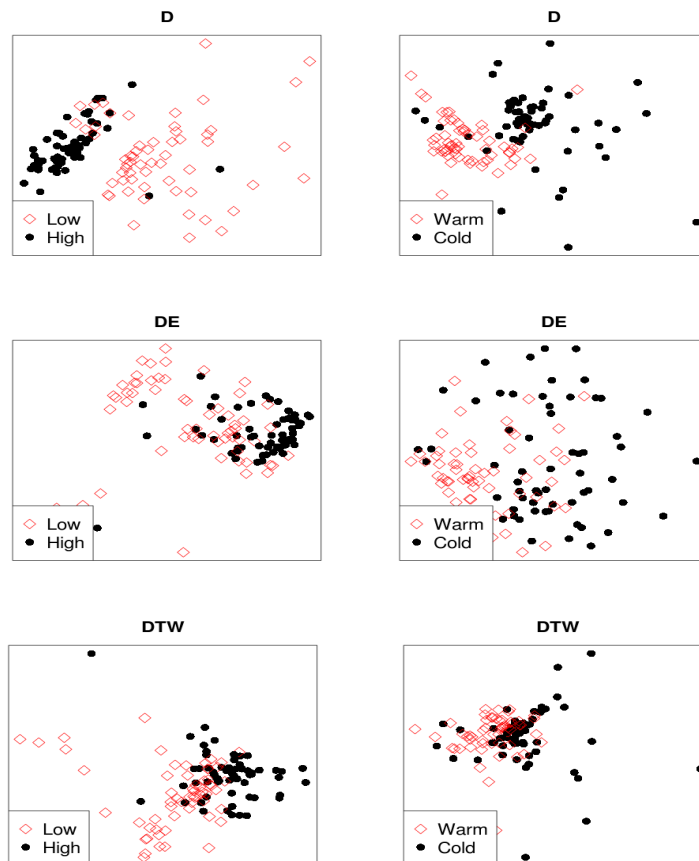


Figure 9: The time series proximities induced by D, DE and DTW for CONSLEVEL and CONSSEASON data.

## 514 References

- 515 [1] Asuncion, A., & Newman, D. (2007). Uci machine learning repository.
- 516 [2] Chopra, S., Hadsell, R., & LeCun, Y. (CA, 2005). Learning a similarity  
517 metric discriminatively, with application to face verification. In *IEEE Con-*  
518 *ference on Computer Vision and Pattern Recognition* (pp. 349–356). San  
519 Diego.
- 520 [3] Cox, T., & Cox, M. (2001). *Multidimensional Scaling*. Chapman and Hall.
- 521 [4] Douzal-Chouakria, A., & Amblard, C. (2012). Classification trees for time  
522 series. *Pattern Recognition*, 45, 1076–1091.

- 523 [5] Gaffney, S. J., & Smyth, P. (2005). Joint probabilistic curve clustering and  
524 alignment. In *Advances in Neural Information Processing Systems, 17*,  
525 473–480.
- 526 [6] Gaudin, R., & Nicoloyannis, N. (2006). An adaptable time warping distance  
527 for time series learning. In *the 5th International Conference on Machine*  
528 *Learning and Applications*. (pp. 213–218).
- 529 [7] Hastie, T., & Tibshirani, R. (1996). Discriminant adaptive nearest neigh-  
530 bor classification. *IEEE Transactions on Pattern Analysis and Machine*  
531 *Intelligence, 18*, 607–616.
- 532 [8] Hebrail, G., Huguency, B., Lechevallier, Y., & Rossi, F. (2010). Exploratory  
533 analysis of functional data via clustering and optimal segmentation. *Neu-*  
534 *rocomputing, 73*, 1125–1141.
- 535 [9] Jeong, Y., Jeong, M., & Omitaomu, O. (2011). Weighted dynamic time  
536 warping for time series classification. *Pattern Recognition, 44*, 2231–2240.
- 537 [10] Kruskal, J., & Liberman, M. (1983). *The symmetric time warping al-*  
538 *gorithm: From continuous to discrete*. In *Time Warps, String Edits and*  
539 *Macromolecules*. Addison-Wesley.
- 540 [11] Listgarten, J., Neal, R., Roweis, S., Puckrin, R., & Cutler, S. (2007).  
541 Bayesian detection of infrequent differences in sets of time series with shared  
542 structure. *Neural Information Processing Systems, 19*.
- 543 [12] Navarro, G. (2001). A guided tour to approximate string matching. *ACM*  
544 *Computing Surveys, 33*, 31–88.
- 545 [13] Paredes, R., & Vidal, E. (2006). Learning prototypes and distances: A pro-  
546 totype reduction technique based on nearest neighbor error minimization.  
547 *Pattern Recognition, 39*, 180–188.
- 548 [14] Ramsay, J., & Li, X. (1998). Curve registration. *Journal of the Royal*  
549 *Statistical Society, B*, 351–363.
- 550 [15] Ratanamahatana, C. A., & Keogh, E. (2004). Making time-series classi-  
551 fication more accurate using learned constraints. In *SIAM International*  
552 *Conference on Data Mining* (pp. 11–22.).
- 553 [16] Sakoe, H., & Chiba, S. (1978). Dynamic programming algorithm optimiza-  
554 tion for spoken word recognition. *IEEE Transactions on Acoustics, Speech,*  
555 *and Signal Processing, 26*, 43–49.
- 556 [17] Sankoff, D., & Kruskal, J. (1983). *Time warps, string edits, and macro-*  
557 *molecules: the theory and practice of sequence comparison*. Addison-  
558 Wesley.
- 559 [18] Wartenberg, D. (1985). Multivariate spatial correlation: A method for  
560 exploratory geographical analysis. *Geographical Analysis, 17*, 263–283.

- 561 [19] Xie, Y., & Wiltgen, B. (2010). Adaptive feature based dynamic time warp-  
562 ing. *International Journal of Computer Science and Network Security*, 10.
- 563 [20] Yu, D., Yu, X., Hu, Q., Liu, J., & Wu, A. (2011). Dynamic time warping  
564 constraint learning for large margin nearest neighbor classification. *Informa-*  
565 *tion Sciences*, 181, 2787–2796.