Chapter 4

Experiments

Contents

In this chapter, we evaluate the efficiency of the proposed M^{2}TML algorithm on public datasets for classification problems of univariate time series. First, we describe the datasets. Then, we detail the experimental protocol. Finally, we present and discuss the obtained results.

4.1 Description

The efficiency of the learned multi-modal and multi-scale dissimilarities D and $D_{\mathscr{H}}$ is evaluated through a 1-NN classification on 30 public datasets[1] [Keo+11]. The 1-NN classifier is used to make the results comparable with the results of the UCR time series data mining archive[2]. Time series come from several fields (simulated data, medical data, electrical data, etc.), are from variable lengths (from small ($q = 24$) to long lengths ($q = 1882$)) and the number of classes to discriminate evolves between 1 and 37 classes. Note that some of the datasets have a small number of time series in the training set ($n < 30$) and others have a large number of time series in the training set ($n > 100$). The results using standard metrics (Euclidean distance, Dynamic time warping) show both easy and challenging classifications problems, the latter being opened for improvements.

[1] PowerCons: `https://archive.ics.uci.edu/ml/datasets/Individual+household+electric+power+consumption`, BME and UMD: `http://ama.liglab.fr/~douzal/tools.html`.

[2] Note: the datasets and results are the ones before the update of August 2015

Table 4.1 gives a description of the datasets considered in the experiments and Fig. 4.1 gives the temporal representation for some of the datasets. Note that for some datasets (*e.g.*, SonyAIBO, ECG200, FaceFour, PowerConsumption), it is visually difficult to discriminate the classes using one modality (value, behavior, frequential).

Dataset	Nb. Class	Nb. Train	Nb. Test	TS length
1 ItalyPowerD	2	67	1029	24
2 CinCECGtorso	4	40	1380	1639
3 BME	3	300	1500	128
4 ECG200	2	100	100	96
5 SonyAIBOII	2	27	953	65
6 Coffee	2	28	28	286
7 ECG5Days	2	23	861	136
8 SonyAIBO	2	20	601	70
9 Adiac	37	390	391	176
10 Beef	5	30	30	470
11 Trace	4	100	100	275
12 CBF	3	30	900	128
13 CC	6	300	300	60
14 DiatomSizeReduc	4	16	306	345
15 Symbols	6	25	995	398
16 GunPoint	2	50	150	150
17 FacesUCR	14	200	2050	131
18 TwoLeadECG	2	23	1139	82
19 UMD	3	360	1440	150
20 MoteStrain	2	20	1252	84
21 Lighting2	2	60	61	637
22 OliveOil	4	30	30	570
23 FISH	7	175	175	463
24 FaceFour	4	24	88	350
25 SwedishLeaf	15	500	625	128
26 MedicalImages	10	381	760	99
27 Lighting7	7	70	73	319
28 PowerCons	2	73	292	144
29 OSULeaf	6	200	242	427
30 InlineSkate	7	100	550	1882

Table 4.1: Dataset table description providing the number of classes (Nb. Class), the number of time series for the training (Nb. Train) and the testing (Nb. Test) sets, and the length of each time series (TS length).

The results of the learned metrics D and $D_{\mathscr{H}}$ are compared to those of three *a priori* combined metrics $D_{Lin}, D_{Geom}, D_{Sig}$ (Eqs. 2.16, 2.17, 2.18) and five alternative uni-modal metrics covering:

1. The standard Euclidean distance d_A (Eq. 2.1) and Dynamic time warping[3] DTW (Eq. 2.13)
2. The behavior-based measures d_B (Eq. 2.6) and $d_{B-\mathrm{DTW}}$ its counterpart for asynchronous time series, that is d_B is evaluated once time series are synchronized using dynamic programing
3. The frequential-based metric d_F (Eq. 2.4).

[3]In this chapter, the term DTW denotes the classically value-based metric computed after an alignment of the time series obtained with the DTW algorithm with a value-based cost function.

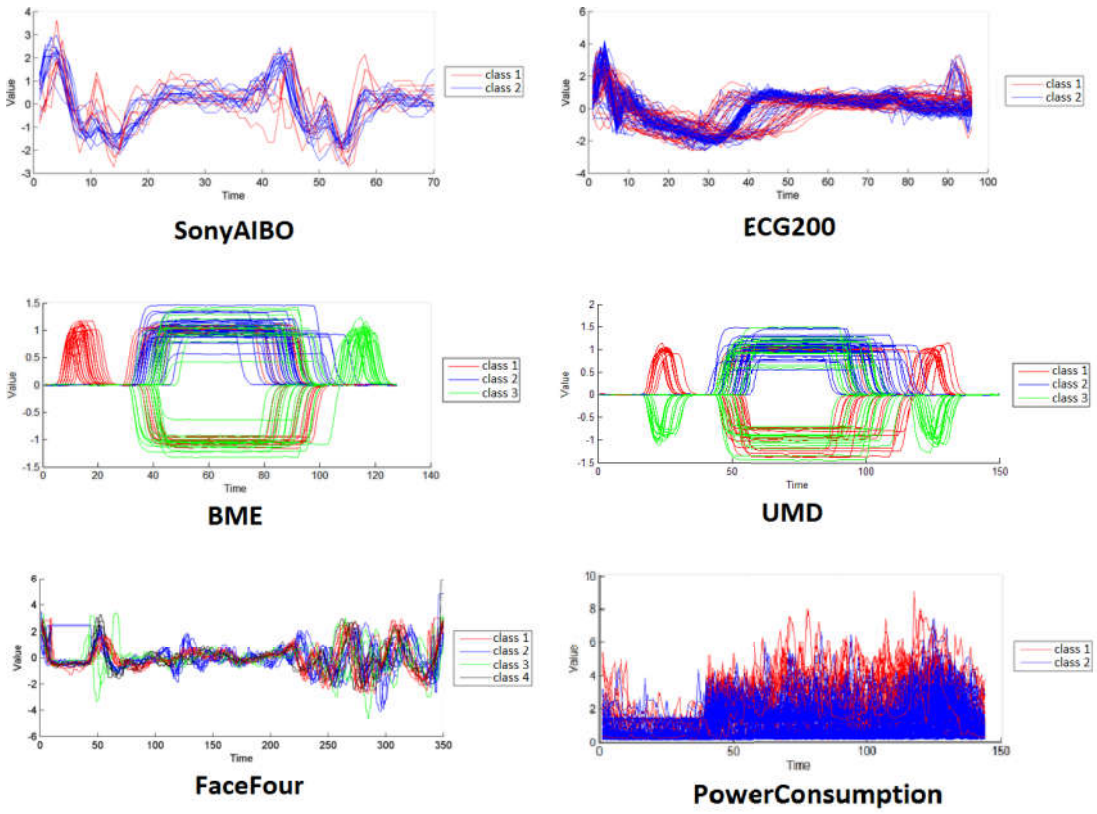


Figure 4.1: Temporal representation of some datasets (SonyAIBO, ECG200, BME, UMD, FaceFour, PowerConsumption) considered in the experiments.

Symbol	Name	Equation reference	Description
d_A	Value-based dissimilarity	Eq. 2.1	Euclidean distance
d_B	Behavior-based dissimilarity	Eq. 2.6	Behavior metric based on cort
DTW	Dynamic time warping	Eqs. 2.13 & 2.1	Euclidean distance after alignment
$d_{B-\text{DTW}}$	Behavior-based aligned dissimilarity	Eqs. 2.13 & 2.6	Behavior metric based on cort after alignment
d_F	Frequential-based dissimilarity	Eq. 2.4	Frequential metric based on Fourier transform
D_{Lin}	Linear combined metric	Eq. 2.16	Combines d_A and d_B (resp. DTW and $d_{B-\text{DTW}}$)
D_{Geom}	Geometric combined metric	Eq. 2.17	Combines d_A and d_B (resp. DTW and $d_{B-\text{DTW}}$)
D_{Sig}	Sigmoid combined metric	Eq. 2.18	Combines d_A and d_B (resp. DTW and $d_{B-\text{DTW}}$)
D	Linear learned metric	Eq. 3.49	M^2TML linear combined metric
$D_{\mathscr{H}}$	Non-linear learned metric	Eq. 3.50	M^2TML non-linear combined metric with a Gaussian kernel

Table 4.2: Considered metric in the experiments

Table 4.2 recalls briefly the considered metrics in the experiments. The *a priori* combined metrics ($D_{Lin}, D_{Geom}, D_{Sig}$) rely, on 2 log-normalized dissimilarities d_A, d_B (resp. DTW, $d_{B-\text{DTW}}$ for asynchronous time series). The alternative metrics and the *a priori* combined metrics are evaluated as usual by involving all time series elements (*i.e.*, at the global scale). For D and $D_{\mathscr{H}}$, we consider a 21-dimensional embedding space $\mathscr{E}$ that relies, for synchronous (resp. asynchronous) data, on 3 log-normalized dissimilarities d_A^s, d_B^s (resp. DTW^s, $d_{B-\text{DTW}}^s$), and d_F^s, at 7 temporal granularities $s \in \{0, ..., 6\}$ obtained by binary segmentation, described in Section 3.3.

4.2 Experimental protocol

The different metrics can be split into two categories. For those without parameters to tune (d_A, DTW), the 1−NN classifier is applied directly on the test set. For those that require to tune parameters (d_B, $d_{B-\text{DTW}}$, D_{Lin}, D_{Geom}, D_{Sig}, D, $D_{\mathscr{H}}$), we recall briefly the grid search and cross-validation procedure (Section 1.1.2). When a learning algorithm requires to tune some parameters, to avoid overfitting, the training set can be divided into two sets: a learning and a validation set. The model is learnt for each combination of parameters (grid search) on the learning set and evaluated on the validation set. The model with the lowest error on the validation set is retained. An other alternative is cross-validation, which partitions the training set into v folds, performs the learning on one subset, and validates on the $v-1$ other subsets. To take into account of variability within the data, multiple rounds of cross-validation are performed using different partitions, and the validation results are averaged over the rounds. Note that for unbalanced datasets in classification problems, it is recommended to use stratified sampling. Table 4.3 resumes the parameter ranges for each metric. We recall that the parameters retained are those that:

- **First**, minimize the average classification error on the validation set.
- **Secondly**, in the case of multiple solutions leading to equal performances, the most discriminant one is retained (*i.e.*, making closer pull pairs and far away push pairs). Precisely, it minimizes the ratio $\frac{d_{intra}}{d_{inter}}$ where d_{intra} and d_{inter} stands respectively to the mean of all intraclass and interclass distances.

As D and $D_{\mathscr{H}}$ involves several parameters to be tuned, we detail hereafter the procedure. The combined metrics D and $D_{\mathscr{H}}$ (κ as the Gaussian kernel) are learned respectively under L_1 and L_2 regularization, using LIBLINEAR and LIBSVM libraries [FCH08]; [HCL08]. The parameters are estimated on a validation set by line/grid search. A cross-validation and stratified sampling for unbalanced datasets are used. Particularly, for each couple (r, λ) $r \in \{1, 4, 10\}$ and $\lambda \in \{0, 10, 30\}$, the pairwise SVM parameters (C, α, γ) are learned by grid search as indicated in Table 4.3.

Dissimilarity	Parameter	Ranges	Description
d_B, $d_{B-\text{DTW}}$	r	$\{1, 2, 3, , \ldots, q-1\}$	Order of behavior-based metric
$D_{Lin}, D_{Geom}, D_{Sig}$	β	$\{0, 0.1, \ldots, 1\}$	Trade-off between value and behavior components
D, $D_{\mathscr{H}}$	λ	$\{0, 10, 30\}$	Strength of the '*push*' term
D, $D_{\mathscr{H}}$	r	$\{1, 4, 10\}$	Order of behavior-based metrics
D, $D_{\mathscr{H}}$	C	$\{10^{-3}, 0.5, 1, 5, 10, 20, 30, ..., 150\}$	Parameter of SVM
D, $D_{\mathscr{H}}$	α	$\{1, 2, 3\}$	Size of the $m = \alpha.k$ neighborhood
$D_{\mathscr{H}}$	γ	$\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$	Parameter of the Gaussian kernel

Table 4.3: Parameter ranges

Note that the temporal order r for the behavior-based metrics d_B is noise-dependent, typically 1 is retained for noise-free data. The parameter λ corresponds to the strength of the '*push*' term; precisely, if no, moderate or strong '*push*' is required during the training process, a λ value of 0, 10 and 30 is learned, respectively.

4.3 Results and discussion

In this section, we first present a summary table of the quantitivative results obtained in the experiment. Secondly, we present an analysis of the performances of the different metrics. Finally, we present the ability of our proposed approach M^{2}TML to extract discriminative features.

4.3.1 Results

Table 4.4 reports the 1-NN classification test errors based on uni-modal metrics (first 5 columns), on three *a priori* combined metrics ($D_{Lin}, D_{Geom}, D_{Sig}$) and on D and $D_{\mathscr{H}}$. The results for each dataset that are statistically and significantly better than the best performance are indicated in bold (Z-test at 5% risk detailed in Section 1.1.3.a). The last column 'WARP' indicates the synchronous (✓) or asynchronous (×) data type.
Data that need 'WARP' are situated above the line. For each type of delay ('WARP' or non-'WARP'), the datasets are ordered from the less challenging datasets according to the performance of the classically used distances (d_A or DTW) to the most challenging datasets.

	Alternative uni-modal metrics					A priori combinations			M^{2}TML		WARP
Dataset	d_A	d_B	d_F	DTW	$d_{B-\text{DTW}}$	D_{Lin}	D_{Geom}	D_{Sig}	$D(\lambda^*)$	$D_{\mathscr{H}}(\lambda^*)$	WARP
1 ItalyPowerD	0.045	**0.028**	0.078	0.050	0.055	**0.028**	**0.028**	**0.030**	**0.034** (0)	0.046 (0)	×
2 CinCECGtorso	0.103	0.367	0.167	0.349	0.367	**0.094**	**0.094**	**0.093**	**0.092** (0)	**0.088** (0)	×
3 BME	0.173	0.160	0.373	0.107	0.120	0.107	0.107	0.107	**0.007** (0)	**0.007** (0)	×
4 ECG200	**0.120**	**0.070**	0.160	0.230	0.190	**0.070**	**0.070**	**0.070**	**0.080** (0)	**0.080** (0)	×
5 SonyAIBOII	**0.141**	**0.142**	**0.128**	0.169	0.194	**0.142**	**0.142**	**0.144**	**0.162** (0)	**0.142** (0)	×
6 Coffee	0.250	**0.000**	0.357	0.179	0.143	**0.000**	**0.000**	**0.071**	0.143 (0)	**0.036** (10)	×
7 ECG5Days	0.203	0.153	**0.006**	0.232	0.236	0.203	0.203	0.203	**0.012** (10)	0.024 (0)	×
8 SonyAIBO	0.305	0.308	0.258	0.275	0.343	0.308	0.308	0.293	**0.188** (0)	**0.228** (0)	×
9 Adiac	0.389	**0.297**	**0.261**	0.396	0.338	0.373	0.363	0.402	0.358 (0)	0.361 (0)	×
10 Beef	0.467	0.300	0.500	0.500	0.500	0.367	0.267	0.467	**0.033** (0)	0.257 (0)	×
11 Trace	0.240	0.240	0.140	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000** (0)	**0.010** (0)	✓
12 CBF	0.148	0.140	0.382	**0.003**	**0.000**	**0.000**	**0.000**	**0.000**	0.097 (0)	**0.008** (0)	✓
13 CC	0.120	0.113	0.383	**0.007**	0.027	**0.007**	**0.007**	**0.007**	**0.007** (0)	**0.007** (0)	✓
14 DiatomSizeR	0.065	0.076	0.069	**0.033**	**0.029**	**0.033**	**0.033**	**0.042**	0.088 (0)	**0.029** (0)	✓
15 Symbols	0.101	0.111	0.080	**0.050**	**0.043**	**0.051**	**0.050**	**0.052**	0.102 (10)	**0.057** (0)	✓
16 GunPoint	0.087	0.113	**0.027**	0.093	**0.027**	**0.027**	**0.027**	**0.040**	**0.033** (0)	**0.053** (10)	✓
17 FacesUCR	0.231	0.227	0.175	0.095	0.102	0.098	0.098	0.099	**0.068** (10)	**0.068** (0)	✓
18 TwoLeadECG	0.253	0.153	0.103	0.096	**0.008**	**0.005**	**0.005**	0.018	**0.006** (0)	0.016 (10)	✓
19 UMD	0.194	0.222	0.229	0.118	**0.090**	0.111	0.111	0.118	0.104 (0)	**0.042** (0)	✓
20 MoteStrain	**0.121**	0.263	0.278	0.165	0.171	0.260	0.248	0.188	0.185 (0)	0.179 (0)	✓
21 Lighting2	**0.246**	**0.246**	**0.148**	**0.131**	**0.213**	**0.131**	**0.131**	**0.131**	**0.213** (0)	**0.131** (0)	✓
22 OliveOil	**0.133**	**0.133**	**0.167**	**0.200**	**0.100**	**0.133**	**0.133**	**0.133**	**0.167** (0)	**0.100** (10)	✓
23 FISH	0.217	**0.149**	0.229	**0.166**	**0.137**	**0.109**	**0.137**	**0.126**	**0.149** (0)	0.240 (0)	✓
24 FaceFour	0.216	0.216	0.239	0.170	0.136	0.170	0.170	0.170	**0.023** (0)	0.114 (0)	✓
25 SwedishLeaf	0.211	0.186	0.146	0.208	**0.109**	**0.115**	**0.110**	**0.125**	**0.142** (0)	**0.114** (0)	✓
26 MedicalImages	0.316	0.313	0.345	**0.263**	0.290	**0.263**	**0.263**	**0.263**	**0.237** (0)	**0.241** (10)	✓
27 Lighting7	**0.425**	**0.411**	**0.316**	**0.274**	**0.288**	**0.342**	**0.356**	**0.342**	**0.411** (0)	**0.233** (0)	✓
28 PowerCons	**0.366**	0.445	**0.315**	0.397	0.401	0.401	0.401	0.401	**0.318** (0)	**0.342** (0)	✓
29 OSULeaf	0.484	0.475	0.426	0.409	**0.265**	**0.264**	**0.264**	**0.322**	0.421 (0)	0.388 (0)	✓
30 InlineSkate	**0.658**	**0.658**	0.675	**0.616**	**0.623**	**0.605**	**0.605**	**0.602**	0.833 (10)	**0.625** (0)	✓

Table 4.4: 1-NN test error rates for standard, *a priori* combined and M^{2}TML measures.

4.3.2 Comparison of the classification performances on the test set

From Table 4.4, we can see first that the 1-NN classification reaches the best results in:

1. Less than one-third of the data when based on unimodal metrics d_A, d_B or d_F
2. Slightly more than one-third for unimodal metrics DTW and $d_{B-\text{DTW}}$
3. Two-thirds (19 - 20 times on 30) when based on *a priori* combined metrics D_{Lin}, D_{Geom} and D_{Sig}
4. More than two-thirds (21 times on 30) when based on learned metrics D or $D_{\mathscr{H}}$.

Particularly, note that for nearly all datasets for which an uni-modal metric succeeds, the M^{2}TML metrics succeed similarly or lead to equivalent results. However, for several challenging datasets (*e.g.* FaceFour, Beef, FaceUCR, SonyAIBO, BME, CinCECGTorso), M^{2}TML realizes drastic improvements, to the best of our knowledge never achieved before for these challenging public data. For instance, a score of 3% is obtained for Beef against an error rate varying from 30% to 50% for alternative metrics, and of 2.3% obtained for FaceFour v.s. 13% to 23% for alternative metrics. Finally, D and $D_{\mathscr{H}}$ are most datasets, either equivalent or better if only compared to the standard metrics d_A (the Euclidean distance) and DTW.

If we compare the *a priori* combined metrics (D_{Lin}, D_{Geom}, D_{Sig}) based on only the unimodal metrics involved in the combination (either d_A and d_B or DTW and $d_{B-\text{DTW}}$), we observe that *a priori* combined metrics achieved on two-third of the data with an equivalent or better score. Compared to the learned metrics (D, $D_{\mathscr{H}}$), the results are globally similar except for 8 datasets where the learned metrics perform better (FaceFour, Beef, ECG5Days, FaceUCR, SonyAIBO, PowerCons, BME, UMD) and one where the *a priori* combined metrics perform better (OSULeaf). Note that the combined metric D_{Sig} is limited to two components and can't be easily extend to other metrics in its combination. D_{Lin} and D_{Geom} could be easily extended and a proposition could be:

$$D_{Lin}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{h=1}^{p} \alpha_h d_h(\mathbf{x}_i, \mathbf{x}_j) \tag{4.1}$$

$$D_{Geom}(\mathbf{x}_i, \mathbf{x}_j) = \prod_{h=1}^{p} \alpha_h d_h(\mathbf{x}_i, \mathbf{x}_j) \tag{4.2}$$

However, by considering p metrics d_h the resulting models requires to optimize p parameters. The grid search to find the best parameters α_h can become time consuming. The M^{2}TML approach has been proposed to prevent such an exhaustive grid search.

In the second part, we perform a graphical analysis for a global comparison on the whole datasets. In Fig. 4.2-a, each dataset is projected according to, on the x-axis its best error rate obtained for D and $D_{\mathscr{H}}$, and on y-axis its best performance *w.r.t* the standard amplitude-based metrics d_A and DTW. In Fig. 4.2-b, the y-axis is related to the best error rate of the

behavior-based metrics d_B and and $d_{B-\text{DTW}}$. In Fig. 4.2-c, the y-axis is related to the best error rate of the two "non-warp" uni-modal metrics d_A and d_B, . In Fig. 4.2-d, the y-axis is related to the best error rate of the two "warp" uni-modal metrics DTW and $d_{B-\text{DTW}}$ that are also the two most performant uni-modal metrics. In Fig. 4.2-e, the y-axis is related to the error rate of the frequential-based metric d_F. In Fig. 4.2-f, the y-axis is related to the best error rate of the a priori-combined metrics $D_{Lin}, D_{Geom}, D_{Sig}$.
For all plots, let first give some interpretations. If the datasets are situated on the first bisector, it means that the considered metrics in x-axis and y-axis have equal performance. For datasets situated above the first bisector, it means in this case that M^{2}TML method is better that the considered metrics in y-axis. Similarly, for datasets situated below the first bisector, it means in this case that the considered metrics in y-axis are better than M^{2}TML. Less challenging datasets (low classification error rate) are situated near the origin and challenging dataset (high classification error rate) are situated far from the origin.
For all plots, we can note that the datasets are principally projected above the first bisector, indicating higher error rates mostly obtained for uni-modal and *a priori* combined metrics than for M^{2}TML. For the less challenging datasets (near the origin of each graph), although almost projected near the bisector denoting equal performances for the compared metrics, M^{2}TML still bring improvements with projections clearly positioned above the bisector. Finally, from all plots, note that some datasets (Adiac, OSULeaf, InlineSkate) remains challenging for all studied metrics.

4.3.3 Analysis of the discriminative features

For the learned metric D, thanks to the L_1 regularization, the learned SVM reveals the features that most differentiate pull from push pairs. We recall that the weight for each feature can be analyzed through the weight vector $\mathbf{w}$ obtained by learning the SVM classifier. Table 4.5 shows the sparse, muti-modal and multi-scale potential of M^{2}TML approach. It gives for each dataset, the weights of the top five 'discriminative' features that contribute to the definition of D. For instance, for FaceFour D reaches an error of 2.3% by combining, in the order of importance, the behavior $d_{B-\text{DTW}}$, frequential d_F and amplitude DTW modalities, at the global (I^0) and local (I^4, I^5, I^2) scales. For Beef, the learned model is very sparse as D involves only the behavior modality based on the segment I^3 (d_B^3). Note that if we look at only the most discriminative feature (1st column), the M^{2}TML method helps to localize discriminative modality and a specific temporal scale (localization) that could not be easily guessed *a priori* (*e.g.*, Lightning7: behavior modality on the segment I_6 ($d^6_{B-\text{DTW}}$), OliveOil: frequential modality on the segment I_5 (d^5_F), TwoLeadECG: behavior modality on the segment I_4 ($d^4_{B-\text{DTW}}$)).

In Fig. 4.3, we plot the weights of all features for SonyAIBO, Beef, CincECGtorso and FaceFour cases as an example. It illustrates both the sparsity of the M^{2}TML approach (Beef, CincECGtorso and FaceFour) and the ability of the algorithm to combine all the features into the metric D (SonyAIBO). In particular, the approach is able to either select one single feature (Beef) or combine several selected features (CinCECGTorso, FaceFour). Fig. 4.4 illustrates the temporal locations of the most discriminative features for these datasets. Note

that from looking at the temporal representation, it is not easy to determine *a priori* which modality (value, behavior, frequential) and at which temporal scale (localization) is the most discriminative feature to separate the classes.

In summary, we can emphasize that for almost all datasets, the definition of D involves no more than five features (the most contributive ones), that assesses not only the model's sparsity but also the representativeness of the revealed features.

Dataset	Feature weights (%)				
ItalyPowerD	d_B^0 (27.5%)	d_F^4 (17.2%)	d_F^1 (12.3%)	d_A^1 (11.2%)	d_B^2 (9%)
CinCECGtorso	d_F^0 (38.4%)	d_A^5 (13.1%)	d_B^4 (11.5%)	d_F^1 (11.2%)	d_A^2 (9.8%)
BME	$d_{B-\text{DTW}}^0$ (75.2%)	d_F^4 (15.5%)	$d_{B-\text{DTW}}^2$ (5.8%)	$d_{B-\text{DTW}}^1$ (1.9%)	d_F^1 (0.7%)
ECG200	d_B^0 (89.6%)	d_B^6 (2.4%)	d_A^3 (2.3%)	d_B^1 (2.2%)	d_B^4 (2%)
SonyAIBOII	d_B^3 (100%)	-	-	-	-
Coffee	d_F^4 (59.4%)	d_B^6 (6.4%)	d_B^2 (5.6%)	d_B^3 (5%)	d_F^5 (4.4%)
ECG5Days	d_B^5 (44.9%)	d_B^6 (36.3%)	d_A^4 (7.9%)	d_F^6 (7.4%)	d_B^4 (2.7%)
SonyAIBO	d_F^3 (30.8%)	d_B^6 (27.3%)	d_B^5 (5%)	d_A^1 (4.1%)	d_B^0 (3.9%)
Adiac	d_F^0 (79.2%)	d_B^4 (13.8%)	d_A^4 (3.5%)	d_F^5 (1.7%)	d_B^5 (1.2%)
Beef	d_B^3 (100%)	-	-	-	-
Trace	DTW^0 (58.3%)	DTW^6 (6.9%)	$d_{B-\text{DTW}}^0$ (5.8%)	DTW^2 (5.6%)	DTW^5 (5.5%)
CBF	d_F^6 (18.5%)	d_F^3 (18.5%)	d_F^0 (15.2%)	DTW^4 (12.4%)	d_F^1 (9%)
CC	d_F^0 (17.1%)	DTW^3 (13.2%)	DTW^2 (11.4%)	$d_{B-\text{DTW}}^2$ (11%)	d_F^1 (7.1%)
DiatomSizeR	d_F^5 (100%)	-	-	-	-
Symbols	d_F^6 (38.2%)	DTW^0 (16.1%)	DTW^1 (12%)	d_F^0 (6.7%)	DTW^2 (4.7%)
GunPoint	$d_{B-\text{DTW}}^0$ (41.1%)	DTW^5 (14.7%)	DTW^2 (9.5%)	DTW^4 (6.1%)	d_F^4 (6%)
FacesUCR	d_F^2 (21.5%)	$d_{B-\text{DTW}}^0$ (19.5%)	d_F^4 (16.7%)	DTW^0 (12.6%)	$d_{B-\text{DTW}}^2$ (8.6%)
TwoLeadECG	$d_{B-\text{DTW}}^4$ (60%)	d_F^1 (12%)	DTW^4 (11.4%)	$d_{B-\text{DTW}}^6$ (7.6%)	$d_{B-\text{DTW}}^1$ (4.2%)
UMD	$d_{B-\text{DTW}}^0$ (99.8%)	$d_{B-\text{DTW}}^5$ (0.2%)	-	-	-
MoteStrain	$d_{B-\text{DTW}}^5$ (93.2%)	$d_{B-\text{DTW}}^6$ (6.8%)	-	-	-
Lighting2	$d_{B-\text{DTW}}^0$ (100%)	DTW^0 (0%)	d_F^0 (0%)	DTW^1 (0%)	$d_{B-\text{DTW}}^1$ (0%)
OliveOil	d_F^5 (97%)	$d_{B-\text{DTW}}^2$ (3%)	-	-	-
FISH	$d_{B-\text{DTW}}^5$ (17.9%)	d_F^0 (10.5%)	$d_{B-\text{DTW}}^6$ (9.9%)	$d_{B-\text{DTW}}^4$ (8.3%)	$d_{B-\text{DTW}}^3$ (7.8%)
FaceFour	$d_{B-\text{DTW}}^4$ (66.7%)	d_F^4 (22.4%)	$d_{B-\text{DTW}}^3$ (5.6%)	$d_{B-\text{DTW}}^1$ (5.3%)	-
SwedishLeaf	d_F^0 (23.9%)	$d_{B-\text{DTW}}^1$ (14.1%)	$d_{B-\text{DTW}}^2$ (10.5%)	$d_{B-\text{DTW}}^6$ (10%)	$d_{B-\text{DTW}}^5$ (6%)
MedicalImages	$d_{B-\text{DTW}}^1$ (53.3%)	d_F^3 (12.9%)	$d_{B-\text{DTW}}^2$ (10.7%)	$d_{B-\text{DTW}}^3$ (10.1%)	$d_{B-\text{DTW}}^0$ (3.8%)
Lighting7	$d_{B-\text{DTW}}^6$ (77.7%)	d_F^6 (20.8%)	$d_{B-\text{DTW}}^5$ (1.5%)	DTW^3 (0%)	DTW^1 (0%)
PowerCons	d_F^0 (26.1%)	DTW^0 (20.3%)	d_F^1 (19.3%)	$d_{B-\text{DTW}}^0$ (6.1%)	d_F^2 (5.1%)
OSULeaf	$d_{B-\text{DTW}}^2$ (84.7%)	d_F^6 (7.7%)	d_F^0 (2.7%)	d_F^5 (1.6%)	DTW^5 (1.2%)
InlineSkate	DTW^2 (25.7%)	d_F^4 (24.3%)	d_F^3 (16.5%)	$d_{B-\text{DTW}}^2$ (11.2%)	$d_{B-\text{DTW}}^3$ (5.2%)

Table 4.5: Top 5 multi-modal and multi-scale features involved in D

4.3.4 Effect on the neighborhood before and after learning

In the last part, we compare the global effect of the alternative and M^{2}TML metrics on the 1-NN neighborhood distribution and class discrimination. For that, a MultiDimensional Scaling[4] (MDS) is used to visualize the distribution of samples according to their pairwise dissimilarities. Briefly, we recall that MDS is a method of visualizing the proximity between samples in a dataset (Section 2.2). Given an input dissimilarity matrix, we can project the time series on a 2-dimensional plot whose configuration reproduces the best the dissimilarities between the time series. Note that the MDS representation has no link with the dissimilarity space representation whose dimensions are basic temporal metrics.

For FaceFour, Fig. 4.5 shows the first obtained plans and their corresponding stresses, the classes being indicated in different symbols and colors. We can see distinctly the effect of the learned D that leads to more compact and more isolated classes with robust neighborhoods for 1-NN classification (*i.e.*, closer pull pairs and far away push pairs) than the best alternative metric $d_{B-\mathrm{DTW}}$ that shows more overlapping classes and heterogeneous neighborhoods.

4.4 Conclusion of the chapter

The large conducted experiments and the impressive performances obtained attest the efficiency of the learned M^{2}TML metrics for time series nearest neighbors classification. As discussed, the datasets encompass time series that involve global or local temporal comparison, require or not time warping, with linearly or non linearly separable neighborhoods.
Finally, let us underline the merit of the M^{2}TML solution, that not only leads to equivalent or better performances from the standard metrics (Euclidean distance, Dynamic time warping), but also provides a comprehensive and fine-grained information about which modalities are mostly discriminant, how they should be combined and precisely at which temporal granularity (localization).

[4]Matlab function: mdscale for metrics and non metrics

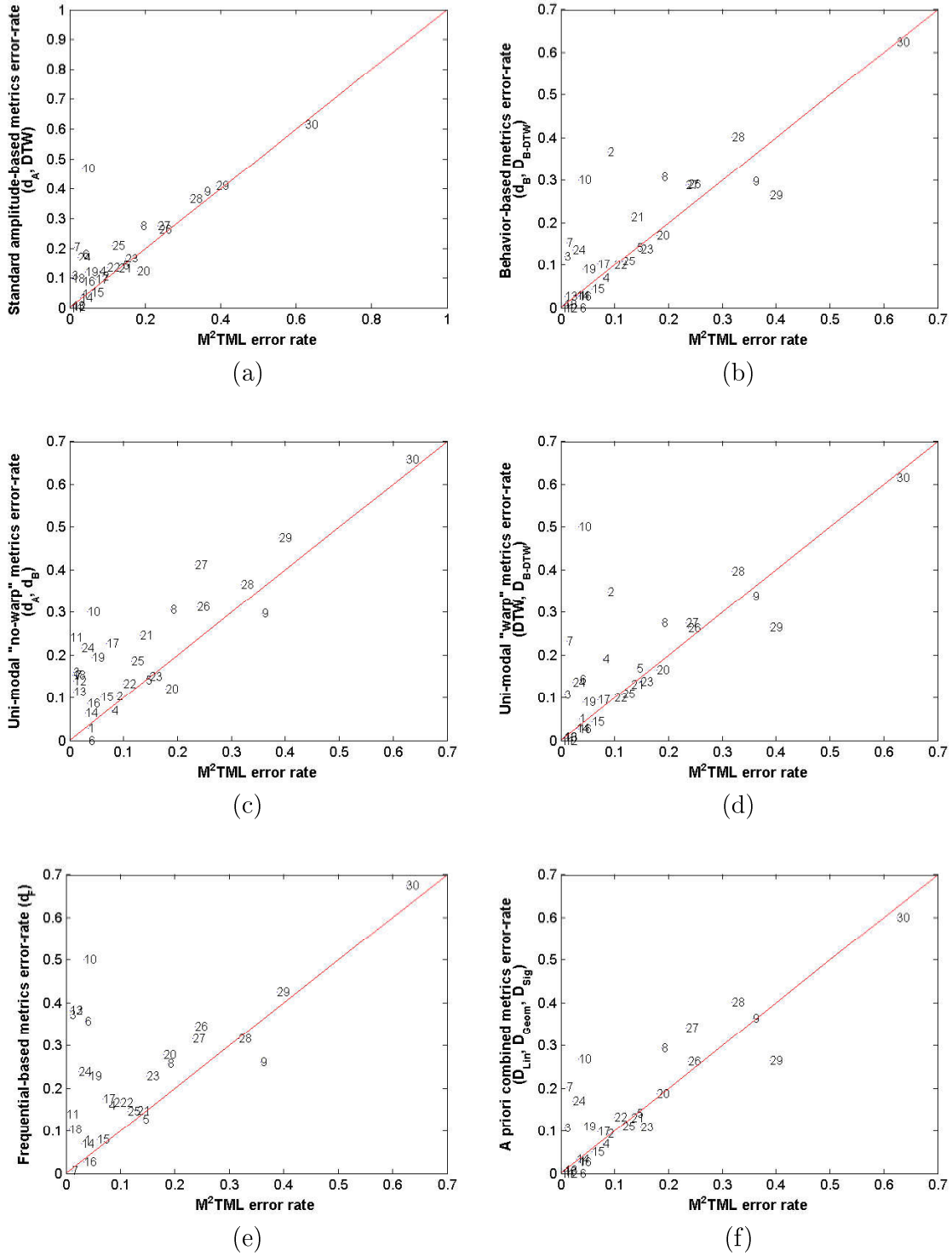
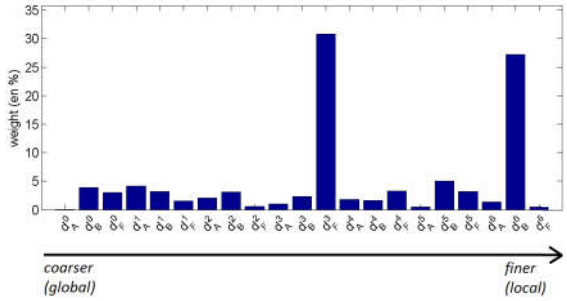
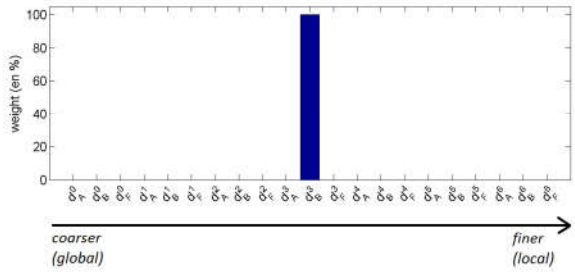


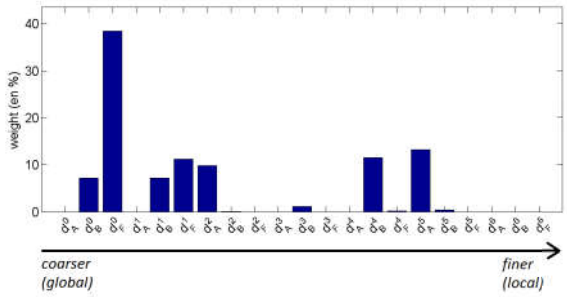
Figure 4.2: (a) Standard amplitude-based (Euclidean distance d_A and DTW) vs. M^{2}TML (D and $D_{\mathscr{H}}$) metrics. (b) Behavior-based (d_B and $d_{B-\mathrm{DTW}}$) vs. M^{2}TML metrics. (c) No-warp (d_A and d_B) vs. M^{2}TML metrics. (d) Warp (DTW and $d_{B-\mathrm{DTW}}$) vs. M^{2}TML metrics. (e) Frequential-based (d_F) vs. M^{2}TML metrics. (f) *A priori* (D_{Lin}, D_{Geom}, D_{Sig}) vs. M^{2}TML metrics.



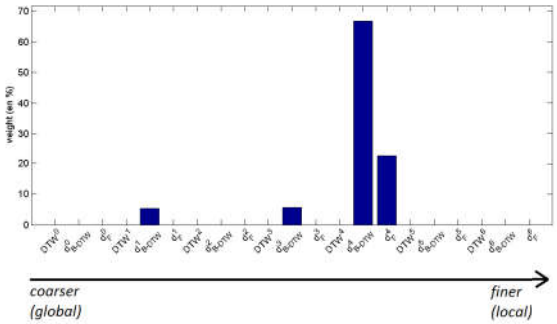
(a) SonyAIBO



(b) Beef

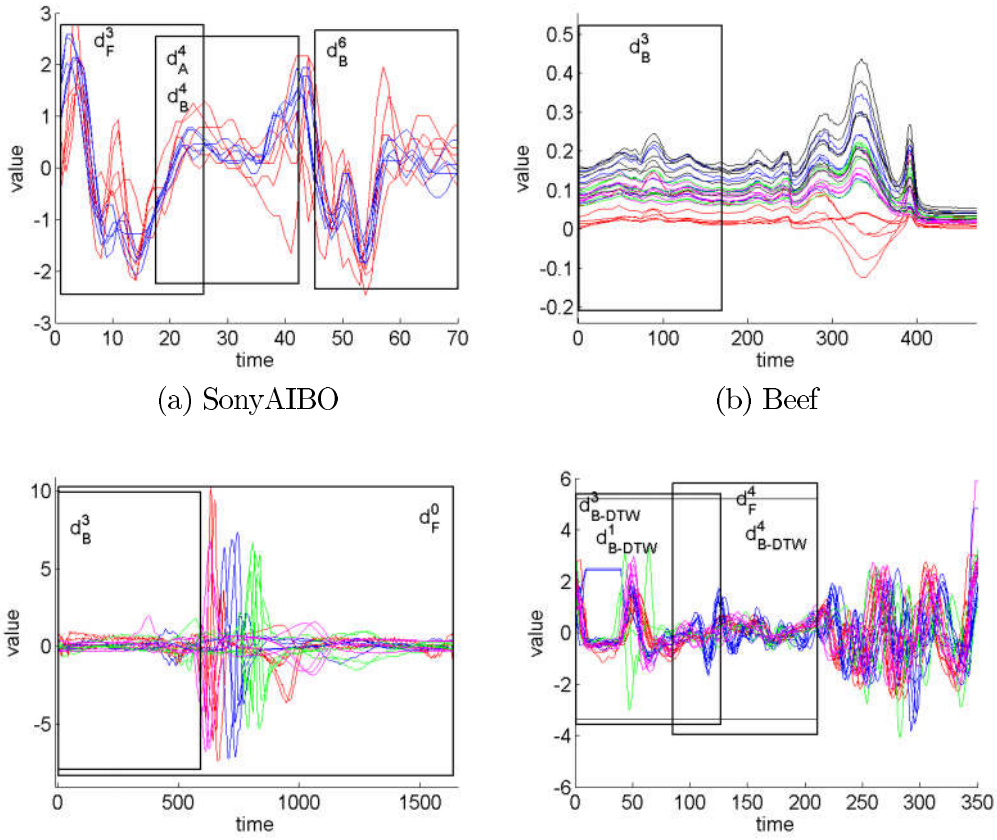


(c) CinC ECG torso



(d) FaceFour

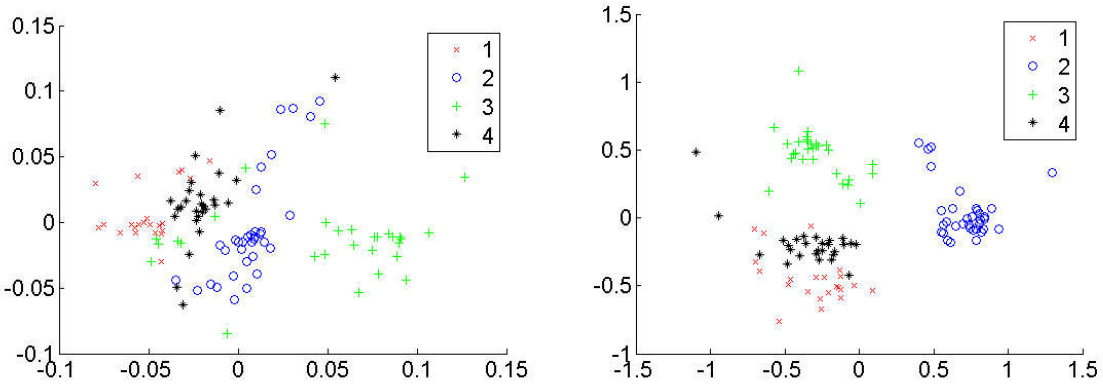
Figure 4.3: M^{2}TML feature weights for 4 datasets.



(a) SonyAIBO (b) Beef

(c) CinC ECG torso (d) FaceFour

Figure 4.4: Temporal representation of the top M^{2}TML feature weights for 4 datasets.



(a) FaceFour ($d_{B-\mathrm{DTW}}$, stress: 20,1%) (b) FaceFour (D, stress: 18,9%)

Figure 4.5: MDS visualization of the $d_{B-\mathrm{DTW}}$ (Fig. a) and D (Fig. b) dissimilarities for FaceFour

Conclusion and perspectives

Conclusion

By considering usual classifiers (*e.g.*, k-NN) that are based on distance between samples, we have proposed a Multi-modal and Multi-scale Temporal Metric Learning (M^{2}TML) framework for a robust k-NN classifier. It is based on a new space representation, the pairwise dissimilarity space, where pairs of time series are embedded as vectors described by different basic temporal metrics. A metric combining the basic metrics can be seen as a function (linear or non-linear) of the pairwise dissimilarity space, learned by using a large margin optimization process (SVM) inspired from the nearest neighbors metric learning framework. The obtained metric satisfies the properties of a dissimilarity (positivity, reflexivity, symmetry), leads to good performances on a large number of public datasets, and gives an interpretable solution that allows to analyze the modalities and scales that are the most discriminant.

Temporal data may be compared based on various characteristics, called modalities. Time series can be compared not only on their amplitudes like static data, but also on other modalities such as their behavior, frequency, etc. To cope with delays in real time series, Dynamic time warping approach can be used to re-align the signals. Some authors propose to combine several modalities through a combination function but the combinations are either, limited to two modalities or in the case of multiple modalities (more than 2), the number of parameters to optimize for a classifier may become time consuming. In general, state of the art approaches compared the time series by involving all observations, restricting the potential of comparison measures (metrics) to capture local differences. We proposed to take into account local characteristics, that we named multi-scale. We have believed that all of these considerations (modality, scale, delays) should be taken into consideration in the definition of a metric in order to improve the performance of the classifier.

The objective is to learn a metric for a robust k-NN. For that, we propose a general formalization of the problem of learning a combined multi-modal and multi-scale temporal metric (M^{2}TML). Based on a pairwise dissimilarity representation of the pairs of time series, the metric learning problem can be reduced to the learning of a linear or non linear function of the dissimilarity space that satisfies the properties of a dissimilarity. Inspired from metric learning work, the problem is formalized as an optimization problem involving a regularization and loss term which aims to pull samples that are expected to be similar and push away samples that should be dissimilar. First, by considering a linear combination of the basic metrics, changing the regularization term leads the general formalization to a linear and quadratic formalization. The latter allows to extend to the learning of non-linear functions thanks to the "kernel" trick. However, the methods can lead to functions that doesn't meet the properties of a dissimilarity (non-positivity). Secondly, we propose to formulate the problem as a SVM problem which aims to separate pull and push samples, then we define a metric that satisfies the required properties of a dissimilarity.

The efficiency of the proposed SVM-based solution has been tested in the case of classification of univariate time series, on a wide variety of datasets coming from various fields (simulated data, medicine, power consumption, etc.), diverse sizes of training and testing, various number of classes, etc. The M^{2}TML solution achieves not only, either equivalent or better performances compared to the standard global metrics (Euclidean distance, dynamic time warping, temporal correlation, Fourier-based distance), but it also provides a sparse and interpretable solution that allows to give a comprehensive analysis of the most discriminative modalities and their respective temporal granularity that may not be always intuitive *a priori*.

Perspectives

Extension to other modalities, multivariate problems and other type of data

In this work, we only focus on three basic temporal metrics (euclidean distance, temporal correlation, Fourier-based distance). Montero & Vilar propose in [MV14] a review on a wide number of metrics dedicated to time series. For remaining challenging datasets in our experiments, it could be interesting to integrate other basic temporal metrics in our framework to see the obtained results.

The framework can be easily extend to multivariate problem. For each dimension, we consider the set of multi-modal and multi-scale description. Then, we consider the union over the dimensions as the pairwise dissimilarity description $d_1, \ldots, d_p$.

The proposed solution has been tested in the case of time series data but the framework is more general. It can be applied to any other type of data (strings, graphs, images) to learn a combined metric. These data might be compared on other characteristics. Deza & Deza makes a detailed review of metrics for various domains in [DD09].

Other possibilities for the multi-scale description

A second improvement is about the multi-scale description that denotes in this work as a temporal segmentation. We propose a multi-scale approach based on a binary segmentation using a dichotomy process. Other solutions could be proposed, in particular, in order to localize automatically finely events of interest. For example, in the case of the dataset SonyAIBO, the discriminative temporal locations of the signal is known *a priori* (Fig. 4.6). With the actual multi-scale description, it is not possible to extract exactly the two red patterns of interest. A solution based on a sliding window of variable lengths could be used to locate precisely these patterns.

[5]source: `http://www.cs.ucr.edu/~eamonn/LogicalShapelet.pdf`

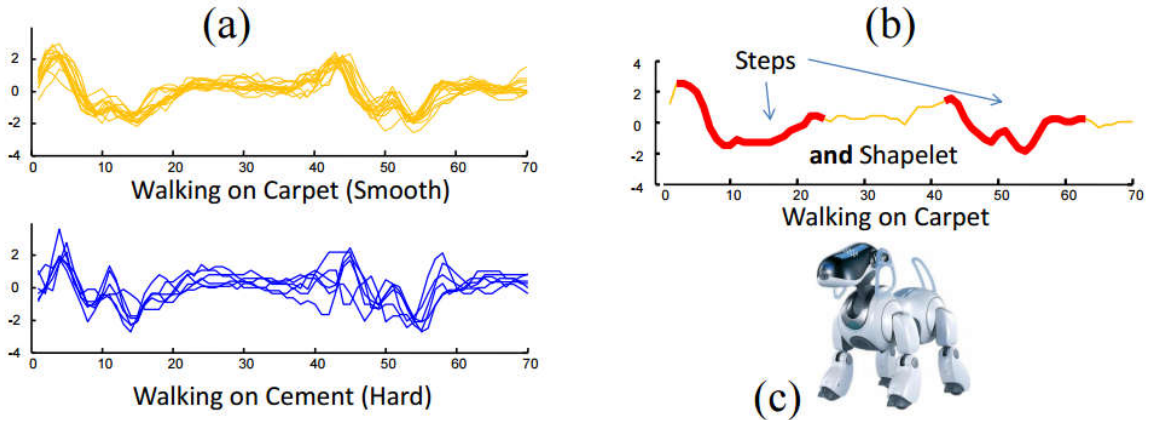


Figure 4.6: (a) Two classes of time series from the Sony AIBO accelerometer. (b) The and-shapelets from the walk cycle on carpet. (c) The Sony AIBO Robot.[5]

Learning of local metrics

Some authors suggest that in some datasets, global linear metric learning approach may not be sufficient to improve the accuracy of k-NN classification [WS09b]; [WWK12]. Since the discriminatory power of the input features might vary between different neighborhoods, learning a global metric cannot fit well the distance over the data. To overcome this difficulty, they propose to learn a metric on each neighborhood, referred as local metric learning.
Similarly, the M^{2}TML framework could be extended to learn local combined temporal metrics for each neighborhood. The objective is to learn for each n set $Pull_i$ and $Push_i$ (n being the number of samples in the training set) a local metric using the same framework than the one we propose in this work. We obtain n local metrics D_i. Then, to classify a new sample $\mathbf{x}_{test}$, we compute the n metrics $D_i(\mathbf{x}_{i,test})$ and classify $\mathbf{x}_{test}$ using the k lowest distances $D_i(\mathbf{x}_{i,test})$.

Re-iteration of the initial metric

Similarly to Large Margin Nearest Neighbors (LMNN) approach proposed by Weinberger & Saul [WS09b], the M^{2}TML approach might inherit the same problem of the initial distance, *i.e.*, fixing the set $Pull_i$ and $Push_i$ according to an initial distance (Euclidean distance in this work). Other initial distances could have been used. If the initial distance is far away from the optimal solution, the definition of the sets $Pull_i$ and $Push_i$ can impact the convergence to the optimal solution. In same spirit as the multi-pass LMNN approach proposed by Weinberger & Saul, we could re-iterate the learning process. At each step, we re-define the sets $Pull_i$ and $Push_i$ using the distance learned at the previous step. Then, we stop the learning when arriving at convergence (*e.g.*, the sets $Pull_i$ and $Push_i$ doesn't evolve anymore or evolve slightly between two steps).

Other propositions to define the combined metric

First, as said in Section 3.7, note that the framework to define the metric D and $D_{\mathscr{H}}$ can also be used in the linear and quadratic formalization. However, the obtained solution for D and $D_{\mathscr{H}}$ can be far away from the original form of D that has been optimized in the optimization problem.

Secondly, we have proposed a form for the metric D and $D_{\mathscr{H}}$ so that it satisfies the properties of a dissimilarity. Other solutions could have been proposed. In particular, instead of using a max operator in the definition of D and $D_{\mathscr{H}}$, an other variant could consider a parameter λ that can be either positive or negative. In the case of negative λ, the action of the exponential term would become a pull term instead of a push term. Note that in both cases, for extreme value of λ, there exists a risk to binarize the metric. In particular, for $\lambda \mapsto -\infty$, there exists a risk of having zero values for the nearest neighbors, which could lead to problems when classifying by the nearest neighbors.

Extension to regression problems

For the SVM-based solution, in the pairwise dissimilarity space, each vector $\mathbf{x}_{ij}$ is labeled y_{ij} by following the rule: if $\mathbf{x}_i$ and $\mathbf{x}_j$ are similar, the vector $\mathbf{x}_{ij}$ is labeled -1; and $+1$ otherwise. For classification problems, the concept of similarity between samples $\mathbf{x}_i$ and $\mathbf{x}_j$ is driven by the class label y_i and y_j in the original space:

$$y_{ij} = \begin{cases} +1 & \text{if } y_i = y_j \\ -1 & \text{if } y_i \neq y_j \end{cases} \tag{4.3}$$

For regression problems, each sample $\mathbf{x}_i$ is assigned to a continuous value y_i. Two approaches are possible to define the similarity concept. The first one discretizes the continuous space of values of the labels y_i to create classes. One possible discretization bins the label y_i into Q intervals as illustrated in Fig. 4.7. Each interval becomes a class which associated value can be set for example as the mean or median value of the interval. Then, the classification framework is used to define the pairwise label y_{ij}.

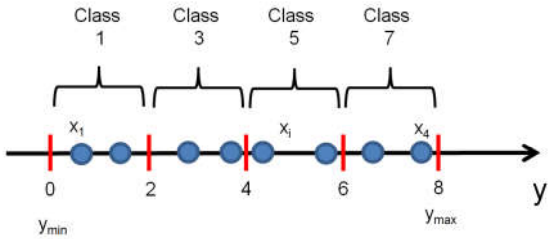


Figure 4.7: Example of discretization by binning a continuous label y into $Q = 4$ equal-length intervals. Each interval is associated to a unique class label. In this example, the class label for each interval is equal to the mean in each interval.

This approach may leads to border effects between the classes. For instance, two samples $\mathbf{x}_i$ and $\mathbf{x}_j$ that are close to a frontier and that are on different sides of the border will be considered as different, as illustrated in Fig 4.8. Moreover, a new sample $\mathbf{x}_j$ will have its labels y_j assigned to a class and not a real continuous value.

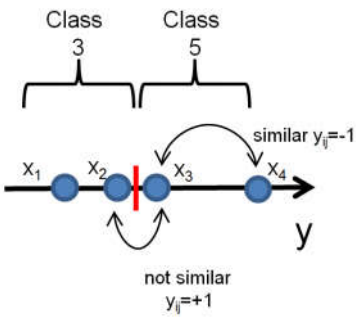


Figure 4.8: Border effect problems. In this example, $\mathbf{x}_2$ and $\mathbf{x}_3$ have closer value labels y_2 and y_3 than $\mathbf{x}_3$ and $\mathbf{x}_4$. However, with the discretization $\mathbf{x}_2$ and $\mathbf{x}_3$ don't belong to the same class and thus are consider as not similar.

The second approach considers the continuous value of y_i, computes a L_1-norm between the labels $|y_i - y_j|$ and compare this value to a threshold ϵ. Geometrically, a tube of size ϵ around each value of y_i is built. Two samples $\mathbf{x}_i$ and $\mathbf{x}_j$ are considered as similar if the absolute difference between their labels $|y_i - y_j|$ is lower than ϵ (Fig. 4.9):

$$y_{ij} = \begin{cases} -1 & \text{if } |y_i - y_j| \leq \epsilon \\ +1 & \text{otherwise} \end{cases} \tag{4.4}$$

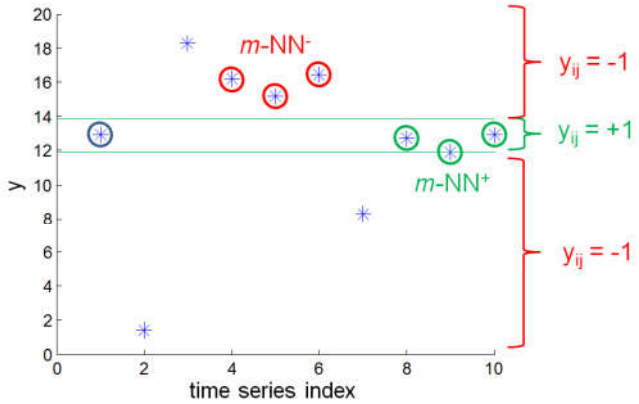


Figure 4.9: Example of pairwise label definition using an ϵ-tube (red lines) around the time series $\mathbf{x}_i$ (circled in blue). For, time series $\mathbf{x}_j$ that falls into the tube, the pairwise label is $y_{ij} = -1$ (similar) and outside of the tube, $y_{ij} = +1$ (not similar). m-NN$^+$ and m-NN$^-$ time series are indicated respectively in green and red circle for k-NN with $k = 1$ and $m = 3$ neighborhood.

Using the learned combined metric in other algorithms

In this work, we propose to learn a temporal metric for a robust k-NN classifier. As explained in Section 1.2.1, for industrial practical usage, the k-NN algorithm may present some disadvantages, mainly due to its computational complexity, both in memory space (storage of the training samples) and time (search of the neighbors).
Inspired from the work on temporal trees in [DCA12], we could use the learned metric in an other classifier such as a decision tree. For multivariate classification problems, an idea could be to learn in a first step a linear or non-linear multi-modal and multi-scale temporal metric for each dimension. Then, in a second step, given the set of multi-modal and multi-scale temporal metrics, we build a temporal tree based on the training samples: First, for each dimension, we split the data into two partitions using a clustering algorithm such as k-means ($k = 2$) with the learned temporal metric for the considered dimension. Secondly, similarly to classical decision tree, we compute a criterion (*e.g.*, a Gini coefficient or Information Gain coefficient) to select the best split, *i.e.*, the best learned temporal metric that minimize the criterion. Thirdly, we compute for each partition the centroid, *i.e.*, the time series that minimizes the mean distance over all the other time series in the same partition. Then, for each obtained partition, we re-iterate the process until the stopping condition is met, *e.g.*, all the sample in the partition have the same class label or the number of sample have fallen below some minimum threshold. It corresponds a leaf node and the label of the node is assigned to the one of the majority. To classify a new time series $\mathbf{x}_{test}$, similarly to classical decision tree, at each node, we compute the distance D or $D_{\mathscr{H}}$ of the new sample to each centroid. The time series $\mathbf{x}_{test}$ is then assigned to the node of the nearest centroid until it reaches a leaf node where its label is assigned.

List of publications

Journal

- C. Do, A. Douzal-Chouakria, S. Marié, M. Rombaut. Multi-modal and Multi-scale Temporal Metric Learning for Time Series Nearest Neighbors Classification, Pattern Recognition Letters 2016 (under submission)
-

papier Springer?

Conference

- C. Do, A. Douzal-Chouakria, S. Marié, M. Rombaut. Multiple Metric Learning for large margin kNN Classification of time series, EUSIPCO'2015
- C. Do, A. Douzal-Chouakria, S. Marié, M. Rombaut. Temporal and Frequential Metric Learning for Time Series kNN Classification. ECML-PKDD'2015 Workshop on Advanced Analytics and Learning from Temporal Data, 39-45