

新人ゼミ課題

大場 大輔

令和3年4月21日

3. 時系列識別

すでに画像処理や Python を用いたプログラミングの経験があることと、Brian 研究室に配属されたことを考慮して、このコースを選択しました。なお、本レポートで使用したソースコードはこのページ¹にて確認できます。具体的には3_TimeSeriesClassification/codesの中に、各レベルごとに使用したコードを載せてあります。不明点があれば、ぜひお知らせください。まずは、今回のレポートで使用した動的時間伸縮法と k 近傍法について簡単に要点を報告します。

動的時間伸縮法について

動的時間伸縮法 (DTW; Dynamic Time Warping) とは、2つの時系列データの距離（類似度）を求める手法であり、具体的には比較するデータの各サンプル同士の距離（今回はユークリッド距離を使用）を総当りで求めて、この2つのデータ間の最短経路を求める手法です。この手法は本研究室の創設者である迫江博昭先生が開発したということで、個人的には非常に驚きました。

k 近傍法について

k 近傍法 (kNN; k-Nearest Neighbor) とは、データ分類手法の一つであり、あるデータが、その特徴空間上で最近接している k 個の他のデータのクラスの情報をもとに、多数決的にそのデータの所属するクラスを判定する手法です。今回の Level1 から Level3 のケースであれば、test1 と test2 の2つのクラスにそれぞれ一つのサンプルしか存在しないため、必然的に k=1 のケース、つまり2つのクラスの DTW の値のうち、小さい方のクラスに属すると考えればよいことがわかります。

¹ <https://github.com/ba-san/assignment2021>

Level1

この設問においては、まず自分で実際に DTW を実装し、さらにその計算結果が `tslearn` のライブラリで提供されている `dtw` の関数によるものと一致することを確認しました。その上で下記の通り、各 reference データに対するクラス 1 とクラス 2 との距離を算出し、グラフを描画しました。結果から、`test1` から `test3` はクラス 1 に、`test4` から `test6` はクラス 2 に属することがわかりました。

```
(test) daisuke@daisuke-FRONTIER:~/Documents/assignment2021/3_TimeSeriesClassification/tools$ python level1.py
self implemented dtw| class1:0.3436514936909193 class2:3.3567946173824814
dtw from library| class1:0.3436514936909193 class2:3.3567946173824814

self implemented dtw| class1:0.6734290546954446 class2:3.2932335433745052
dtw from library| class1:0.6734290546954446 class2:3.2932335433745052

self implemented dtw| class1:0.3455674741783433 class2:3.596772577144544
dtw from library| class1:0.3455674741783433 class2:3.596772577144544

self implemented dtw| class1:3.6975090716425565 class2:0.22210845177975555
dtw from library| class1:3.6975090716425565 class2:0.22210845177975555

self implemented dtw| class1:3.71819451711849 class2:0.2258337265489812
dtw from library| class1:3.71819451711849 class2:0.2258337265489812

self implemented dtw| class1:3.5602242325460343 class2:0.29442989019289467
dtw from library| class1:3.5602242325460343 class2:0.29442989019289467
```

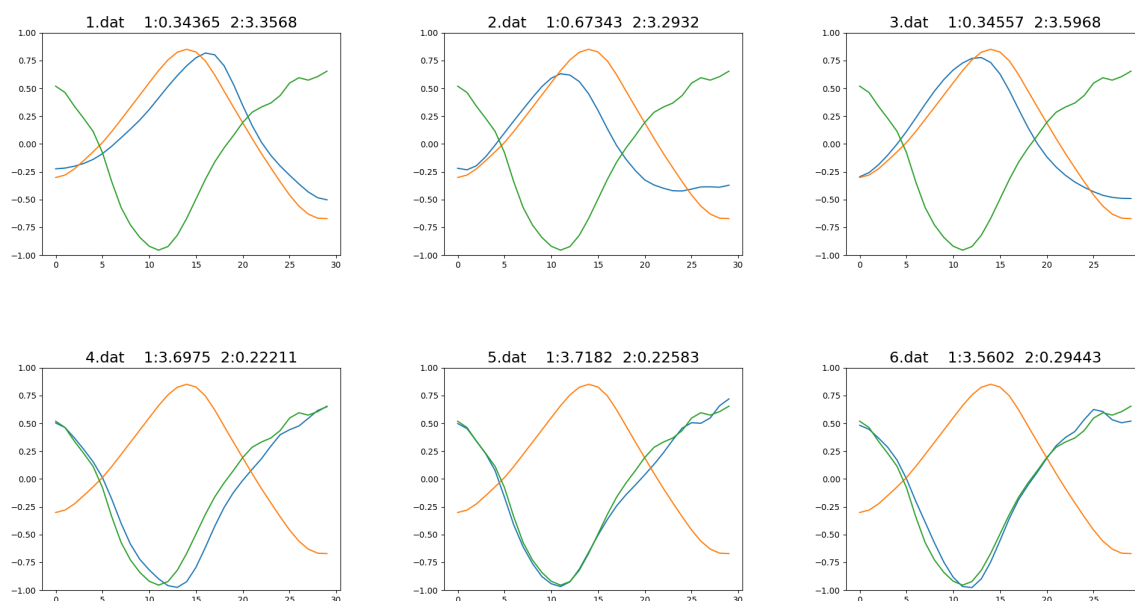


図 1: 各 reference のグラフ。青線が評価対象で、オレンジ線がクラス 1、緑線がクラス 2 を示しています。

Level2

この設問は Level1 で作成した自作dtw のコードを 3 次元ベクトルに対応できるように拡張したことで解決しました。結果から、test1,test3,test6 はクラス 1 に、test2,test4,test5 はクラス 2 に属することがわかりました。

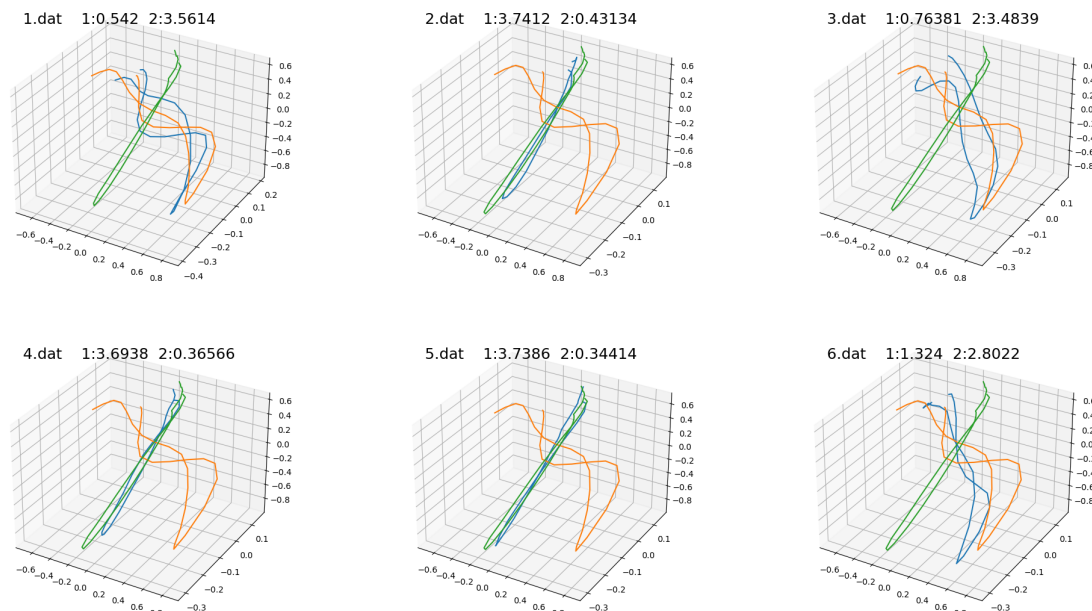


図 2: 各 reference のグラフ。青線が評価対象で、オレンジ線がクラス 1、緑線がクラス 2 を示しています。

Level3

この設問は Level2 で使用した自作dtw のコードにおいて、累積行列を時系列長の大きい方に合わせた正方行列にすることで、可変長でも対応できるように改変しました。結果から、test1,test3,test6 はクラス 1 に、test2,test4,test5 はクラス 2 に属することがわかりました。

なお、test3 と test5 の結果は Level2 のときと異なっています。これは Level3 のデータが Level2 のものと、test3 と test5 に限っては対応づいていないことが推測されます。(例えば Level3 の test3 のデータは Level2 の test3 のデータではなく、test5 をベースに改変されたものだと言グラフの形状から推測されます。逆も同じです。)

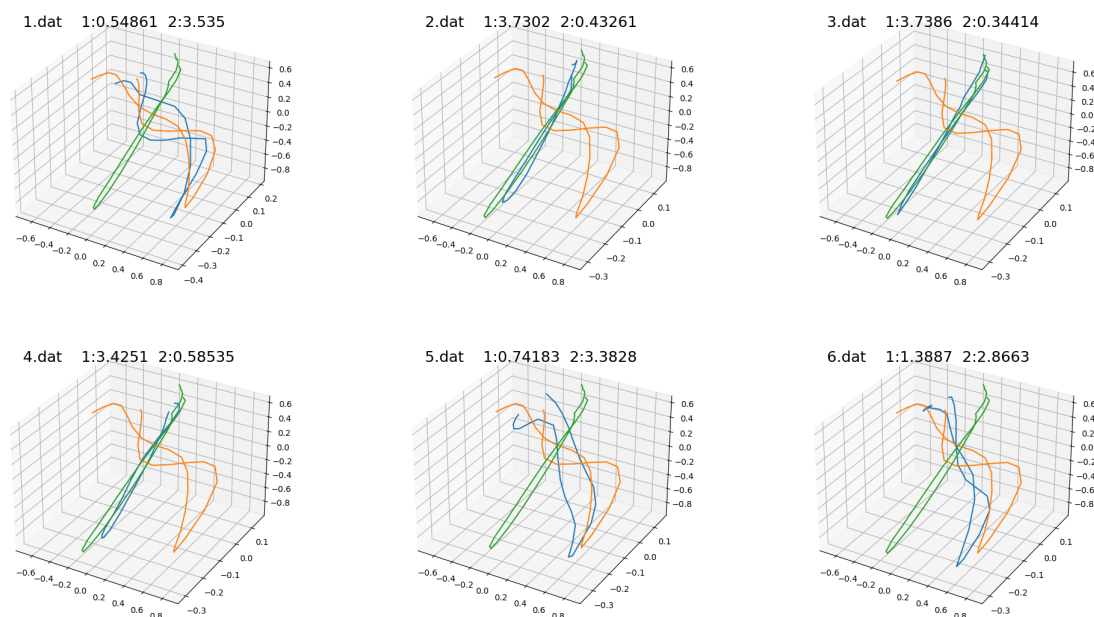


図 3: 各 reference のグラフ。青線が評価対象で、オレンジ線がクラス 1、緑線がクラス 2 を示しています。

Level4

これまでと同様にして考え、時系列長 256 の 64 次元データも扱えるようにプログラムを拡張しました。このプログラムの実行は数分程度で終わったため、特に次元削減や間引きなどの処理は行っていません。

このデータの分類を行うに当たって、各クラス 3 つのデータが用意されていたため、まずは各クラスのデータ平均をそれぞれ取得して、その値をクラス間で比較しました。以下の表に示すとおり、data1 から data7 及び data13 がクラス 1 に、data8 から data12 及び data14 はクラス 2 に属することがわかりました。

この他、 $k=3$ とする k 近傍法も試してみました。この分類結果は平均を用いた場合と変わりませんでした。

表 1: DTW の平均値による分類 (太字が該当クラス。小数点第3位以下は切り捨て)

	test1 average	test2 average
data1	12.468	12.861
data2	12.931	13.588
data3	10.386	11.996
data4	10.041	15.563
data5	10.661	14.449
data6	11.137	15.459
data7	11.227	13.167
data8	12.726	10.115
data9	20.307	18.842
data10	11.123	10.844
data11	13.092	9.869
data12	14.833	11.643
data13	21.166	21.435
data14	17.545	11.752

表 2: k 近傍法による分類 (太字が該当クラス。k=3)

	test1	test2
data1	2	1
data2	2	1
data3	2	1
data4	3	0
data5	3	0
data6	3	0
data7	2	1
data8	1	2
data9	1	2
data10	1	2
data11	0	3
data12	0	3
data13	2	1
data14	0	3