

S.D.A.

Rapport du projet 3

Pierre HOCKERS, Quentin LOWETTE, Olivier MOITROUX
ingénieur civil

19 mai 2017

1 Analyse théorique

1.1 DTW

1.1.1 Le principe

DTW (ou déformation temporelle dynamique en français) est un des principaux algorithmes utilisés pour mesurer la similarité entre deux séquences linéaires ayant des échelles temporaires différentes tel que, par exemple, la vitesse de deux marcheurs si l'on compare leurs pas ou encore la vitesse de locution de deux personnes si l'on compare des enregistrements vocaux. Pour cette dernière application, il convient d'éliminer les fluctuations non-linéaires pouvant intervenir entre l'échelle temporelle et la modélisation de la locution de l'enregistreur. L'algorithme DTW a alors pour fonction de trouver le couple optimal entre l'enregistrement et les échantillons dont il dispose en ignorant ces différences de vitesse et en prenant en compte des restrictions programmées.

Plus précisément, ces deux séquences sont modélisées (*wrapped*) selon une dimension temporelle pour pouvoir mesurer ensuite au mieux leur ressemblance sous forme d'une distance entre deux points, et ce, en prenant en compte les variations ayant lieu pour qu'elles n'impactent pas le résultat final. De plus, on produit un *wrapping path* permettant d'aligner temporellement les deux séquences. Un signal ayant des points de coordonnées initiales x, y se verra ainsi transformé en un signal (x_2, y) . Comme proposé dans l'énoncé, nous utiliserons la distance absolue moyenne :

$$d(s_1[i], s_2[j]) = \frac{1}{m} \sum_{h=1}^m |s_1[i]^{(h)} - s_2[j]^{(h)}| \quad (1)$$

1.1.2 Contraintes de localité

Cet algorithme se doit d'être en quelque sorte calibré pour fonctionner au mieux avec les données que nous lui fournissons. C'est ainsi que nous introduisons le paramètre (dit paramètre fenêtre) w afin de rajouter des contraintes locales. Celui-ci apparaît donc naturellement dans la liste de paramètre de notre fonction $dtw()$. Nous imposerons alors, si l'élément $s[i]$ de la séquence $s[1...N]$ à analyser a trouvé son correspondant $S[j]$ dans la base de donnée de référence S , que cette correspondance ne sera validée uniquement si la distance $|i - j|$ n'est pas plus grande que notre paramètre w . En l'occurrence, nous avons choisi de fixer la valeur de w à 10% soit $n/10$ dans le cadre de cette application. Les contraintes de localité permettent ainsi de fixer le décalage maximum entre deux équivalences.

1.1.3 Formulation récursive

L'algorithme dtw que nous allons implémenter pour ce projet prendra en compte des contraintes locales que nous fixerons à 10%. Cependant, pour rester général, nous allons nommer cette variable w (pour respec-

ter ce qui semble être la convention). En voici le pseudo code ¹ récursif suivant une programmation dynamique :

```
DYNAMICTIMEWARPING(record, sample, locality)
1  n = record.length
2  m = sample.length
3
4  Let dtw[0...n, 0...m] be a new array
5  w = max(locality, |n-m|)
6
7  for i = 0 to n
8      dtw[i, 0] =  $\infty$ 
9  for i = 1 to m
10     dtw[0,i] =  $\infty$ 
11 dtw[0,0] = 0
12
13 for i = 1 to n
14     for j = 1 to m
15         cost = dist(record[i], sample[j]) // absolute mean distance
16         dtw[i,j] = cost + min(dtw[i-1, j], dtw[i, j-1], dtw[i-1, j-1])
17
18 return dtw[n,m]
```

1.1.4 Analyse de la complexité

TODO

1.2 Découpage optimal

TODO

2 Analyse Empirique

2.1

2.2

Wikipedia

1. source wikipedia