Université Catholique de Louvain-la-neuve

LINFO2275 - Data Mining and Decision Making

Project 2 : Gestures Recognition

Group 30
Tchoupe Djatcheu Patrick
1844-2100 DATI2MS

Debongnie Nathan
8731-2200 DATI2MS

Nkweya Tofeun Constance Diana
0969-2200 DATI2MS

Professor : Marco Saerens
Teaching assistants : Sylvain Courtain and Pierre Leleux

May 2023
Academic year 2022 - 2023

1 Introduction

Gesture recognition plays a vital role in enabling natural and intuitive human-computer interaction. With the advent of smart user interfaces, accurately recognizing and interpreting hand gestures has become an essential component of various applications, ranging from virtual reality and augmented reality systems to sign language translation and interactive gaming. The objective of this project is to develop a sketch recognition system identifying the category of hand recorded sequences. More precisely, we implemented the Dynamic Time Warping (DTW) as a baseline method and the $1 Recognizer as a state-of-the-art approach. In this report, we will provide a theoretical explanation of both DTW and the $1 Recognizer, including their main equations and steps. We will describe our implementation of these algorithms and discuss the methodology employed for evaluation. We will finish by comparing both methods and identifying the best one.

2 Theoritical explanations

2.1 K-nearest neighbors (KNN)

The k-nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point.The main idea is to find the K nearest neighbors surrounding each unlabelled datapoint and infer the class label for these datapoints through majority voting. In order to determine which data points are closest to a given query point, the distance between the query point and the other data points will need to be calculated. These distance metrics help to form decision boundaries, which partitions query points into different regions.In the context of this project, we work with time series so, we implemented KNN using the Dynamic Time Warping as its metric.

2.2 Dynamic Time Warping (DTW)

Dynamic Time Warping is a dynamic programming algorithm used to measure the similarity between two temporal sequences, even when they exhibit temporal distortions or have different lengths.In the case of hand gesture recognition, each gesture is represented as a sequence of three-dimensional spatial coordinates over time.The basic idea behind DTW is to construct a two-dimensional matrix, called the DTW matrix, where each element represents the cumulative similarity between two corresponding sub-sequences of the input sequences. The goal is to find the optimal path through this matrix that minimizes the total distance or maximizes the similarity between the two sequences which is the equivalent to finding the optimum alignment between two time series sequences.In our case, by finding the optimal alignment, DTW captures the inherent temporal characteristics of the gestures and enables accurate recognition.

To compute the DTW, we need to take two time series whose points consist of vectors (which are represented in our case by the positions x, y and z). Then we compute the cost matrix M of the two time series to be compared. The cost matrix uses the formula mentioned below :

$$\mathbf{M(i,j)} = \|\mathbf{S_A(i)} - \mathbf{S_B(j)}\| + \mathbf{min}(\mathbf{M(i-1,j-1)}, \mathbf{M(i,j-1)}, \mathbf{M(i-1,j)})$$

where `M` is the matrix ; `i` is the iterator for serie S_A ; `j` is the iterator for serie S_B.

It's also important to notice that,some users may be slower than others when it comes to writing specific numbers that can be confusing. For example, the numbers 1, 2 and 7 can look the same when written. The number 1 could be confused with the number 7. To avoid such a situation, one solution would be to normalize the value of the DTW distance obtained. We can do this in two ways : The first is to divide the DTW distance by the path length obtained from the cost matrix. Another way would be to divide this distance by the sum of the lengths of the two sequences. In the end, we opted for the first way.

2.3 K-nearest neighbors (KNN) with DTW distance

As highlighted earlier, the (KNN) is a classification algorithm that can be used in combination with the Dynamic Time Warping (DTW) metric for gesture recognition. Here's how KNN based on the DTW metric works :

- DTW Calculation : first, the DTW distance between the test sample and each training sample is computed. DTW measures the similarity between two time series by finding the optimal alignment between them, accounting for possible temporal distortions or variations.

- Nearest Neighbor Selection : The K training samples with the smallest DTW distances to the test sample are selected as the nearest neighbors. K is a user-defined parameter that determines the number of neighbors to consider.

- Voting and Classification : The class label of the test sample is determined through a voting process. Each of the K nearest neighbors contributes to the voting based on their class labels. The majority class among the neighbors is assigned as the predicted class label for the test sample.

Also,Choosing an appropriate value for K is crucial. A smaller K value may make the algorithm more sensitive to noise or outliers, while a larger K value may result in over-smoothing and decreased sensitivity to local variations. The optimal K value depends on the characteristics of the dataset and can be determined through experimentation or cross-validation.

2.4 $P Point-Cloud Recognizer

For this project, we decided to use the $P Point-Cloud Recognizer as our state-of-the-art method. $P is member of the $-familly approaches which are gesture recognition approaches caracterized by their low-cost, easy to understand and implement and namely their high performance. They involve simple geometric computations and straight forward internal representations. The $P Point-Cloud Recognizer is a gesture recognition algorithm proposed by **Daniel Vogel** and **Patrick Baudisch**.

The task of the recognizer in $P is to match the point cloud of the candidate gesture (C) to the point cloud of each template (T) in the training set and compute a matching distance. As in the tradition of the Nearest-Neighbor approach, the template located at the smallest distance from C delivers the classification result.

By defining the matching between 2 points cloud C and T as a function $\mathbb{M}$ that associates each point $C_i \in C$ with exactly 1 point $T_j \in T$, $T_j = \mathbb{M}(C_i)$, if one resamples both C and T (to uniformize both input data) into the same number of points n, then the matching will consist of exactly npairs of points. Inspired by the Euclidean sum of $1 and the Proportional Shape Distance of Shark [1] , the goodness of Matching $\mathbb{M}$ is defined as the sum of Euclidian distances of all the pairs of points from $\mathbb{M}$:

$$\sum_{i=1}^{n} \|C_i - T_j\| = \sum_{i=1}^{n} \sqrt{(C_i.x - T_j.x)^2 + (C_i.y - T_j.y)^2}$$

With the definition above, a point cloud recognizer need to search for the minimum matching distance between C and T from $n!$ possible alignments. Finding the best match in such a problem can be represented as a problem from graph theory [2] where the Hungarian algorithm [3] delivers the ideal minimum-cost matching performance but at high time complexity ($O(n^3)$). With this time complexity the so called Hungarian recognizer can hardly fits into $-family. To fit $-family The $P makes use of that Hungarian algorithm and Avis [4] to implement heuristics execution time named GREEDY-X (X=1..5) for matching clouds. To ease this subsection we refer the reader to [5] (section 3) for more details about different GREEDY-X. However it's worth mentionning that the purpose of those GREEDY-X is to reduce time complexity of the recognizer in order to fit best the $-familly. While GREEDY-2 and GREEDY-4 present complexity of $0(n^2)$, GREEDY-1 presents complexity of $0(n^2 * \log(n))$ and GREEDY-5 and GREEDY-3 present both complexity of $0(n^{2+\epsilon})$, with $\epsilon < 1$. Results conducted in [2] (Section 4) shows that GREEDY-5 with $\epsilon = 0.5$, named GREEDY-5(0.5) and denoted as $P recognizer, exibits better performance with highest accuracy.

3 Implementation

In order to facilitate the opening of the 1000 files used in this project, a class called Dataset has been implemented. This class has two parameters :

- elements : This parameter represents the list of labels in all project files.It is useful because it allows us to iterate easily on the different files and to store the dataset labels accordingly.
- path : This parameter represents the path of the folder that contains the 1000 files provided for the project. The files are stored in the same order as they appear in the folder. Each file is represented by a triplet of positions [x, y, z], and the time is not stored as it is not necessary.

The Dataset class provides two accessible lists : "data" and "labels". These lists contain the time series sequences and the corresponding labels for each sequence, respectively.

3.1 K-nearest neighbors with Dynamic Time Warping

The KNN_DTW class is designed to perform the KNN (K-Nearest Neighbors) algorithm on time series data using the DTW (Dynamic Time Warping) distance as the metric. The two key functions utilized by it :

- The "distance" function takes two vectors as parameters and computes the Euclidean distance between them.
- The "dtw_distance" function takes two sequences as parameters and calculates the DTW distance between the two sequences. Notably, this function utilizes the Euclidean distance obtained from the "distance" function to compute the distance between the three-dimensional vectors of each time series.

The KNN_DTW class takes a single parameter, which is the number of neighbors (represented by the variable "n_neighbors" within the class). The main functions implemented in this class are as follows :

- The "fit" function takes the training set, which consists of a list of time series sequences and their corresponding labels. This function sets the training set and the target values, which will be used later for predicting classes.
- The "predict" function requires one parameter, namely the test set, which is a list of unlabeled time series. This function computes the distance matrix between the training and test sets using the DTW distance metric and identifies the nearest neighbors based on the specified number of neighbors. Subsequently, it determines the majority votes among the labels to assign labels to each unlabeled observation/time series sequence.

3.2 $P Point-cloud Recognizer

Even though it would be possible to extend $P (a gesture recognition technique) to include 3D gestures by incorporating the z dimension, for the sake of simplicity in this project, we have chosen a different approach. We have performed a principal component analysis (PCA) on our 3D data to reduce it to 2D. This allows us to utilize existing libraries and pseudocode for 2D hand gesture recognition. The first principal component accounts for 98.9% of the total variability, which appears to be more than sufficient for obtaining reliable results.

4 Results comparison

We evaluated the accuracy of the different models by cross-validation with two different parameters : User-independent and user-dependent gesture identification.

For the KNN with DTW, we set the number of neighbors to 3.

4.1 User-independent setting

The user-independent evaluation quantifies the ability of identifying the gestures of a new unknwon user. The results of the two models can be seen in the tables below.

User	3-NN with DTW		$P Recognizer	
	Domain 1	Domain 4	Domain 1	Domain 4
1	0.73	0.55	0.89	0.63
2	0.72	0.38	0.94	0.6
3	0.89	0.6	0.91	0.75
4	0.72	0.71	0.94	0.6
5	0.69	0.7	0.98	0.7
6	0.78	0.52	0.97	0.65
7	0.73	0.46	0.91	0.78
8	0.86	0.51	0.9	0.67
9	0.64	0.6	0.9	0.71
10	0.32	0.43	0.91	0.53
Mean	**0.708**	**0.546**	**0.925**	**0.662**
Standard deviation	**0.147**	**0.103**	**0.029**	**0.071**

TABLE 1 – Table summarizing the accuracies, average and standard deviation for each model on the domain 1 and domain 4 - User independent setting

We observe that the accuracies of the $P recognizer are better than those of the 3-NN with DTW. Accuracy varies very little for P. We conclude from these results that the $P recognizer significantly better predicts the classes of a new user's data.

4.2 User-dependent setting

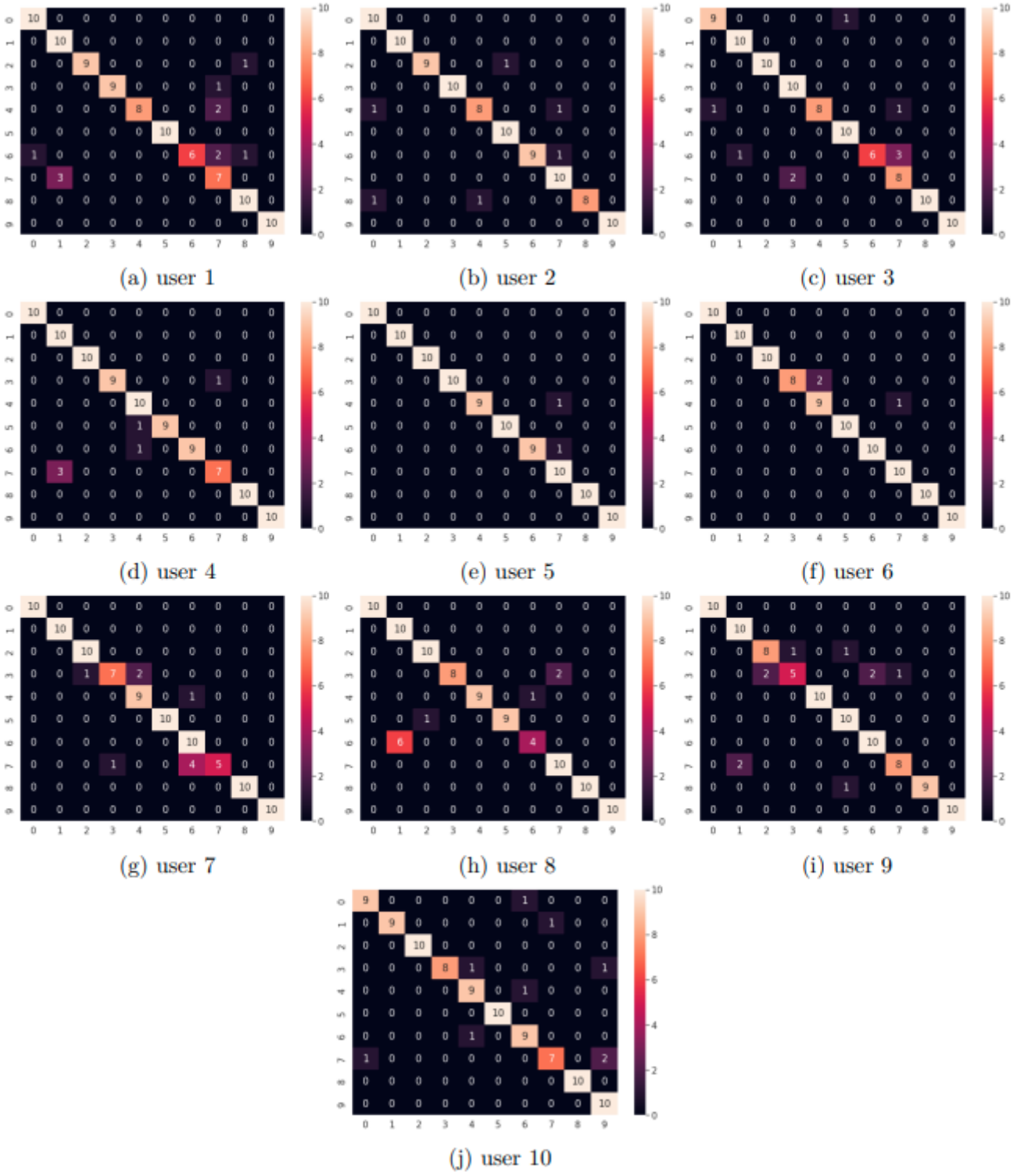
User	3-NN with DTW		$P Recognizer	
	Domain 1	Domain 4	Domain 1	Domain 4
1	0.92	0.84	0.09	0.16
2	0.97	0.94	0.05	0.08
3	0.98	0.98	0.12	0.11
4	1	0.96	0.08	0.12
5	1	0.99	0.1	0.05
6	0.98	0.94	0.1	0.04
7	0.96	0.97	0.14	0.09
8	1	0.98	0.11	0.09
9	0.99	0.95	0.12	0.11
10	0.99	0.96	0.11	0.09
Mean	**0.98**	**0.95**	**0.102**	**0.094**
Standard deviation	**0.023**	**0.040**	**0.023**	**0.03**

TABLE 2 – Table summarizing the accuracies, average and standard deviation for each model on the domain 1 and domain 4 - User dependent setting

The surprising results show that 3-NN with DTW performs better than the $P recognizer.

5 Confusion matrix of the best model

In the user independent setting, the $P dollar appears to be the best model compared to the 3-NN with DTW. Below are the confusion matrix of each user with the $P dollar model.



(a) user 1 (b) user 2 (c) user 3

(d) user 4 (e) user 5 (f) user 6

(g) user 7 (h) user 8 (i) user 9

(j) user 10

FIGURE 1 – Confusion matrix for user with $P dollar model on domain 1

6 Discussion of the results

After looking at the results by studying the confusion matrix of the classification results, we have observed the $P Point-Cloud Recognizer outperformed the KNN with DTW in terms of accuracy and robustness in user-independent settings. However, in user-dependent settings, KNN with DTW outperformed the $P Point-Cloud Recognizer in user-dependent setting.

We thus made this deduction :In the user-independent setting, the goal is to evaluate the ability of the models to generalize to unseen users. The $P Point-Cloud Recognizer, with its focus on geometric features and point cloud representation, may have a higher ability to generalize across different users.This could be due to the fact that geometric features capture more universal characteristics of the hand gestures, making it easier to distinguish between different gestures regardless of the user.On the other hand, in the user-dependent setting, the focus is on evaluating the models' performance on individual users.

The KNN with DTW approach, by considering the temporal alignment and shape similarity between gestures, may be better suited for capturing the individual characteristics of each user's hand movements. It takes into account the specific patterns and variations exhibited by a particular user, resulting in higher accuracy when applied to the same user's gestures.

Conclusion

In conclusion, this project aimed to develop a hand gesture recognition system for a smart user interface using data mining and statistical learning techniques. Two methods were implemented and evaluated : K-nearest neighbors (KNN) with Dynamic Time Warping (DTW) and the $P Point-Cloud Recognizer.The DTW algorithm proved to be effective in capturing the similarity between hand gesture sequences by aligning them in time and allowing for temporal distortion.

By using DTW as the distance metric in the KNN, we were able to classify new gestures based on their proximity to the training set.On the other side, The $P Point-Cloud Recognizer introduced a different approach by representing hand gestures as point clouds and computing the similarity between them based on geometric features. Through cross-validation experiments in both user-independent and user-dependent settings, we evaluated the performance of the implemented methods. The results indicated that the $P Point-Cloud Recognizer outperformed the KNN with DTW approach in terms of accuracy and robustness in user-independent settings.

However, in user-dependent settings,the KNN with DTW approach outperformed the $P Point-Cloud Recognizer.

Références

[1] Radu-Daniel Vatavu, Lisa Anthony, and Jacob Wobbrock. Gestures as point clouds : A $p recognizer for user interface prototypes. ICMI'12 - Proceedings of the ACM International Conference on Multimodal Interaction, pages 273–280, 10 2012.

[2] Wilson A. D. Wobbrock, J. O. and Y. Li. Gestures without libraries, toolkits or training : a $1 recognizer for user interface prototypes. UIST, pages 159–168, 07 2007.

[3] . Anthony and J. O. Wobbrock. $n-protractor : A fast and accurate multistroke recognizer. GI, pages 117–120, 2012.

[4] P.-O. Kristensson and S. Zhai. Shark2 : a large vocabulary shorthand writing system for pen-based computers. UIST, pages 43–52, 04 2004.

[5] J. Edmonds. Paths, trees, and flowers. Canad. J. Math. 17, page 449–467, 1965.