Pôle Projet Intelligence Artificielle & ML

Smart Chair

Eléonore Belloni
Ethan Bitton
Iman Tachrift
Lucas Tramonte
Manal Kermoss

February 2024 - June 2024

Contents

1 Introduction **2**
1.1 Presentation of the Client and the Company: CEA-List 2
1.2 Problem Statement 2
1.3 Project Objectives 2

2 Project Structure **3**
2.1 Deliverables 3
2.2 Gantt chart 3
2.3 Meeting Schedule 3
2.4 Risk Analysis 4

3 Team Structure **4**
3.1 Team Description 4
3.2 Collaborative Tools 5

4 State of the Art **5**
4.1 Supervised Classification 5
4.1.1 Support Vector Machine(SVM) 5
4.2 Domain Adaptation 5
4.2.1 Motivation 5
4.2.2 Key Concepts 6
4.2.3 Dictionary Learning 6
4.3 Optimal Transport Theory 7
4.4 Novelty Detection 9
4.5 Clustering 10
4.6 Dimensionality Reduction 11

5 Results of the work carried out **12**
5.1 Data Analysis 12
5.1.1 Data visualization 12
5.1.2 DaDiL-E 15
5.1.3 K-means 16
5.1.4 DBSCAN 16
5.2 About the distance in clustering 17
5.2.1 Novelty Detection 19

6 Exploration work on optimal transport theory **20**

7 Conclusion **21**
7.1 Final Thoughts and Future Perspectives 21
7.2 Added Value 22

8 Algorithm Overview **23**

1 Introduction

1.1 Presentation of the Client and the Company: CEA-List

The Laboratory for Integration of Systems and Technologies (LIST) at the French Atomic and Alternative Energies Commission (CEA) is a key player in advancing intelligent digital systems. Based in Saclay, Essonne, CEA LIST focuses on delivering technological advancements for industrial partners.

CEA LIST's research concerns diverse fields in applied sciences and technologies, focusing on optimizing production processes with advanced manufacturing techniques, developing specialized computer systems for embedded applications, leveraging data analysis and machine learning for data intelligence, and utilizing radiation technology for healthcare advancements. In particular, our client conducts research in Dictionary Learning in this departement.

1.2 Problem Statement

In today's modern society, where sedentary occupations have become the norm, the issue of prolonged sitting and poor posture in the workplace has emerged as a significant concern. The majority of individuals spend a substantial portion of their day seated at desks or in front of computers, often maintaining static postures for extended periods. However, this prevalent sedentary behavior is not without consequences. Research has increasingly linked prolonged sitting and poor posture to a various health problems. According to the World Health Organization, physical inactivity is the fourth leading risk factor for global mortality, contributing to an estimated 3.2 million deaths annually.

Within this context, our client expects us to work on the matter. He has provided to us a sensing mattress which provides data of seating postures of a subject. To tackle the problem of morphological differences in our subjects, we are going to exploit the domain adaptation method to generalize the detection process.

The client has provided us data recorded using the mattress for seven different **subjects**, that is people. For each subject, there are two types of data : **controlled data** and **spontaneous data**(i.e. uncontrolled data or continuous data). The controlled data are labeled, with each subject instructed to maintain specific positions. In contrast, the spontaneous data is unlabeled, with subjects free to sit as they wished, without any imposed positions.

1.3 Project Objectives

Quantify the bias induced by spontaneous data on the classifier. The primary objective of our project, as discussed with the client, is to train a dictionary for supervised classification using data from the source subjects. This involves training dictionary atoms that, through linear combination, form a customized classifier for the target subject. The accuracy of the classifier will be evaluated based on errors made on the controlled data from the target subject.

Research a method for identifying out-of-distribution postures. The second objective of our project is to research a method for identifying out-of-distribution postures. The client has requested us to visualize the distribution difference for a given user between their controlled data and their spontaneously acquired data. To achieve this, we can apply clustering methods and dimensionality reduction techniques. We would then be able to identify novel postures that are not represented in the labeled dataset.

Limit the impact of unknown postures in learning model customization coefficient. To mitigate the impact of out-of-distribution postures on the estimation of personalization coefficients, one potential strategy is to adjust the weights of samples from the target domain. In the current barycentric regression implementation, all target domain samples are given equal weight. However, by assigning lower weights to samples suspected to correspond to unknown postures, we can potentially improve estimation accuracy.

2 Project Structure

2.1 Deliverables

The client expects us to leverage the dictionary learning methods previously developed on randomly generated data and apply them to our dataset. More precisely, the client expects us to use some visualization tools in order to gain insights into the structure of our data. Deliverables should also include utilizing the DaDiL-E framework for learning and developing personalized classifiers through barycentric regression.

2.2 Gantt chart

Here is the last version of our Gantt chart.

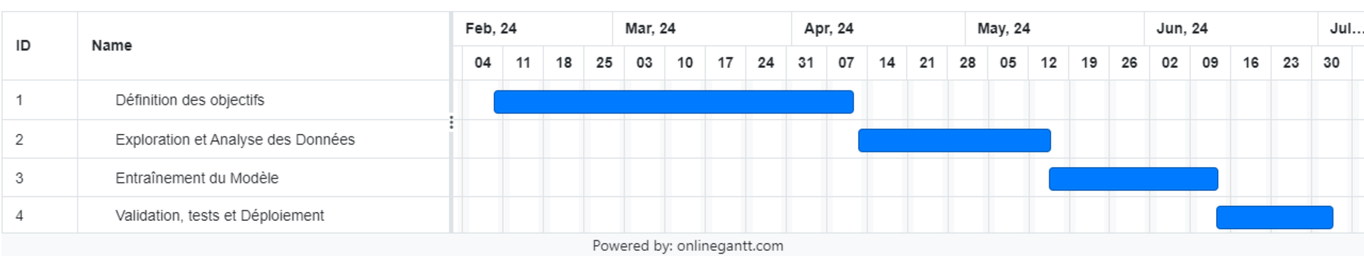


Figure 1: Gantt chart.

2.3 Meeting Schedule

Frequency of Client Meetings. At the start of the project, we met with the client once a month, due to communication problems. Subsequently, we met with the client every three weeks. We find useful to have these regular meetings, since the knowledge we have to acquire is complex and it is better for us to have more feedback.

Group Work Sessions. During the project sessions, we usually met in person. We found it better for communication. Outside of Pôle Projet classes, we maintain ongoing communication throughout the week to share progress based on assigned tasks.

2.4 Risk Analysis

Risk	Severity	Probability	Preventive/Curative Action
Poor communication with the client	High	Medium	Preventive: • Regular meetings • Clear documentation Curative: • Revise communication plan • Calling on the supervising professors
Poor communication within the group members	High	Low	Preventive: • Set clear roles • Use collaboration tools Curative: • Mediate conflicts • Reassign tasks as needed
Poor management of deadlines	High	Medium	Preventive: • Create a detailed timeline • Distribute work effectively Curative: • Redouble our efforts
Lack of expertise	Medium	Medium	Preventive: • Provide training • Establish a state of the art Curative: • Upgrade our skills • Consult supervising professors

Table 1: Risk Matrix

3 Team Structure

3.1 Team Description

Organizer. Iman Tachrift. Ensures deliverables are handed in on time. Updates the Gantt Chart and risk matrix.

Coordinator. Eléonore Belloni. Coordinates work among team members. Defines short-term objectives and help achieve them.

Propeller. Ethan Bitton. Encourages team members and stimulates the team on a regular basis. Keeps abreast of managerial and customer expectations.

Promoter. Lucas Tramonte. Dives into the bibliography and look for tools (pack- ages, modules, and methods) for implementation.

Explorer. Manal Kermoss. Delves into Novelty Detection. Also organizes meetings with supervisors.

3.2 Collaborative Tools

- Communication: Microsoft Teams, WhatsApp
- Document and Report Sharing: Overleaf | Latex
- Code Repository: https://github.com/nobelloni/Smart_Chair_mars - Private, access available upon request.

4 State of the Art

4.1 Supervised Classification

4.1.1 Support Vector Machine(SVM)

SVM is a class of supervised learning algorithms widely used for classification and regression tasks. SVM aims to find the hyperplane that best separates the classes by maximizing the margin between data points of different classes that are linearly separable. If the data are not linearly separable, we can introduce a soft-margin which uses a regularization parameter (C) to balance maximizing the margin and minimizing classification errors. We can also use kernel functions(polynomial kernel, sigmoid kernel etc.) to project the data into a higher-dimensional space where a hyperplane can be used for separation.

4.2 Domain Adaptation

Domain adaptation is a field of research in artificial intelligence which aims at tackling the challenge of transferring knowledge from a source domain to a target domain where data distributions may differ. In many machine learning scenarios, models trained on a particular dataset may not generalize effectively when deployed in an environment different from the one they were trained on.

For our project we will apply Domain Adaptation to distributions of sensor frame. Indeed each patient constitutes a distinct domain, due to its specific morphology and type of seat used.

4.2.1 Motivation

We can see bellow distributions from three different domains. We observe that the separating lines between the different labels(red or blue) are specific of each domain, such that no regression can offer an accurate generalization. Domain Adaptation is then relevant.

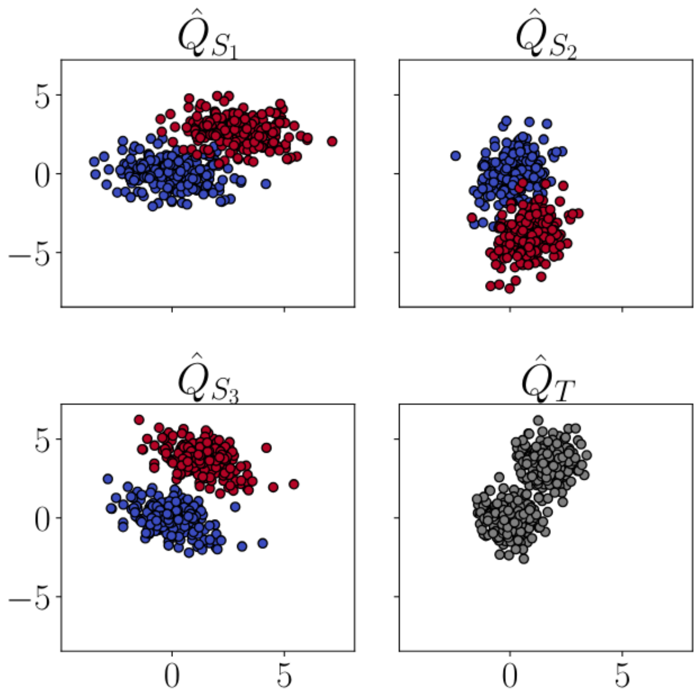


Figure 2: Set of datasets $Q = \{Q_s, Q_{S_2}, Q_{S_3}, Q_T\}$, where Q_T is the unlabeled target domain.

4.2.2 Key Concepts

Source Domain : This is the data distribution on which the model is trained using labeled examples. In our case, each dataset registered by one of the 7 subjects constitutes a source domain.

Target Domain : This is the data distribution on which a model pre-trained on a different domain is used to perform a similar task. The target domain is the dataset generated by a new subject.

Domain Translation : Domain Translation is the problem of finding a meaningful correspondence between two domains.

Domain Shift : A domain shift is a change in the statistical distribution of data between the different domains(like the training, validation, and(test sets) for a model.

4.2.3 Dictionary Learning

Dictionary learning is a branch of machine learning that aims at finding what is called a dictionary in which some training data admits a sparse representation. The sparser the representation, the better the dictionary. The dictionary consists of atoms, with each data distribution contained within the convex envelope of these atoms. Essentially, a dictionary is a set of atoms.

DaDiL-E. The Dictionary Adaptive Deep Image Learning Embedding, is a more specific version of DaDiL that focuses on integrating dictionaries into an encoding space to improve image representation.

The reconstruction of a new learner is illustrated in 3. We can see that the atoms are combined in order to reconstruct the data in the target domain.

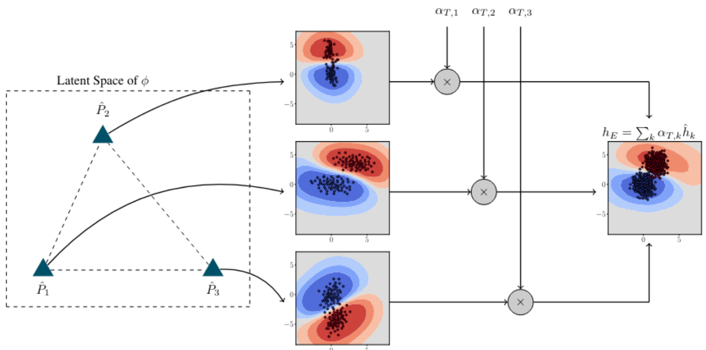


Figure 3: Conceptual Illustration of DaDiL-E

4.3 Optimal Transport Theory

Optimal Transport Theory, also known as optimal transportation theory, is a field of mathematics and economics that studies problems of resource allocation in transportation efficiently. The simplest form of the optimal transport problem is finding the most cost-effective way to transfer resources between a set of origins and a set of destinations while accounting for distribution costs. This can be used in a variety of fields, including as image processing, neurology, and logistics and economics.

One approach to developing a different concept of the distance between probability distributions is through the use of optimal transport theory.

Let Q and P be distributions with i.i.d samples $x_i(P) \sim P$ and $x_j(Q) \sim Q$ We can approximate P and Q empirically using mixtures of Dirac deltas [1]:

$$\hat{P}(x) = \frac{1}{n_p} \sum_{i=1}^{n_p} \delta(x - x_i^{(P)}) \tag{1}$$

We want to model distributional shift in Multi-Source Domain Adaptation. Each source domain is a labeled distribution:

$$\hat{Q}_l(x, y) = \frac{1}{n_{ql}} \sum_{i=1}^{n_{ql}} \delta((x, y) - (x_i^{(Q_l)}, y_i^{(Q_l)})) \tag{2}$$

And the target domain is an unlabeled distribution:

$$\hat{Q}_T(x) = \frac{1}{n_q} \sum_{i=1}^{n_q} \delta(x - x_i^{(Q)}) \tag{3}$$

As a solution to this problem, we want to learn a set of distributions $P = \{\hat{P}_k\}_{k=1}^K$ with free support $P = \{\ (x_i^{(P_k)}, y_i^{(P_k)})\}_{i=1}^{n_q}$, where $x_i^{(P_k)}$ denote the features and $y_i^{(P_k)}$ the labels.

We want to learn atoms $\mathbf{P}$, and a set of baricentric coordinates $A = \{\alpha_l; \sum_{k=1}^K \alpha_{l,k} = 1, \alpha_{l,k} \geq 0\}$, such that

$$W_c(\hat{Q}_l, \mathbf{B}(\alpha_l; P)) \tag{4}$$

with W_c the Wassertein distance and $\alpha_{l,k}$ the barycentric coordinate of the l-th dataset with respect to the k-th atom.

In order to create the variables that we will optimize, we can parametrize empirical distributions through their support:

$$\hat{P}_k(x, y) = \frac{1}{n_q} \sum_{i=1}^{n_q} \delta((x, y) - (x_i^{(P_k)}, y_i^{(P_k)})) \tag{5}$$

where $x \in \mathbf{R}^d$ is a feature vector, and $y \in \delta_{nc}$ is a vector containing the probability of a sample belonging to class $c = 1, \ldots, n_c$.

Now, let us analyze the loss function of DaDiL:

$$\mathcal{L}(\hat{Q}_l, \hat{B}_l) = \begin{cases} W_c(\hat{Q}_l, \hat{B}_l) & \text{if } \hat{Q}_l \text{ is labeled} \\ W_2(\hat{Q}_l, \hat{B}_l) & \text{otherwise} \end{cases} \tag{6}$$

If labels are available, we compute a supervised Wasserstein distance, which ground-cost:

$$C_{ij} = \left\| x_i^{(P)} - x_j^{(Q)} \right\|_2^2 + \beta \left\| y_i^{(P)} - y_j^{(Q)} \right\|_2^2 \tag{7}$$

otherwise, we calculate a standard W_2 between P and Q.

We can fit a SVM classifier on each atom:

$$\hat{h}_k = \arg\min_{h \in H} \hat{R}_{P_k}(h) \tag{8}$$

We weight the $\hat{h}_k$ using α_T, which gives:

$$\hat{h}_E(x_j^{(Q_T)}) = \sum_{k=1}^K \alpha_T \hat{h}_k(x_j^{(Q_T)}) \tag{9}$$

4.4 Novelty Detection

Novelty detection, also known as anomaly detection or outlier detection, is a critical task in machine learning and statistics that involves identifying unusual patterns or data points that differ significantly from the norm. Novelty detection methods are designed to learn a model from normal data and then detect instances that do not conform to this model.[2]

Feature Engineering : Effective novelty detection often relies on selecting relevant features that capture the essence of normal behavior and distinguish it from anomalous behavior.

One class svm

One-Class SVM (Support Vector Machine) is a powerful algorithm for novelty detection, particularly effective when the majority of data represents normal behavior and anomalies are infrequent. In our project, which involves monitoring seating postures using a pressure-sensing mattress, One-Class SVM proves to be highly useful. The pressure mattress collects data on pressure distribution, which we transform into a feature set representing typical postures.

To train the One-Class SVM, we use data from the controlled, labeled postures of the seven subjects. The objective of training is to define a decision boundary that encompasses the normal posture data, learning the underlying distribution of standard postures.

Data Preparation: The posture data is preprocessed to ensure consistency and reliability. Model Training: The One-Class SVM is trained using the preprocessed data to establish a boundary that includes the majority of normal data points. Detection Mechanism: Once trained, the One-Class SVM evaluates new data points by determining if they fall within the learned boundary. For novelty detection:

Novelty Prediction: The algorithm predicts whether each new posture is normal or novel based on its position relative to the decision boundary. Boundary Interpretation: Data points within the boundary are considered normal postures, while those outside are flagged as novel.

Autoencoders

An autoencoder is a type of artificial neural network used to learn a compressed representation of data. It consists of an encoder that maps input data to a lower-dimensional latent space, and a decoder that reconstructs the original data from this latent space.

To train the autoencoder, we used data from the seven correct postures. The goal of the training is to minimize the reconstruction error, which is the difference between the input data and the data reconstructed by the network.

- Data Preparation: The posture data is preprocessed to be normalized and standardized.
- Model Architecture: The autoencoder used consists of several dense layers, with a bottleneck layer representing the latent space.
- Cost Function: We use the Mean Squared Error (MSE) as the cost function to minimize.
- Training: The model is trained on the data from the seven correct postures over multiple epochs until convergence.

The autoencoder was trained on known sitting postures to learn their underlying distribution and reconstruct them accurately. By doing so, the autoencoder minimizes the reconstruction error for familiar postures. We then applied this trained model to continuous data to identify new postures. If a posture significantly differs from the known ones, the autoencoder fails to reconstruct it accurately, resulting in a high reconstruction error.

After training, the autoencoder is used for novelty detection:

- Calculating Reconstruction Error: For each new observed posture, we calculate the reconstruction error by passing the data through the autoencoder.
- Detection Threshold: A threshold is determined for the reconstruction error beyond which a posture is considered new or incorrect.
- Posture Classification: Postures with a reconstruction error below the threshold are considered correct, while those with a higher error are flagged as new.

4.5 Clustering

k-means

The primary objective of k-means is to partition a set of data into K clusters, where each observation belongs to the cluster with the nearest mean(centroid). The k-means algorithm follows an iterative process to assign data points to K clusters. The standard process consists of the following steps :

1. **Initialization**: Choose K initial centroids(randomly or using techniques like K-means++ to improve initialization).
2. **Cluster Assignment**: Assign each data point to the nearest cluster centroid.
3. **Centroid Recalculation**: Recalculate the position of the centroids as the mean of the points assigned to each cluster.
4. **Convergence**: Repeat steps 2 and 3 until the centroids no longer change significantly.

Although K-means is powerful, it presents several challenges. Indeed K-means is sensitive to outliers since they can disproportionately affect the centroids calculation.

Objective Function : Given a dataset $X = \{x_1, x_2, \ldots, x_n\}$ of n data points in a d-dimensional space, the K-means algorithm aims to minimize the within-cluster sum of squared distances:

$$J = \sum_{j=1}^{K} \sum_{i=1}^{N_j} \|x_i - c_j\|_2^2$$

DBSCAN(Density-Based Spatial Clustering of Applications with Noise)

DBSCAN identifies clusters in a dataset based on the density of data points. Unlike partitioning methods such as k-means, DBSCAN does not require the number of clusters to be specified beforehand and can identify clusters of arbitrary shapes.

More precisely, DBSCAN is partitioning the data between three different categories : the core points, the border points and the noise points(outliers).

For each point in the dataset, DBSCAN counts the number of points within a radius ϵ. If the count is at least $MinPts$, the point is labeled as a core point. For each core point, form a cluster by including all points within its ϵ neighborhood. Recursively include points from the ϵ neighborhood of each new core point added to the cluster. Points within the ϵ neighborhood of a core point but not themselves core points are labeled as border points. Points that are neither core nor border points are labeled as noise.

4.6 Dimensionality Reduction

PCA(Principal Component Analysis)

PCA is a technique of dimensionality reduction. It transforms a dataset into a set of orthogonal components that capture the maximum variance within the data, making it invaluable for simplifying complex datasets while preserving their essential structure.

PCA is a statistical procedure that uses an orthogonal transformation to convert a set of possibly correlated variables into a set of linearly uncorrelated variables called principal components. The steps involved in PCA are :

1. **Standardization** : Normalize the data to have zero mean and unit variance.
2. **Covariance Matrix Computation** : Calculate the covariance matrix to understand how variables vary with respect to each other.
3. **Eigen Decomposition** : Compute the eigenvalues and eigenvectors of the covariance matrix. The eigenvectors(principal components) define the directions of the new feature space, while the eigenvalues indicate the magnitude of variance along each principal component.
4. **Projection** : Project the original data onto the principal components to obtain a lower-dimensional representation.

t-SNE(t-distributed Stochastic Neighbor Embedding)

t-SNE is particularly effective at capturing the local and global structure of complex datasets, making it invaluable in exploratory data analysis, clustering, and pattern recognition. t-SNE aims at preserving the pairwise similarities between points as much as possible.

UMAP(Uniform Manifold Approximation and Projection)

UMAP is a nonlinear dimensionality reduction technique known for its ability to preserve both local and global structure in high-dimensional data, as t-SNE. It has gained significant popularity due to its effectiveness in discovering underlying patterns. UMAP tends to preserve global structure better, is more scalable, less sensitive to hyperparameters, produces deterministic embeddings, and offers greater flexibility compared to t-SNE. To summarize, if two points are close in their original space, UMAP will represent them as close in the resulting embedding space.

For example in NLP, UMAP can be used to visualize word embeddings and capture semantic relationships between words.

5 Results of the work carried out

5.1 Data Analysis

At the beginning of the project, our client provided us with a pressure mat from which data is collected.

This mat can be placed on any seating surface. The mat consists of a grid of pressure sensors, 16 by 16, totaling 256 pressure sensors. The pressure map thus varies depending on the seating surface, the patient, and the position taken by the patient. The pressure mat can be connected to a computer via a wired connection. From this computer, we use the Sensing Mat software to both visualize the pressure map in real-time for the first time and to record the data in JSON format.

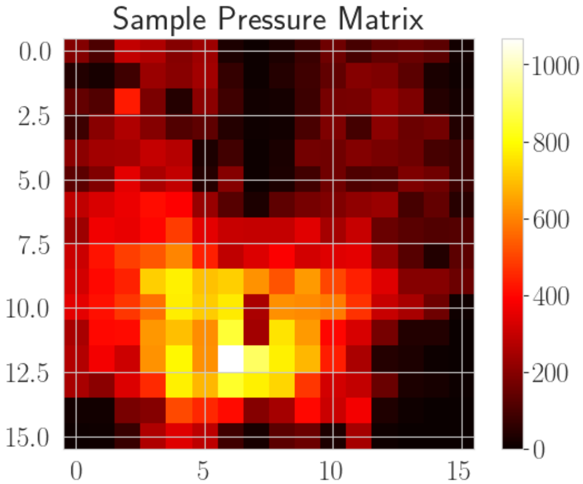
Thanks to the data provided by the client and the data recorded by the previous group, we initially had data from 7 patients. As described above, the data can be of two types, either spontaneous or controlled. For controlled data, the patients had to maintain each of the following 7 postures, one by one, for several tens of seconds :

1. "Correct" posture
2. Leaning forward with feet on the ground
3. Leaning back with right leg crossed
4. Leaning back with left leg crossed
5. Leaning forward with right leg crossed
6. Leaning forward with left leg crossed
7. Leaning back with firm backrest pressure

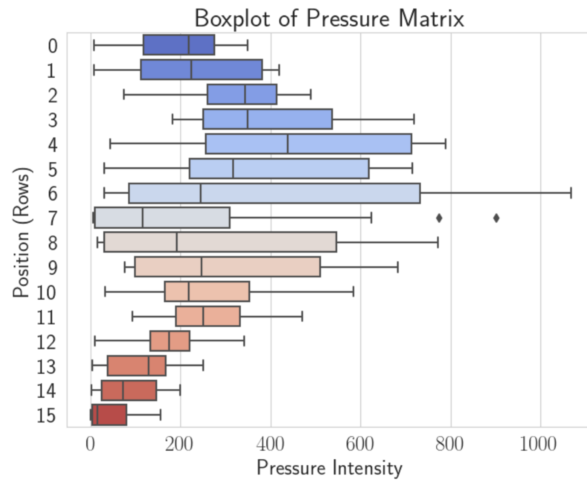
Moreover we have reused the experimental protocol used by the previous group by recording new data of ourselves. This time, we maintained the postures for several minutes to collect more data. This allowed us to have data from 2 additional subjects (3 and 6) to train the atoms. All of this recordings can be find in the assets folder of the Github repository.

5.1.1 Data visualization

A heat map was made for the first collection of the pressure matrix of Subject 1, and also a boxplot to analyze the pressure intensity at each position on the mesh, illustrated in the Figure 4



(a) Sample Pressure Matrix for Subject 1

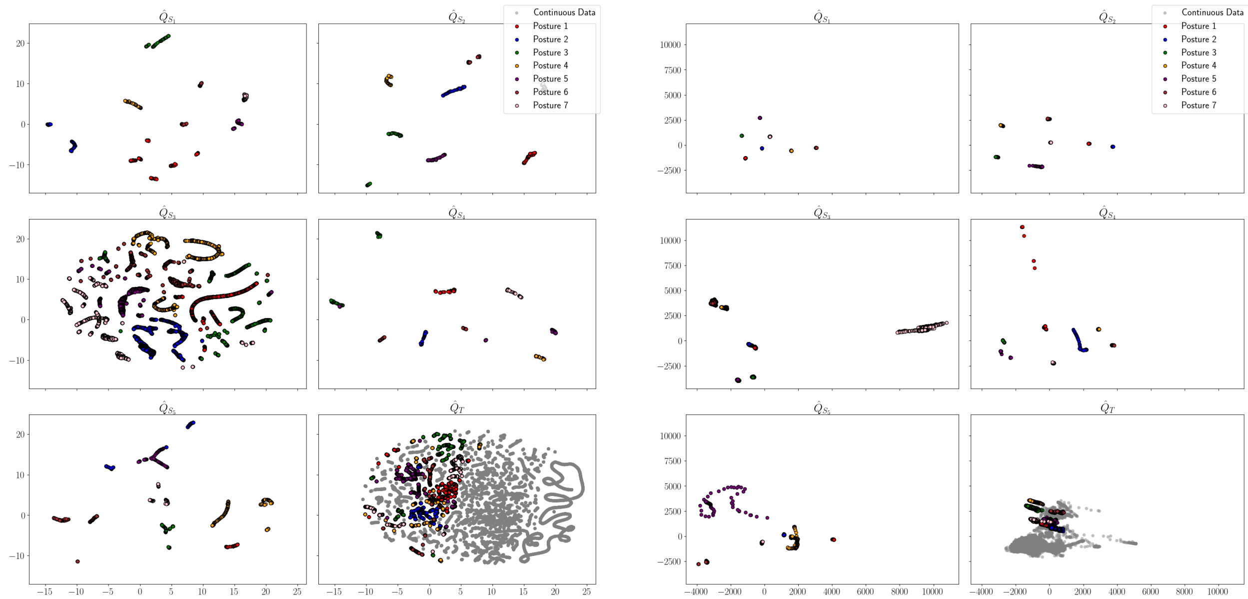


(b) Boxplot pressure matrix for Subject 1

Figure 4: Pressure Analysis for Subject 1

In the heatmap, colors represent the intensity of pressure values, mapping each value to a specific color. The colormap we utilized displays weaker pressure intensities with darker shades of red and stronger intensities with lighter shades of yellow.

As an example, we can visually observe in Figure 5 the PCA applied to the controlled and continuous data. The colors of the dots represent the 7 classes, and the first 5 subjects have been treated as sources while the sixth subject has been treated as the target domain, which is 'not' labeled. We observe here that we have a multi-class classification problem, thanks to the visible clusters.



(a) UMAP for 6 subjects

(b) PCA for 6 subjects

Figure 5: UMAP and PCA for 6 subjects

Moreover, it's important to note that we didn't have enough data for the multi-class classification task. We can see in Figure 6 that only for Subjects 3 and 6 (the data we collected) we do have a significant quantity of samples for each class.

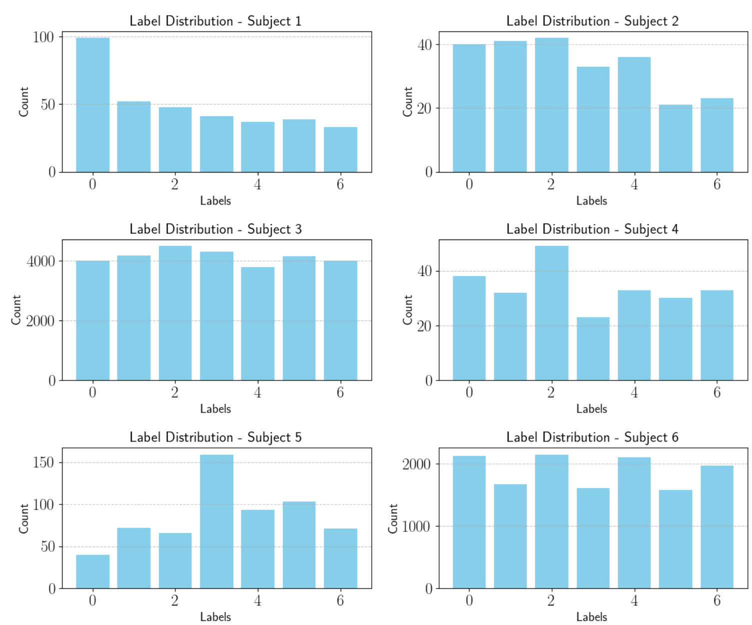
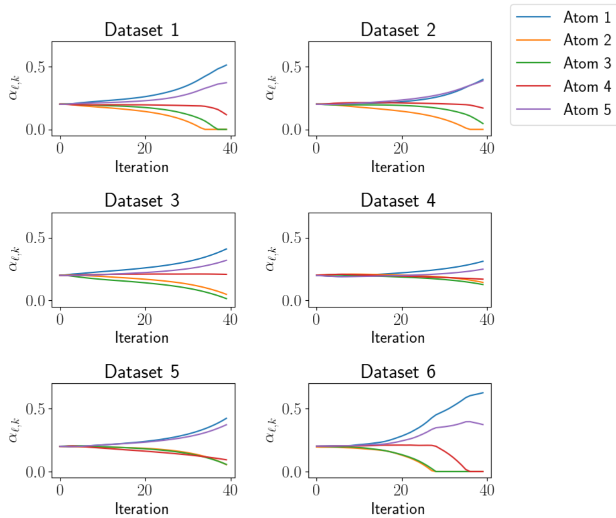


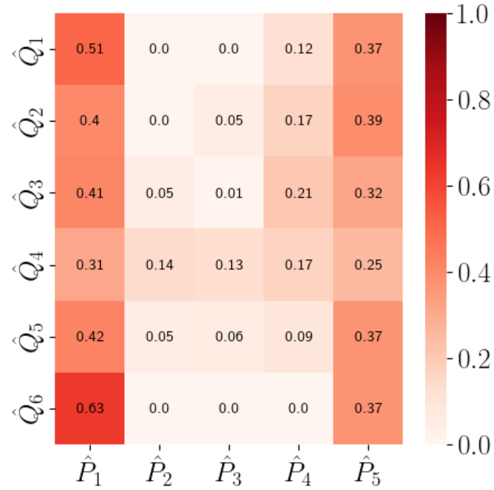
Figure 6: Posture distribution

5.1.2 DaDiL-E

We obtained a customized classifier for a target subject whose data has not been used to learn atoms. First, we can see the barycentric coordinates for each atom after the barycentric regression in Figure 7. As an example, we can see that dataset 6 has a distribution closer to atom 1 than to the other atoms.



(a) α_ℓ evolve throughout training



(b) α_ℓ for each atom (matrice A)

Figure 7: α_ℓ for each atom

Furthermore, after calculating the coefficients α_6 for the target domain, we used them to perform the personalized classification with a linear combination, which led to the following results:

```
Classification Report for Support Vector Machine:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00      2129
           1       0.00      0.00      0.00      1678
           2       0.00      0.00      0.00      2149
           3       0.00      0.00      0.00      1616
           4       0.00      0.00      0.00      2110
           5       0.00      0.00      0.00      1579
           6       0.15      1.00      0.26      1967

    accuracy                           0.15     13228
   macro avg       0.02      0.14      0.04     13228
weighted avg       0.02      0.15      0.04     13228
```

(a) SVC

```
Classification Report for Gradient Boosting:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00      2129
           1       0.00      0.00      0.00      1678
           2       0.00      0.00      0.00      2149
           3       0.45      1.00      0.62      1616
           4       0.00      0.00      0.00      2110
           5       0.32      0.80      0.46      1579
           6       0.00      0.00      0.00      1967

    accuracy                           0.22     13228
   macro avg       0.11      0.26      0.15     13228
weighted avg       0.09      0.22      0.13     13228
```

(b) Gradient boosting

Classification Report for Random Forest:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	2129
1	0.00	0.00	0.00	1678
2	0.00	0.00	0.00	2149
3	0.00	0.00	0.00	1616
4	0.49	0.98	0.65	2110
5	0.49	1.00	0.66	1579
6	0.00	0.00	0.00	1967
accuracy			0.28	13228
macro avg	0.14	0.28	0.19	13228
weighted avg	0.14	0.28	0.18	13228

(a) Random Forest

Classification Report for Logistic Regression:

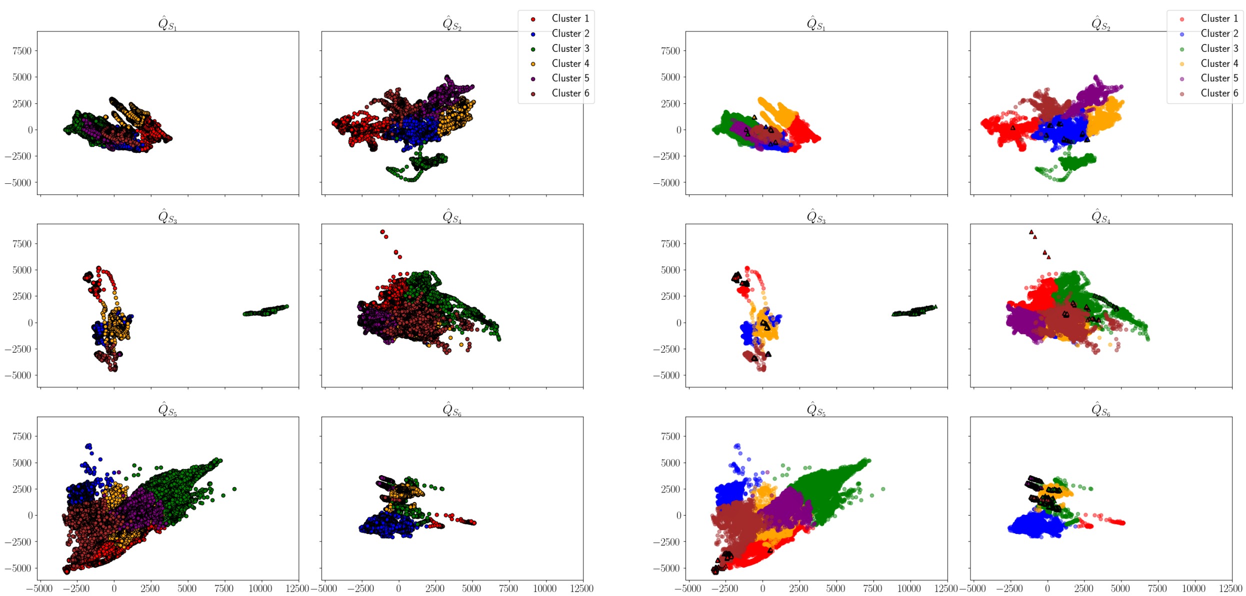
	precision	recall	f1-score	support
0	0.00	0.00	0.00	2129
1	0.00	0.00	0.00	1678
2	0.00	0.00	0.00	2149
3	0.00	0.00	0.00	1616
4	1.00	0.01	0.02	2110
5	0.14	1.00	0.24	1579
6	0.00	0.00	0.00	1967
accuracy			0.12	13228
macro avg	0.16	0.14	0.04	13228
weighted avg	0.18	0.12	0.03	13228

(b) Logistic regression

The main factor that has led to the model having difficulties in making predictions is the large amount of continuous data that belongs to new, unidentified postures.

5.1.3 K-means

An unsupervised machine learning algorithm (k-means) was used to group the data into clusters. Figure 10 (a) shows the clusters of all the data, and Figure 10 (b) shows the controlled data in the form of triangles surrounded by the continuous data.



(a) K means (b) K means with controlled data

Figure 10: K means with PCA

5.1.4 DBSCAN

As the k-means version forced us to decide of a number of clusters before using it, we decided to try another method of unsupervised machine learning clustering named DBSCAN. This model is based on density and it allows us to compare it to the Kmeans clustering. For now our values of ϵ were selected arbitrary that's why our number of cluster is absurd in the fifth graph. We will discuss this choice later.

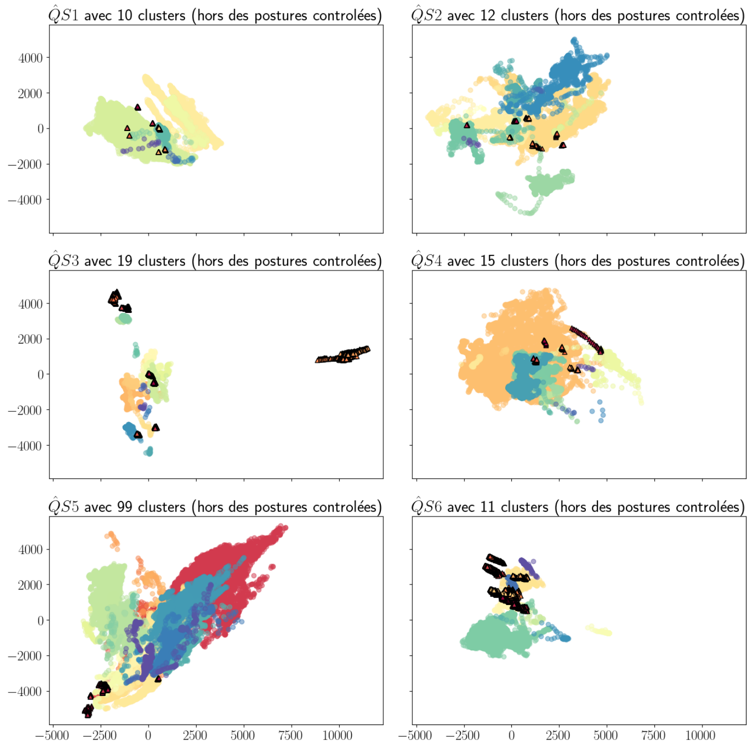


Figure 11: DBSCAN and PCA visualization

5.2 About the distance in clustering

The two method we use are based on comparing centroids or different points to a parameter via a distance. The thing is : this distance is the norm

$$||.||_2 = \sqrt{p_1^2 + \ldots + p_n^2}$$

. The thing is with this norm two completely different heatmaps can be at the same distance to another one and if we take the same posture but we multiply every coefficients by λ we obtain a new heatmap even further away. for example in 3 dimensional space :

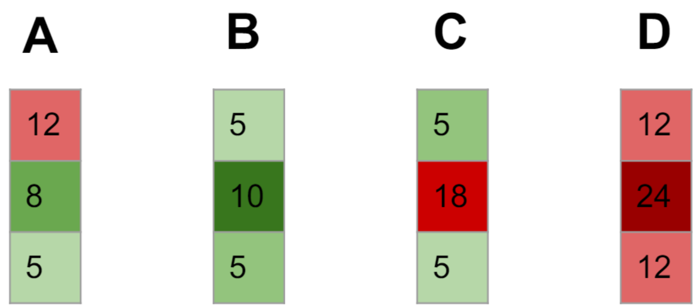


Figure 12: Exemple in 3D

We see that the 2 colinear vectors (B and D) are far away from each other compare to other postures. (d(A,B) 12 and d(B,D) 16).

Hence we have decided to try out different other metrics. The idea is the following : we consider that two different subjects with the same posture generate 2 pressure vectors which are colinear. And so we tried out the Frobenius Norm wich is the norm 2 of the generalisation of the cross product in N dimensions. This way 2 colinear vectors will have a distance equal to 0 and otherwise we can take the norm of the Frobenius product and normalized it by the norms of the two initials vectors. We have implemented it but the results were disapointing :

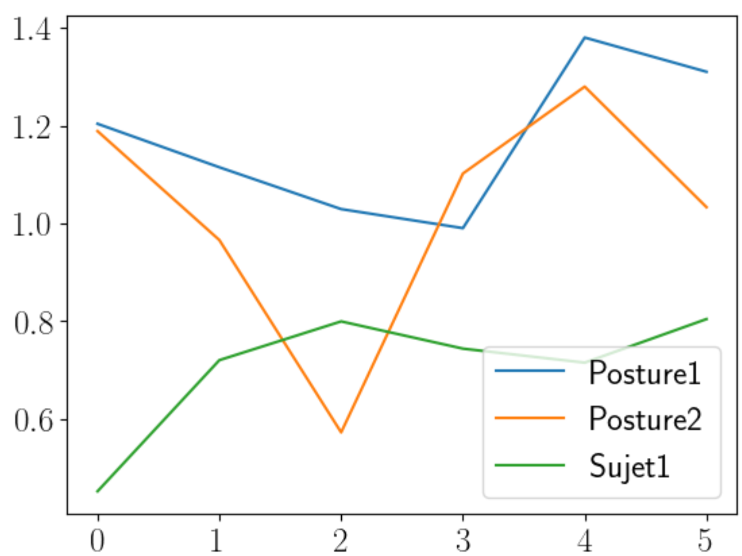


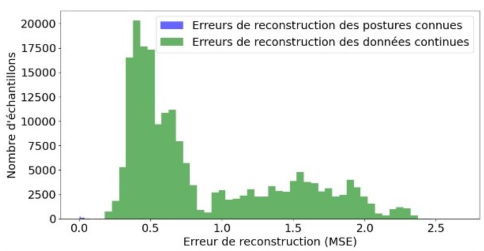
Figure 13: Implementation of new metric

Indeed we had the distance between the posture 1 for subject 1 and subject 2 equal to 1,2 and the one between posture 1 et 2 for subject 1 is 0,95. This shows that for now our method is noneffective but it might works better if we did only consider the high pressure zones of every pressure matrix. Otherwise 2 postures with the same subject will always be closer because most of the pressure matrix between different posture of a subject is similar in the low pressured area. We did also tried distance using covariance and standard deviation without much success...

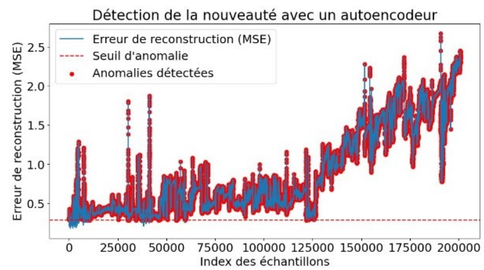
5.2.1 Novelty Detection

a - Novelty Detection with autoencoders

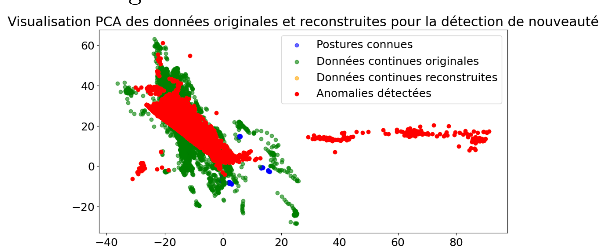
The results below show how well the autoencoder trains on the recorded postures. New or incorrect postures produce a significantly higher reconstruction error, allowing for reliable novelty detection.



(a) Reconstruction Error Histogram



(b) Novelty Detection with autoencodeurs



(c) PCA visualisation of Novelty Detection with autoencodeurs

Figure 14: Novelty Detection with autoencoders

The Reconstruction Error Histogram shows that the reconstruction error of the posture data revolves around a low value comparing to continuous data, which is normal because the autoencoder is trained to better reconstruct these data. This indicates that the model has reconstructed the known postures well. In other words, the autoencoder successfully learned the distribution of known sitting postures and detected new postures among the continuous data. The use of PCA showed us that the detected anomalies (red points) are postures that are not well reconstructed by the autoencoder, meaning they differ from the postures on which the autoencoder was trained.

b - Novelty Detection with one class svm

The graph generated from the One-Class SVM analysis provides valuable insights into the distribution of our data and the model's performance in novelty detection. In here we focused on subjet number 1. In the graph, the majority of the normal (inlier) data points are clustered within a defined boundary, which represents the decision function of the One-Class SVM. This boundary is typically shaped to encompass the high-density regions of the normal data, thereby capturing the common patterns and variations inherent in the seating postures. The novelties, or outlier data points, appear outside this boundary, indicating deviations from the normal postures except some of them that were inside which indicates a potential error in our program. The visualization shows that while many of the novel data points are situated near the edge of the normal data cluster, some are dispersed further away. This dispersion suggests that the One-Class SVM is capable of detecting a range of novelties, from those closely related to the normal data to those significantly different.

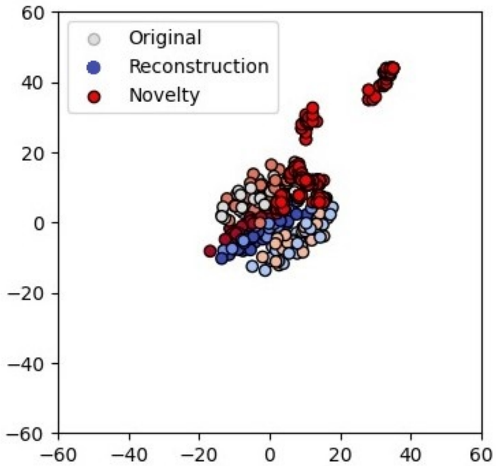


Figure 15: Novelty Detection with Class way svm

6 Exploration work on optimal transport theory

We have decide to implement the theorie in an algorithm in which we compare gaussian distributions randomly generated. Thanks to the ot library we can generate cost-matrix between two distributions and, by comparing the norms, we can obtain a distance between two distribution.

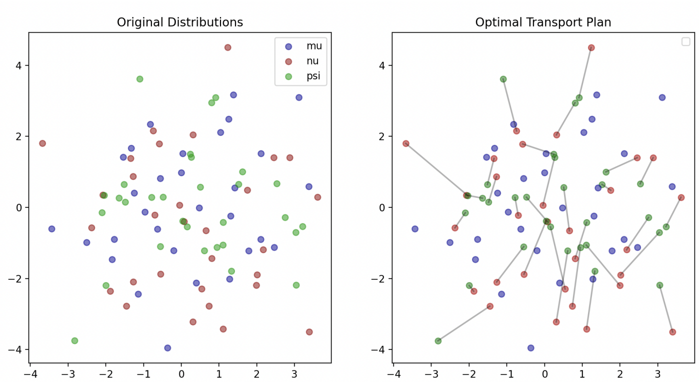


Figure 16: Example of an output where the closest distributions are linked

We then did the same thing with 3D gaussian distributions, and plot them as heatmaps to link with our project.

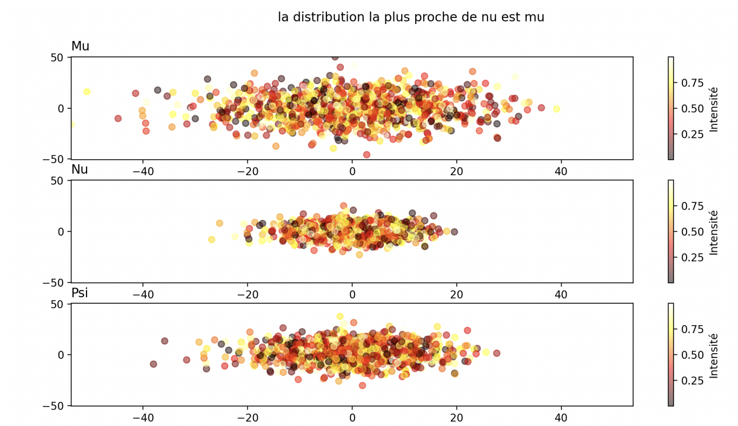


Figure 17: Example of an output with heatmaps

The goal of this work was to get better how to use the ot library and how the theorie could help us.

7 Conclusion

7.1 Final Thoughts and Future Perspectives

At first, the group dedicated itself to understanding the problem and the existing methods in the literature for the desired solution. During the structuring and planning phase, we focus our energy on learning about existing theories, and it was crucial for us to meet with the client. Indeed his knowledge of the theories of Dictionary Learning and Domain Adaptation is precious for us to understand the expectations and to work effectively on the project.

Overall, we have successfully cleaned the data and recorded new data over a longer period, which will enable us to achieve more accurate results. We implemented the DaDiL-E method on our data (both from previous groups and our own) and obtained preliminary results on the model's accuracy. Additionally, we developed visualization tools, primarily using UMAP and PCA, which allowed us to observe the outcomes of our clustering. We clustered our distributions to identify differences between controlled and uncontrolled data. Moreover, we worked on several novelty detection algorithms. The first method of novelty detection stemmed from our clustering, while we also explored other algorithms for this purpose. Moving forward, it will be essential to continue improving both clustering and novelty detection to adjust the training coefficients of the distributions and observe the impact on dictionary learning outcomes. You can find bellow some potential steps that we find could add a significant value to the project.

Use some labels in the target user's data. At this level, it will be a question of moving to a semi-supervised approach, by integrating into the strategy adopted in the second task, the knowledge of a few labels in the target user's data.

Register new distributions. Currently, the database includes controlled and spontaneous distributions from six distinct subjects. Despite the acquisition of new data, there is potential for enrichment by capturing new spontaneous data, now labelled. Consequently, the creation of an 8th category, called "Non-distribution," becomes necessary.

Development of a new metric When applying our clustering algorithms, we rely on Euclidean distance. However, it can be biased, sometimes considering two different postures of the same subject closer than the same posture of two distinct subjects. To address this, we propose using covariance or the vector product to compare distributions.

7.2 Added Value

Our project offers significant methodological value for our client, prioritizing it over immediate practical applications such as improving sitting posture. As our client operates within a research laboratory, our work gives greater importance to the methodology of the research. The application to sitting postures provides an opportunity to explore the impact of unclassified postures, contributing valuable insights into the DaDiL-E methodology. Ultimately, our aspiration is for this project to establish itself as a model for other innovative applications across diverse fields.

The value added to the project also lies in our work on Novelty Detection and visualization, which complement the initial group's work.

Finally, the project has provided not only added value for the client but also significant benefits for us. Our group gained valuable experience from this project. We were excited and satisfied to work in an area of artificial intelligence that was entirely new to us. None of us had previously studied or were familiar with Domain Adaptation, Optimal Transport and Novelty Detection. We were also pleased to engage in research work, and we extend our gratitude to our client, Fred Ngole Mboula, for sharing his expertise with us, assisting us with our challenges, and introducing us to his research topic. It was a challenge for us, but we are happy to have made concrete progress. Additionally, we discovered new data visualization techniques such as UMAP, and clustering methods like DBSCAN. This project met our expectations by allowing us to deepen our knowledge and implement practical solutions. This was our primary reason for choosing the Artificial Intelligence track, and we are very happy with our choice. We also thank our supervisors, Jean Philippe and Wassila, for their guidance.

References

[1] Eduardo Fernandes Montesuma, Fred Mboula, and Antoine Souloumiac. *Multi-Source Domain Adaptation Through Dataset Dictionary Learning in Wasserstein Space*. 09 2023.

[2] Stephen Marsland. Novelty detection in learning systems. 2003.

8 Algorithm Overview

Algorithm 1 Labeled Wasserstein Barycenter

Input: $\{X^{(P_k)}, Y^{(P_k)}\}_{k=1}^K$, $\alpha \in \Delta_K$, $\tau > 0$, N_{itb}.
for $i = 1, \dots, n_B$ **do**
    $x_i^{(B)} \sim \mathcal{N}(\mathbf{0}, I_d)$, $y_i^{(B)} = \text{randint}(n_c)$
end for
while $|J_{it} - J_{it-1}| \geq \tau$ and $it \leq N_{itb}$ **do**
    for $k = 1, \dots, K$ **do**
        $\pi^{(k,it)} = \text{OT}(X^{(P_k)}, Y^{(P_k)}); (X_{it}^{(B)}, Y_{it}^{(B)})$
    end for
    $J_{it} = \sum_{k=1}^K \alpha_k \langle \pi^{(k,it)}, C^{(k)} \rangle_F$
    $X_{it+1}^{(B)} = \sum_{k=1}^K \alpha_k T_{\pi^{(k,it)}}(X_{it}^B)$
    $Y_{it+1}^{(B)} = \sum_{k=1}^K \alpha_k T_{\pi^{(k,it)}}(Y_{it}^B)$
end while
Output: Labeled barycenter support $(X^{(B)}, Y^{(B)})$.

Algorithm 2 Your Algorithm

Input: $Q = \{Q^\ell\}_{\ell=1}^N$, number of iterations N_{iter}, number of atoms K, number of batches M, batch size n_b, learning rate η
Initialize $x_j^{(P_k)} \sim \mathcal{N}(0, I_d)$, $a_\ell \sim \mathcal{N}(0, I_K)$
for it $= 1, \dots, N_{\text{iter}}$ **do**
    for batch $= 1, \dots, M$ **do**
        for $\ell = 1, \dots, (N_S + 1)$ **do**
            Sample $\{x_1^{(Q_\ell)}, \dots, x_{n_b}^{(Q_\ell)}\}$
            if Q_ℓ is labeled **then**
                Sample $\{y_1^{(Q_\ell)}, \dots, y_{n_b}^{(Q_\ell)}\}$
            end if
            for $k = 1, \dots, K$ **do**
                Sample $\{(x_1^{(P_k)}, p_1^{(P_k)}), \dots, (x_{n_b}^{(P_k)}, p_{n_b}^{(P_k)})\}$
                Change variables $y_j^{(P_k)} = \text{softmax}(p_j^{(P_k)})$
            end for
            Calculate $X^{(B_\ell)}$, $Y^{(B_\ell)} = B(\alpha_\ell; P)$
        end for
        $L = \frac{1}{N} \sum_{\ell=1}^N \mathcal{L}(\hat{Q}_\ell, \hat{B}_\ell)$
        $x_j^{(P_k)} \leftarrow x_j^{(P_k)} - \eta \frac{\partial L}{\partial x_j^{(P_k)}}$
        $p_j^{(P_k)} \leftarrow p_j^{(P_k)} - \eta \frac{\partial L}{\partial p_j^{(P_k)}}$
        $\alpha_\ell \leftarrow \text{proj}_{\Delta_K}\left(\alpha_\ell - \eta \frac{\partial L}{\partial \alpha_\ell}\right)$
    end for
end for
Output: Dictionary $P^\star$ and weights $A^\star$