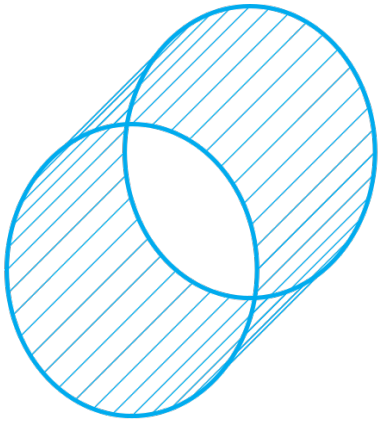


Khoa Toán - Tin học
Fac. of Math. & Computer Science

XỬ LÍ ĐA CHIỀU

TIME SERIES CLUSTERING

Giảng viên:
TS. Ngô Minh Mẫn
Khoa Toán - Tin học

Nhóm:
Vòng Vĩnh Phú - 19110413
Đỗ Hữu Quân - 19110160
Võ Huyền Bảo Hân - 19110303
Lê Huy Hoàng - 19110319

Ngày 9 tháng 6 năm 2022

Mục lục

1 Giới thiệu bài toán

1.1 Giới thiệu vấn đề

Trong bài báo cáo này, chúng ta sẽ tập trung vào các phương pháp xử lí, trực quan và phân nhóm dữ liệu chuỗi thời gian. Đây là một vấn đề tương đối nền tảng và thực tế trong lĩnh vực xử lí dữ liệu đa chiều, hỗ trợ cho các bài toán unsupervised learning liên quan đến phân loại và dự đoán hành động.

1.2 Giới thiệu bộ dữ liệu

Dữ liệu sử dụng là một Dataset cho việc nhân diện ADL với Wrist-worn Accelerometer. Đây là một bộ dữ liệu công khai với các bản ghi dữ liệu của gia tốc kế được gắn nhãn được sử dụng để tạo và xác nhận các mô hình gia tốc của ADL đơn giản. Mỗi điểm dữ liệu sẽ gồm dữ liệu 3 trục X, Y, Z trên một chuỗi thời gian, mô tả về hành vi của con người trong một khoảng thời gian được đo bởi 1 thiết bị đeo tay.

Cụ thể bộ dataset bao gồm các bản ghi của 14 hoạt động thường nhật với các labels (brush_teeth, climb_stairs, comb_hair, descend_stairs, drink_glass, eat_meat, eat_soup, getup_bed, liedown _bed, pour_water, sitdown_chair, standup_chair, use_telephone, walk) được hoàn thiện bởi 16 tình nguyện viên.

Phân bố dữ liệu các hoạt động không đều nhau (Imbalance Data), cụ thể thì số lượng bản ghi lớn nhất có thể có trong một hoạt động là 102 bản ghi (brush_teeth) và số lượng nhỏ nhất là 3 bản ghi (eat_meat). Ngoài ra, Các điểm dữ liệu có số time step chênh lệch nhiều. Cụ thể thì số time step dao động từ khoảng 100 đến 3500 time step.

2 Xử lí data

2.1 Xử lý số chiều dữ liệu

Sau khi thực hiện chuẩn hóa bộ dữ liệu của toàn bộ dữ liệu, nhóm đã thực hiện một số cách xử lí dữ liệu

1. Flatten toàn bộ Dataset và sử dụng mỗi time step như 1 điểm dữ liệu
    - Điểm mạnh: giải quyết vấn đề khác biệt về số time step trong các điểm dữ liệu
    - Điểm yếu:
        - Không thể hiện được đặc trưng về biến động của mỗi label
        - Mô hình trở nên nặng và hình ảnh plot bị rối không cần thiết.
2. Sử dụng DTW (dynamic time warping)
    - DTW là thuật toán cho phép đánh giá khoảng cách giữa 2 chuỗi thời gian dựa thông qua việc hình thành ánh xạ giữa chúng.
    - Điểm mạnh: giải quyết vấn đề khác biệt về số time step trong các điểm dữ liệu
    - Điểm yếu:
        - Độ phức tạp quá cao so với các phương pháp khác, thời gian chạy quá lâu nhưng không đem lại hiệu quả tương xứng.
        - Chỉ cho ra khoảng cách giữa các chuỗi thời gian, không thể áp dụng PCA hay một số thuật toán giảm số chiều tương tự
        - Đôi khi không nhận biết được 2 điểm dữ liệu cùng label. Lý do là vì thuật toán quá tập trung vào yếu tố số lần lên xuống của chuỗi thời gian. Nói một cách dễ hiểu, DTW sẽ xem 2 chuỗi thời gian của hành động đi bộ có khoảng cách xa nhau nếu số bước đi thực hiện trong chuỗi thời gian 1 lớn hơn nhiều so với chuỗi thời gian 2.
3. Nối dài các chuỗi thời gian ngắn bằng việc bổ sung phần khuyết bằng số 0. Sau đó sử dụng Euclidean metric.
    - Điểm mạnh: Thuật toán tương đối đơn giản.
    - Điểm yếu:
        - Không giải quyết được vấn đề chênh lệch số time step.
        - Thường không phát hiện được 2 chuỗi có cùng label.
4. Sử dụng khai triển Fourier để đưa về frequency domain.
    - Điểm mạnh: Nhận diện được một số đặc trưng của chuỗi thời gian.
    - Điểm yếu: Không giải quyết hoàn toàn việc chênh lệch time step.

Thông qua phân tích và thử nghiệm các phương pháp đã nêu, nhóm quyết định chọn phương án sử dụng khai triển Fourier để đưa các điểm dữ liệu về frequency domain.

Mô tả phương pháp sử dụng khai triển Fourier

Về mặt ý tưởng, ta sẽ thực hiện chiếu chuỗi thời gian lên hệ cơ sở $E = \{\cos(\frac{2\pi}{P}nx)|n \in \mathbb{N}\}$. Bằng cách này, ta có thể trích ra đặc trưng của chuỗi thời gian mà không bị chịu ảnh hưởng nhiều bởi sự chênh lệch về time step như phương pháp sử dụng Euclidean metric thông thường.

Về mặt tính toán, thuật toán sẽ dựa trên giá trị của n time step trong chuỗi thời gian để tính toán hình chiếu của chuỗi thời gian lên n chiều đầu tiên của cơ sở E. Do cách tính toán này, ta không thể khắc phục được sự chênh lệch số time step 1 cách hoàn toàn.

Cụ thể thì khi ta sử dụng thư viện "scipy"để thực hiện fast fourier transform, thư viện sẽ đưa các chuỗi thời gian về 4000 time step bằng cách bổ sung các giá trị 0 vào các chuỗi cho đến khi đủ. Sau đó ta thực hiện flatten bộ dữ liệu để thu được dataset gồm các điểm dữ liệu 12000 chiều.

Như vừa phân tích, ta có thể thấy điểm yếu lớn nhất của việc sử dụng khai triển Fourier là không xử lí tốt các chuỗi thời gian quá ngắn. Cụ thể thì sau khi thực hiện đưa về frequency domain, các chuỗi thời gian quá ngắn sẽ có xu hướng trông giống nhau vì phần lớn giá trị đoạn sau sẽ bằng 0 như nhau. Tuy nhiên, việc sử dụng Fourier giúp nhận diện được nhiều đặc trưng của các label mà metric khác không thể làm được, cho ra kết quả tốt nhất trong các phương pháp xử lí dữ liệu.

2.2 Xử lý mất cân bằng dữ liệu

.

1. **Gom nhóm dữ liệu tương đồng**
   Để giải quyết vấn đề về mất cân bằng, ta cần tìm cách giảm bớt các cluster thiểu số hoặc tăng số lượng phần tử các cluster thiểu số. Trong quá trình tìm hiểu, nhóm nhận thấy ở lớp thiểu số eat_meat (3 điểm dữ liệu) và eat_soup (5 điểm dữ liệu) đây là 2 hành động gần như tương tự nhau về hình dáng trong đồ thị time series và Frequency domains. Để tránh việc gặp phải cluster có mẫu quá ít cũng như tránh việc 2 cluster này sẽ "chồng lên nhau", nhóm quyết định gộp 2 hành động này thành 1 label chung có tên là eat. Tuy nhiên điều này chưa có tác động quá đáng kể tới việc mất cân bằng dữ liệu.

2. **Oversample**
   Do số lượng dữ liệu thấp kèm với sự mất cân bằng, việc oversample dữ liệu là cần thiết. Ở đây, ta sẽ đề cập hai phương pháp phổ biến để oversampling kiểu dữ liệu này là Random oversampling và SMOTE.

   - Random oversampling:

     – Đây là phương pháp làm giàu dữ liệu bằng cách lặp lại ngẫu nhiên các điểm dữ liệu đã có sẵn nhiều lần.
     – Ưu điểm: thuật toán đơn giản, không yêu cầu số lượng data gốc lớn.
     – Nhược điểm: tạo ra các điểm dữ liệu có trọng số quá lớn, dễ dẫn đến overfit.
   - SMOTE:

     – SMOTE làm giàu dữ liệu bằng cách tạo ra các synthetic data point. Các data point này sẽ được tạo từ các vector lấy được thông qua việc tính k-neighbor.
     – Parameter đáng quan tâm khi sử dụng SMOTE là k-neighbor
     – Ưu điểm: Khắc phục được phần nào nhược điểm của random sampling.
     – Nhược điểm: Yêu cầu lượng dữ liệu ban đầu cao hơn chỉ số k-neighbor.

   Dựa vào kết quả chạy thử cũng như phân tích về cơ chế của 2 thuật toán, nhóm quyết định áp dụng SMOTE là phương pháp Oversampling data.

Mô tả SMOTE

- Như đã phân tích, random oversampling có một điểm yếu rất lớn là tạo ra các vị trí điểm dữ liệu có trọng số rất lớn. Hiểu đơn giản thì sẽ có một lượng lớn các điểm dữ liệu tụ lại tại 1 vị trí. Việc này dẫn đến hiện tượng overfit đối với các cluster thiểu số. Trên thực tế, model sau khi sử dụng bộ dữ liệu đã được random sampling có kết quả tương đối tệ, các cluster thiểu số thường bị nhận diện như 1 điểm và gộp chung với các cluster lớn.
- Ở chiều ngược lại, SMOTE dường như khắc phục được phần lớn điểm yếu của random sampling. Về cơ bản, SMOTE sẽ tạo ra các vector k-neighbor (vector nối một điểm đến k điểm gần nhất cùng label) cho các điểm ở các nhóm label thiểu số, sau đó tạo ngẫu nhiên điểm dữ liệu trên các vector đó. Việc này giúp chúng ta tránh được việc đè các điểm dữ liệu lên nhau quá nhiều lần, góp phần giúp ta tránh được hiện tượng overfit tương đối nặng như ở random sampling.
- Tuy nhiên, SMOTE cũng có 1 số điểm yếu nhất định. Bên cạnh việc yêu cầu số data point của 1 nhóm label thiểu số luôn phải lớn hơn chỉ số k-neighbor, thuật toán đôi khi hoạt động không tốt trong quá trình oversampling các nhóm dữ liệu có cấu trúc đặc biệt.

Tại sao cần Oversampling data trước khi thực hiện giảm chiều dữ liệu và rủi ro của việc Oversampling

- Trước tiên ta cần quay lại khái niệm của Oversapling data. Đây là quá trình làm giàu dữ liệu bằng cách thêm các điểm có thông tin tương tự các điểm dữ liệu gốc vào bộ dữ liệu.
- Giả sử ta thực hiện làm giàu dữ liệu sau khi giảm số chiều hay trực quan hóa chúng, ta sẽ không thể chứng minh được nguồn dữ liệu làm giàu được thêm vào nếu được ánh xạ ngược về không gian dữ liệu gốc sẽ có các đặc tính liên quan hoặc tương tự với dữ liệu gốc. Việc này làm mất đi ý nghĩa của kết quả nghiên cứu.
- Mặt khác, việc ta làm giàu dữ liệu trước trực quan sẽ đảm bảo dữ liệu thêm vào có các đặc tính liên quan dữ liệu gốc. Điều này sẽ hợp lí hơn với mục tiêu ban đầu của quá trình làm giàu dữ liệu.
- Một điều cần chú ý là chúng ta thực hiện làm giàu dữ liệu sau khi đã xử lí fourier transform, việc này cực kì rủi ro khi có thể dẫn đến hiện tượng overfit. Ta có thể chứng minh việc các dữ liệu chuỗi thời gian có đặc tính tương tự nhau thì sẽ khoảng cách gần nhau trong frequency domain nhưng việc chứng minh điều ngược lại thì không. Nói cách khác, kết quả làm việc với dữ liệu được oversampling chỉ mang tính chất tương đối.

3 Các phương pháp giảm số chiều dữ liệu

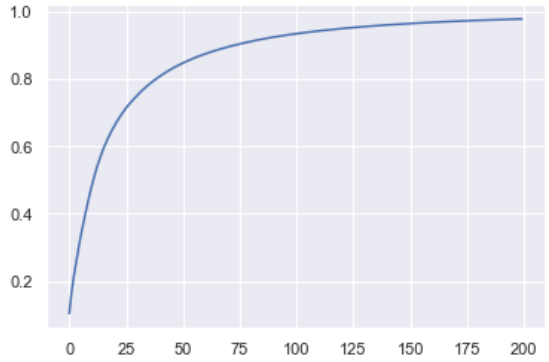
Sau quá trình xử lí ban đầu, ta thu được bộ dữ liệu 12000 chiều (4000 time step $\times$ 3 axis) gồm khoảng hơn 1000 điểm dữ liệu. Việc xử lí trên bộ dữ liệu có số chiều lớn sẽ gây khó khăn lớn trong tính toán và trực quan.

3.1 PCA

(a) **Khái niệm**
PCA (Principle Component Analysis) là một thuật toán giảm chiều dữ liệu thông qua phép biến đổi trực giao (Orthogonal Transformation). Thông qua đó, ta sẽ lấy ra được các features có nghĩa nhất đối với bộ dữ liệu. Nói một cách dễ hiểu, mục tiêu của thuật toán là tìm một cơ sở ít chiều hơn để biểu diễn dữ liệu nhưng vẫn đảm bảo lượng thông tin lớn nhất có thể.

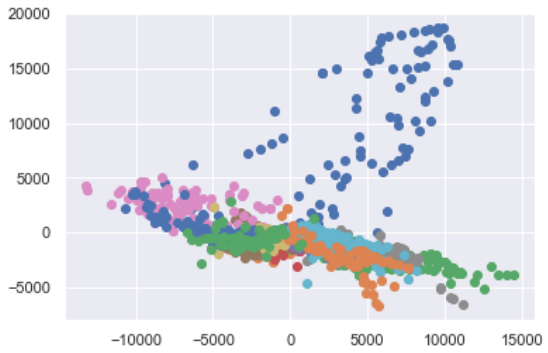
(b) **Sử dụng PCA và tham số cần quan tâm**

- Tham số đáng quan tâm nhất trong việc sử dụng PCA là số chiều của kết quả. Trong thực tế, việc chọn số chiều sao cho vẫn đảm bảo lượng thông tin đôi khi không mang lại quá nhiều lợi ích trong việc đơn giản hóa dữ liệu. Nói một cách đơn giản, đôi khi ta không thể đáp ứng đồng thời việc đơn giản hóa hay trực quan hóa dữ liệu và vẫn đảm bảo lượng thông tin cùng lúc. Cụ thể thì với bộ dữ liệu 12000 chiều (4000 time step $\times$ 3 trục), việc giảm số chiều mà vẫn đảm bảo lượng dữ liệu sẽ yêu cầu đến hơn 100 chiều.



Hình 1: Explained Variance cumulative sum

- Mặt khác, bỏ qua yêu cầu đảm bảo dữ liệu, PCA mắc tương đối nhiều vấn đề trong quá trình classify. Cụ thể thì các cluster trong thực tế sau khi thực hiện PCA sẽ "chồng lên nhau". Hiện tượng này xảy ra do PCA chỉ tập trung feature có nhiều thông tin nhất thay vì feature có lợi cho classify nhất. Nói cách khác, các đặc điểm dùng để phân biệt các lớp đôi khi sẽ bị xem là không quan trọng và bị loại bỏ.



Hình 2: Plot True Label PCA

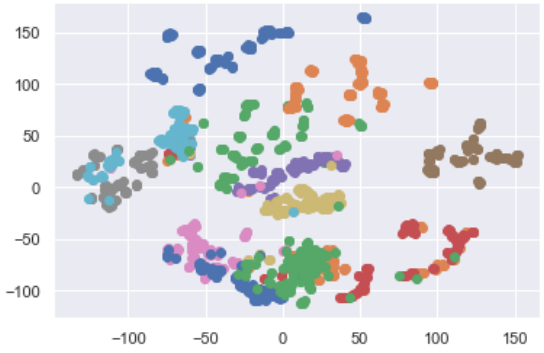
- Từ kết quả plot, ta có thể nhận thấy các điểm cùng cluster sẽ có xu hướng gần nhau. Mặc dù vậy, việc clustering sẽ tương đối khó khăn do các cluster chồng lên nhau khá nhiều.

3.2 t-SNE

(a) **Khái niệm**
t-SNE (t-distributed stochastic neighbor embedding) một thuật toán giảm chiều dữ liệu bằng cách đặt cho mỗi điểm trong bộ dữ liệu một vị trí trong bản đồ 2D hoặc 3D. Mục tiêu chính của t-SNE là bảo toàn tính gần xa của các điểm dữ liệu, qua đó ta có thể sử dụng các thuật toán clustering cơ bản để phân nhóm bộ dữ liệu.

(b) **Sử dụng t-SNE và tham số cần quan tâm**

- Một số tham số cần lưu ý trong t-SNE là số bước thực hiện và perplexity. Trong đó, perplexity là tham số giúp cân bằng giữa việc bảo toàn cấu trúc toàn cục và bảo toàn cấu trúc liên thông của bộ dữ liệu. Ta có thể tham khảo trong blog https://distill.pub/2016/misread-tsne/



Hình 3: Plot True predict after t-SNE

- Ta có thể nhận thấy các điểm dữ liệu thuộc cùng 1 cluster sau khi xử lí và biểu diễn trên t-SNE plot vẫn có xu hướng ở gần nhau. Điều này chứng tỏ phương pháp Fourier transform thực hiện tương đối tốt việc trích xuất đặc trưng của các điểm dữ liệu.

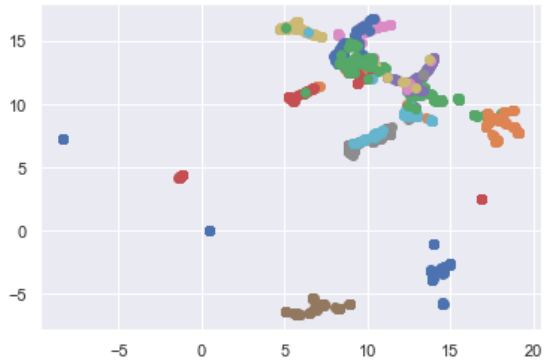
- Tuy nhiên các cluster vẫn chồng lên nhau ở một số vùng. Một trong những nguyên nhân việc này có thể là việc các chuỗi thời gian ngắn sẽ có xu hướng "gần nhau"vì có cấu trúc tương đối giống nhau (trình bày rõ trong phần cuối mục 2.1) .

3.3 UMAP

(a) **Khái niệm**
Uniform Manifold Approximation and Projection (UMAP) là một thuật toán giảm chiều dữ liệu dưa trên manifold learning techniques và topo phân tích dữ liệu. Với cách xây dựng tương tự t-SNE nhưng thay vì định nghĩa lân cận theo khoảng cách và phân phối, UMAP sẽ định nghĩa lân cận theo khái niệm quả cầu mở. Việc này đảm bảo cho UMAP một cách tính toán đơn giản song vẫn đảm bảo tính chặt chẽ của định nghĩa lân cận. Đây là lí do UMAP có thời gian tính toán tương đối thấp so với t-SNE mà vẫn đạt hiệu quả tốt.

(b) **Sử dụng UMAP và tham số cần quan tâm**

- Hai tham số cần quan tâm khi sử dụng UMAP là min_dist và n-neighbors. Đây chính là hai tham số dùng để định nghĩa quả cầu mở trong UMAP, tác động trực tiếp đến định nghĩa lân cận giữa các điểm dữ liệu. Việc điều chỉnh 2 tham số này sẽ tương tự với điều chỉnh tham số perplexity trong t-SNE. Ta có thể tham khảo qua đường dẫn sau: https://pair-code.github.io/understanding-umap/



Hình 4: Plot True predict after UMAP

- Tương tự như kết quả t-SNE, các điểm dữ liệu cùng cluster thật được biểu diễn trên UMAP plot cũng có xu hướng gần nhau. Mặt khác, các cluster "chồng lên nhau"ở một số vùng. Từ kết quả có được, ta cũng đưa ra các nhận định tương tự với t-SNE.

4 Đánh giá các phương pháp giảm số chiều thông qua kết quả thuật toán K-means

Để kiểm tra các nhận định ở mục trên, ta sử dụng K-means để clustering dữ liệu rồi sử dụng kết quả đó so sánh với true_label.

4.1 K-means

K-means là thuật toán unsupervised learning dùng để nhóm các dữ liệu theo lân cận. Thuật toán sẽ gồm các bước.

1. Tạo ngẫu nhiên 13 centroid ứng với 13 nhóm hành động sau khi xử lí dữ liệu.

2. Gán label các điểm dữ liệu dựa trên khoảng cách của chúng đến các centroid.

3. Cập nhật lại centroid của các cluster bằng cách tính trung bình giá trị của các điểm dữ liệu thuộc cluster.

4. Lặp lại 2 và bước 3 trên cho đến khi các centroid vừa cập nhật không có sự thay đổi hoặc label các điểm dữ liệu không thay đổi.

4.2 So sánh kết quả K-means sau khi sử dụng các phương pháp giảm số chiều

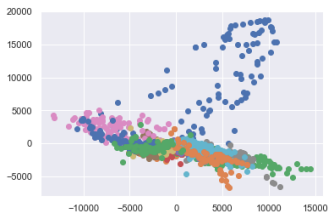
Khi clustering bằng K-means, label của từng cluster sẽ được đánh ngẫu nhiên nên ta phải dùng một chỉ số để đánh giá độ hiệu quả của các thuật toán đã dùng. Hai chỉ số được sử dụng để đánh giá đó là

- homogeneity_score: kết quả có homogeneity_score cao nếu mỗi cluster hầu như chỉ chứa các điểm thuộc 1 class.
- completeness_score: kết quả có completeness_score cao nếu hầu như tất cả các điểm của một class thuộc về củng một cluster.

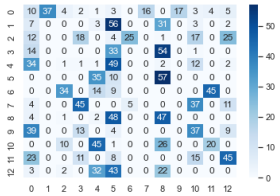
Nói một cách dễ hiểu, homogeneity_ score biểu diễn cho độ thuần khiết của các cluster dự đoán còn completeness_ score biểu diễn cho độ bao quát của các cluster dự đoán.

Ngoài ra, nhóm sử dụng công cụ là confusion matrix để tính toán cách chọn label thực tối ưu. Confusion matrix là một cách tương đối trực quan để quan sát kết quả clustering.

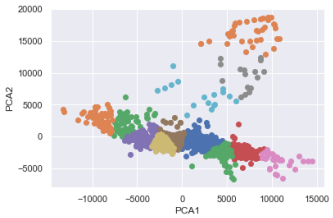
1. **PCA**



(a) True Cluster Plot



(b) PCA



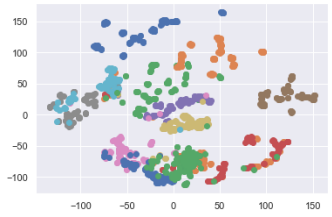
(c) Predict Cluster Plot

- homogeneity_score: 35.98
- completeness_score: 57.23

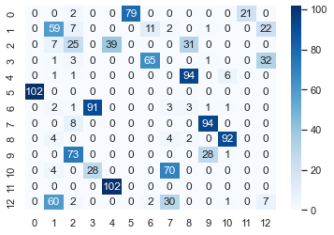
Ta có thể thấy kết quả của PCA tương đối thấp. Có thể thấy trong hình các cluster không thực sự tách biệt nhau. Đây cũng chính là lí do tại sao homogeneity_score tương đối thấp, K-means chỉ đơn giản chia khối dữ liệu ra thành 13 cluster ngẫu nhiên chứ không thực sự phân biệt được các cluster. Việc này hoàn toàn hợp lí với các phân tích ban đầu của nhóm ở phần 3.1.

Qua đó ta có thể thấy thuật toán giảm số chiều tuyến tính không hiệu quả với bộ dữ liệu trên, đặc biệt trong việc classify.

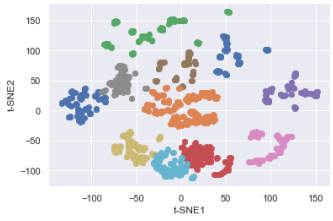
2. **t-SNE**



(a) True Cluster Plot



(b) t-SNE

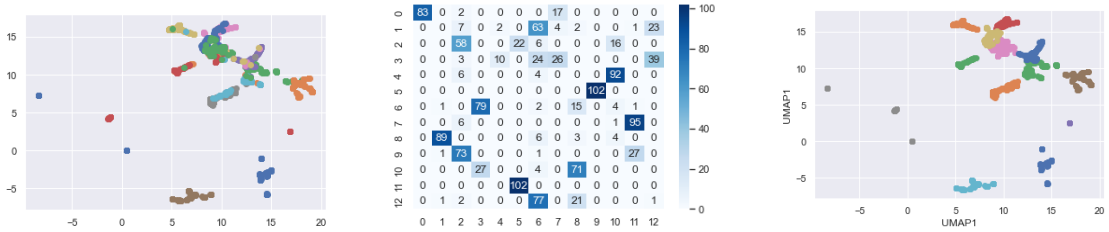


(c) Predict Cluster Plot

- homogeneity_score: 73.84
- completeness_score: 75.74

Ta có thể thấy kết quả của t-SNE tốt hơn rất nhiều so với PCA từ trực quan đến số liệu tính toán. Quan sát trực tiếp confusion matrix, ta có thể thấy các cluster K-means từ kết quả PCA thường bị lẫn lộn nhiều điểm từ nhiều label khác nhau. Điều này được cải thiện rất nhiều đối với t-SNE.
Tuy nhiên, t-SNE vẫn chưa thể tránh khỏi sự chồng lên nhau của một số cluster. Điều này khá hợp lí với những phân tích về kết quả t-SNE ở phần trước.

3. **UMAP**



(a) True Cluster Plot (b) UMAP (c) Predict Cluster Plot

- homogeneity_score: 71.09
- completeness_score: 74.16

Tương tự t-SNE, kết quả K-means của plot UMAP cho ra kết quả khá tốt. Từ kết quả của t-SNE và UMAP, ta có thể nhận thấy nguyên nhân dẫn đến kết quả không quá tốt đến từ bước xử lí về số chiều dữ liệu. Mặc dù xử lí tương đối tốt so với các phương pháp đã thử, FFT (fast fourier transform) vẫn gặp hạn chế trong việc phân biệt một số điểm dữ liệu.

5 Tổng kết

- Nhận xét về bài toán, dataset tương đối thực tế khi chứa nhiều khó khăn thường gặp trong quá trình xử lí dữ liệu dạng chuỗi thời gian. Cụ thể thì các time series có độ dài khác nhau, nhiều hành động tương đối giống nhau, số mẫu trong từng label không đều nhau.
- Về mặt hạn chế, như đã phân tích trong báo cáo, kết quả của nhóm chưa thực sự tốt cũng như có khả năng overfit tương đối nặng với bộ data. Mặt khác, việc data bị mất cân bằng của phần nào hạn chế điểm mạnh của các thuật toán giảm số chiều và phân nhóm dữ liệu.
- Về định hướng tương lai, để có thể cải thiện kết quả, chúng ta có thể cải thiện bộ dữ liệu đa dạng và cân bằng hơn, tìm kiếm thêm một số metric thích hợp cho việc chỉ ra đặc trưng chuỗi thời gian.

Tổng kết bài báo cáo, nhóm đã thực hiện thử nghiệm, phân tích và định hướng giải quyết các vấn đề dựa trên kiến thức hiện có. Ngoài ra, báo cáo còn chỉ ra điểm mạnh và điểm yếu của một số thuật toán xử lí dữ liệu cũng như giảm số chiều dữ liệu.

Tài liệu

[1] Human Activity Recognition on wrist-worn accelerometers using self-supervised neural networks, Niranjan Sridhar, Lance Myers

[2] TimeCluster: dimension reduction applied to temporal data for visual analytics, Mohammed Ali, Mark W. Jones, Xianghua Xie & Mark Williams

[3] Understanding UMAP, Andy Coenen, Adam Pearce | Google PAIR

[4] How to Use t-SNE Effectively

[5] Visualizing Data using t-SNE, Maaten, L.v.d. and Hinton, G., 2008. Journal of Machine Learning Research

[6] SMOTE: Synthetic Minority Over-sampling Technique, Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall & W. Philip Kegelmeyer

[7] Time Series, Signals, & the Fourier Transform, Shawhin Talebi

[8] Scikit-learn: Machine Learning in Python

[9] Imbalanced Dataset with Python