ĐẠI HỌC QUỐC GIA TPHCM

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

ILAB-03: Trực quan hoá dữ liệu với t-SNE

TRỰC QUAN HOÁ DỮ LIỆU

Sinh viên thực hiện:

Tôn Anh Huy (21120257)

Giáo viên hướng dẫn:

Thầy Lê Nhựt Nam

Ngày 9 tháng 6 năm 2024

Mục lục

1 Các yêu cầu đã thực hiện

Các yêu cầu được thực hiện:

- Tìm hiểu về t-sne
- Áp dụng t-sne trên dữ liệu (file notebook)
- So sánh giữa PCA và t-SNE

2 Giới thiệu

t-SNE (t-distributed Stochastic Neighbor Embedding) là một kỹ thuật giảm chiều dữ liệu phi tuyết tính (non-linear technique) hỗ trợ cho việc khám phá dữ liệu và trực quan hoá đối với dữ liệu có nhiều chiều.

Kĩ thuật giảm chiều phi tuyến nghĩa là thuật toán này cho phép chúng ta phân tách các điểm dữ liệu mà không thể dùng một đường thẳng.

t-SNE giúp chúng ta có thể trực quan các bộ dữ liệu phức tạp bằng cách sử dụng từ 2-3 chiều, qua đó cho phép việc hiểu dữ liệu và khám phá ra các pattterns ẩn có giá trị, các mối quan hệ có trong dữ liệu

3 Lý thuyết nền tảng

3.1 Stochastic Neighbor Embedding

SNE (Stochastic Neighbor Embedding) chuyển các giá trị khoảng cách Euclidean của các diểm dữ liệu trong không gian nhiều chiều thành các giá trị xác suất có điều kiện (đại diện độ tương tự của các điểm dữ liệu).

Độ tương tự của điểm dữ liệu x_j so với x_i là một xác suất có điều kiện $p_{j|i}$. Xác suất này đại diện cho khả năng mà điểm x_i chấp nhận x_j là láng giềng của nó. Giá trị xác suất này được tính bằng hàm phân phối xác suất (pdf) của phân phối Gaussian có tâm là x_i.

Các điểm dữ liệu càng gần x_i thì sẽ có giá trị xác suất cao, ngược lại các điểm dữ liệu ở càng xa thì xác suất càng thấp. Công thức để tính $p_{j|i}$ như sau:

$$p_{i|j} = \frac{exp(-\|x_i - x_j\|^2/2\sigma_i^2)}{\sum_{k \neq i} exp(-\|x_i - x_k\|^2/2\sigma_i^2)} \tag{1}$$

Trong đó, giá trị σ_i là phương sai của phân phối Gaussian có tâm tại x_i. Đối với không gian có số chiều thấp, y_i và y_j là 2 điểm dữ liệu tương ứng với x_i và x_j ở không gian có số chiều cao. Ta có thể tính được xác suất $q_{j|i}$ (hay độ tương tự) của y_i và y_j.

Phân phối xác suất có điều kiện của y_j đối với y_i được tính như sau:

$$q_{i|j} = \frac{exp(-\|y_i - y_j\|^2)}{\sum_{k \neq i} exp(-\|y_i - y_k\|^2)} \tag{2}$$

Trong đó, phương sai có giá trị là $\frac{1}{\sqrt{2}}$

Nếu 2 điểm trong không gian số chiều thấp là y_i và y_j giữ được độ tương tự giữa x_i và x_j trong không gian có số chiều cao ban đầu thì các giá trị xác suất có điều kiện của chúng nên bằng nhau:

$$p_{j|i} = q_{j|i} \tag{3}$$

Để đo lường được khả năng $q_{j|i}$ có thể mô hình hoá được $p_{j|i}$, người ta sử dụng một thước đo có tên Kullback-Leibler divergence (đo lường sự khác biệt giữa 2 phân phối). SNE sẽ cố gắng tối tiểu hoá giá trị này trên tất cả các điểm dữ liệu bằng cách sử dụng Gradient Descent. Hàm mất mát C được cho như sau:

$$C = \sum_i KL(P_i \parallel Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}} \tag{4}$$

Trong đó:

- P_i: phân phối của độ tương tự trên tất cả điểm dữ liệu x_i
- Q_i: phân phối của độ tương tự trên tất cả điểm dữ liệu y_i

Ví dụ với hàm mất mát này, nếu như các điểm dữ liệu trong không gian gốc nằm ở gần nhau (độ tương tự cao) mà trong không gian có số chiều thấp lại nằm xa nhau (độ tương tự thấp) thì giá trị hàm mất mát sẽ tăng lên, đây cũng chính là điều ta cần tối ưu.

Giá trị phương sai σ_i kiểm soát độ rộng của phân phối Gaussian, giá trị càng cao thì phân phối có động rộng càng lớn và ngược lại. Thông thường, sẽ không có giá trị σ_i tối ưu cho tất cả các điểm dữ liệu. Đối với các vùng có mật độ dữ liệu cao, σ_i có giá trị nhỏ sẽ phù hợp hơn và ngược lại.

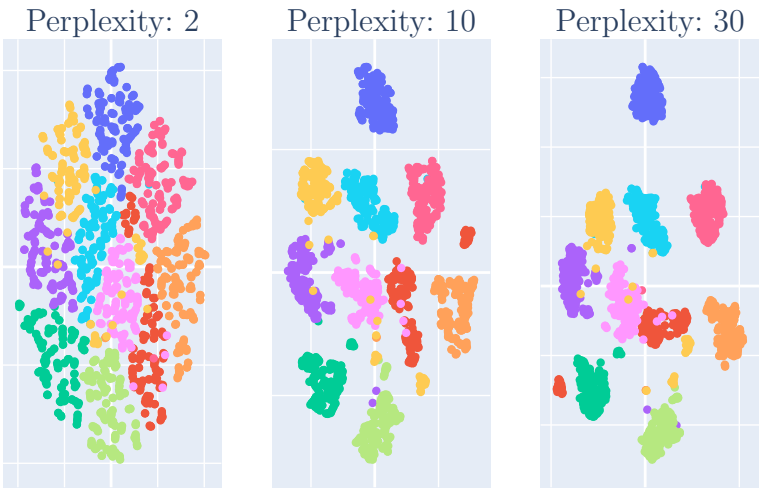
Mỗi một giá trị bất kỳ σ_i sẽ tương ứng với một phân phối P_i đối với toàn bộ dữ liệu. Phân phối này có giá trị entropy tăng khi σ_i tăng. Thuật toán SNE sẽ thực hiện tìm kiếm nhị phân trên giá trị σ_i dựa trên một siêu tham số được chỉ định bởi người dùng, được gọi là **perplexity**.

Perplexity có thể được xem như số điểm láng giềng của một điểm dữ liệu cụ thể. Đây là siêu tham số được xem là quan trọng nhất. Giá trị thông thường là từ 5 - 10, ta cần phải thực hiện trên nhiều giá trị để xem tác động của nó tới kết quả.

$$Perp(P_i) = 2^{H(P_i)} \tag{5}$$

Trong đó, $H(P_i)$ là giá trị Shannon entropy của phân phối P_i.

$$H(P_i) = -\sum_j p_{j|i} log_2 p_{j|i} \tag{6}$$



Hình 1: So sánh các giá trị perplexity khác nhau, load_digits dataset

Để có thể tối tiểu hoá giá trị của hàm mất mát C, người ta sử dụng Gradient Descent và giá trị gradient có dạng như sau:

$$\frac{\partial C}{\partial y_i} = 2\sum_j (p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j})(y_i - y_j) \tag{7}$$

Quá trình Gradient Descent được khởi tạo bằng cách lấy mẫu ngẫu nhiên cho các điểm dữ liệu trong không gian thấp chiều từ phân phối Gaussian có giá trị phương sai nhỏ. Đảm bảo cho các điểm dữ liệu sẽ được phân tán đều xung quanh tâm.

Để tăng tốc cho quá trình tối ưu và tránh gặp phải các điểm cực tiểu cục bộ, một giá trị đại diện cho quán tính được thêm vào. Điều này giúp cho việc tối ưu được tiếp tục theo hướng của lần lặp trước đó, đặc biệt khi giá trị gradient chuyển hướng đột ngột hay gặp phải các cực tiểu cục bộ. Giá trị Gradient được tính như sau:

$$\mathcal{Y}^t = \mathcal{Y}^{t-1} + \eta\frac{\partial C}{\partial \mathcal{Y}} + \alpha(t)(\mathcal{Y}^{t-1} - \mathcal{Y}^{t-2}) \tag{8}$$

Trong đó:

- $\mathcal{Y}$: toạ độ của các điểm dữ liệu trong không gian thấp chiều (low-dimensional space) ở lần lặp thứ t
- η: hệ số học trong Gradient Descent
- $\alpha(t)$: lực quán tính tại lần lặp thứ t

Ở giai đoạn đầu, $\mathcal{Y}$ được thêm vào các giá trị nhiễu Gaussian với mục đích làm nhiễu các giá trị toạ độ, ngăn ngừa việc tối ưu bị mắc kẹt tại các điểm cực tiểu cục bộ.

Phương sai của các giá trị nhiễu Gaussian sẽ được dần dần giảm bớt, cho phép thuật toán dần trở nên ổn định và giúp hội tụ dễ dàng hơn, khám phá được cấu trúc tổng thể của dữ liệu.

Tính hiệu quả của việc thêm các giá trị nhiễu bị ảnh hưởng bởi các tham số khác trong quá trình tối ưu, như là lực quán tính, step size trong gradient descent...

3.2 t-Distributed Stochastic Neighbor Embedding

Mặc dù SNE thể hiện rất tốt trong việc trực quan hoá, tuy nhiên nó phải đối mặt với 2 vấn đề chính:

- Hàm mất mát khó tối ưu
- Đối mặt với "Crowding Problem"

Do đó, một kỹ thuật mới là **t-Distributed Stochastic Neighbor Embedding** được đề xuất để cải thiện vấn đề này thông qua 2 việc:

- Sử dụng hàm mất mát có tính đối xứng, dễ dàng tối ưu hơn
- Sử dụng phân phối Student-t thay cho Gaussian để tính toán độ tương tự của các điểm dữ liệu trong không gian số chiều thấp (low-dimensonal space).

3.2.1 Tính đối xứng của SNE

Với 2 điểm dữ liệu x_i và x_j trong không gian số chiều lớn, ta mong đợi rằng độ tương tự của chúng là như nhau, tức là $p_{j|i} = p_{i|j}$. Tuy nhiên, điều này là không đảm bảo.

Khi tồn tại một điểm ngoại lai trong không gian số chiều lớn (gọi là x_i). Độ tương tự của nó đối với các điểm khác là rất nhỏ (hay $p_{j|i}$ luôn rất nhỏ). Dẫn đến tác động của nó tới hàm mất mát là rất ít, làm cho vị trí của nó trong không gian số chiều thấp được xác định chưa đủ tốt. Tác giả đề xuất một giá trị xác suất chung là $p_{ij} = \frac{p_{j|i}+p_{i|j}}{2n}$, điều này giúp cho các điểm ngoại lai có thể đóng góp tác động vào hàm mất mát. Bên cạnh đó, nó cũng giúp cho việc tính gradient trở nên đơn giản hơn:

$$\frac{\partial C}{\partial y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j) \tag{9}$$

3.2.2 Crowding Problem

SNE cố gắng duy trì xác suất tương đồng giữa các điểm trong không gian gốc và không gian mới, nhưng số điểm có thể nằm trong một vùng nhỏ của không gian chiều thấp bị hạn chế so với không

gian chiều cao. Điều này dẫn đến việc các điểm dữ liệu ở xa trong không gian chiều cao có thể bị nén lại gần nhau trong không gian chiều thấp, gây ra hiện tượng chèn lấn (crowding).

Sử dụng phân phối Student thay vì phân bố Gaussian để biểu diễn sự tương đồng giữa các điểm trong không gian chiều thấp. Điều này giúp giảm thiểu vấn đề crowding bằng cách cho phép các điểm dữ liệu có khả năng nằm xa nhau hơn trong không gian chiều thấp. Các lợi ích của việc dùng phân phối Student

- Chi phí tính toán giảm bớt do không cần tính giá trị exp
- Không cần thực hiện việc tìm kiếm nhị phân cho σ

3.3 Thuật toán

Algorithm 1 t-SNE Algorithm

Input: dataset $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$, cost function parameters: perplexity $Perp$, optimization parameters: number of iterations T, learning rate η, momentum $\alpha(t)$

Output: low-dimensional data representation $\mathcal{Y}^{(T)} = \{y_1, y_2, \ldots, y_n\}$

Compute pairwise affinities $p_{j|i}$ with perplexity $Perp$
Set $p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$
Sample initial solution $\mathcal{Y}^{(0)} = \{y_1, y_2, \ldots, y_n\}$ from $\mathcal{N}(0, 10^{-4}I)$
for $t = 1$ to T **do**
    Compute low-dimensional affinities q_{ij}
    Compute gradient $\frac{\delta C}{\delta \mathcal{Y}}$
    Set $\mathcal{Y}^{(t)} = \mathcal{Y}^{(t-1)} + \eta \frac{\delta C}{\delta \mathcal{Y}} + \alpha(t)(\mathcal{Y}^{(t-1)} - \mathcal{Y}^{(t-2)})$
end for=0

4 Các phương pháp tối ưu hoá cho t-SNE

4.1 Tối ưu hoá hàm mất mát

Phương pháp này thêm vào một giá trị đại diện cho lực quán tính, giá trị này sẽ giảm dần khi không gian số chiều thấp dần trở nên phù hợp. Thêm vào đó, có thể sử dụng adaptive learning rate (hệ số học thích ứng) tăng tốc quá trình tối ưu để tăng giá trị của hệ số học theo hướng mà gradient có xu hướng ổn định.

4.2 Early compression

Khoảng cách giữa các điểm dữ liệu trong không gian số chiều thấp sẽ có xu hướng nằm ở cạnh nhau khi bắt đầu quá trình tối ưu. Điều này giúp cho thuật toán khai thác các mối quan hệ và đặc tính của các cluster dễ dàng hơn.

4.3 Early exeggeration

Đây là kĩ thuật được áp dụng trong vài lần lặp đầu tiên của quá trình tối ưu. Kỹ thuật này sẽ làm gia tăng khoảng cách giữa các điểm dữ liệu trong không gian có số chiều thấp với mục đích làm cho dữ liệu trở nên phân tán hơn. Điều này giúp thuật toán dễ dàng trong việc khám phá ra đặc tính và duy truỳ cấu trúc vốn có của không gian ban đầu.

5 Ứng dụng thực tế

t-SNE được ứng dụng chủ yếu trong 2 tác vụ chính là:

- Trực quan hoá dữ liệu
- Phân tích dữ liệu

Một vài ứng dụng của t-sne trong thực tế:

- Sinh học và y học: phân tích dữ liệu tế bào đơn: sử dụng để trực quan sự biểu hiện gene của từng tế bào, nhận diện các loại tế bào và trạng thái.
- An ninh mạng: giúp phát hiện các mẫu dữ liệu bất thường trong các hệ thống quan sát lưu lượng mạng.
- Xã hội học và nhân văn: trực quan các mẫu và xu hướng trong dữ liệu khảo sát phức tạp, nghiên cứu mạng xã hội thông qua trực quan các cấu trúc.
- Xử lý ngôn ngữ tự nhiên: sử dụng t-SNE để trực quan các vectors có dạng word2vec, ta có thể quan sát được các từ ngữ có mối quan hệ gần nhau
- Thương mại: trực quan hoá dữ liệu đánh giá, phản hồi và thông tin, hành vi để tìm ra các nhóm khách hàng tương đồng

6 So sánh với PCA

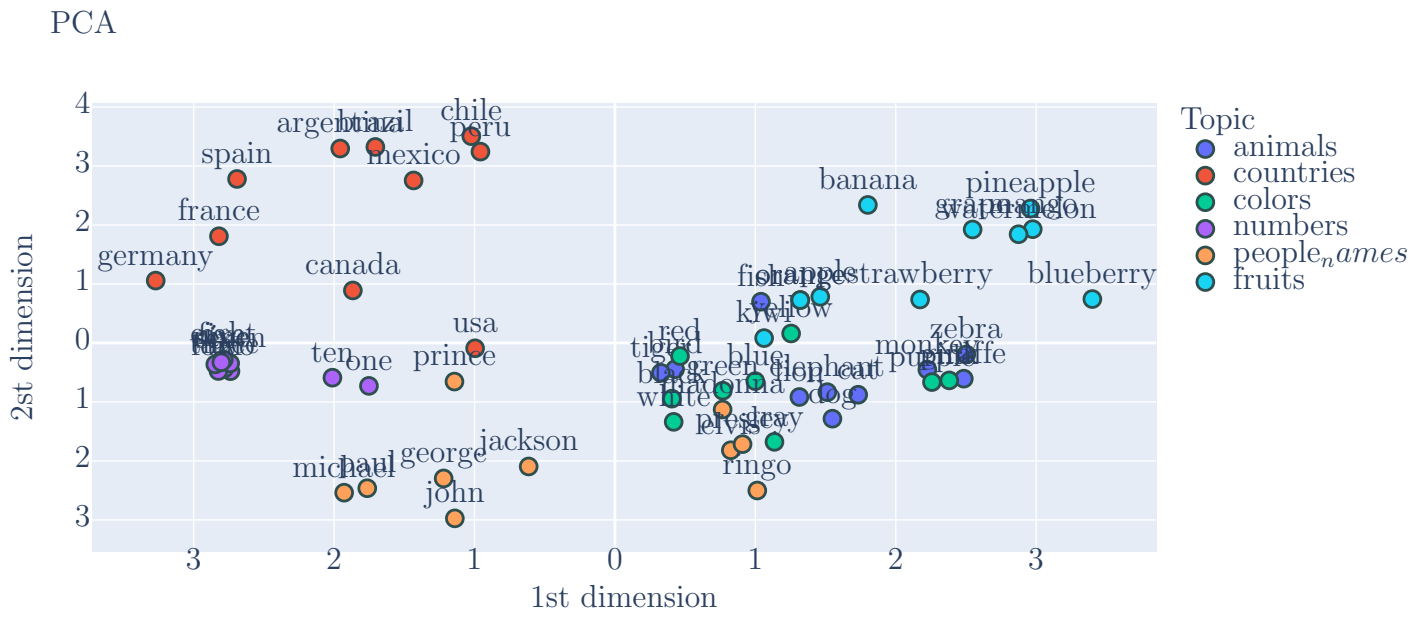
Mặc dù đã có kỹ thuật giảm chiều khá hiệu quả là PCA, tuy nhiên nó gặp phải nhiều hạn chế khi xử lý dữ liệu phi tuyến. Trong khi t-SNE đều có thể xử lý tốt cả dữ liệu tuyến tính và phi tuyến.

Tuy nhiên, t-SNE vướng phải nhiều điểm yếu như:

- Khó khăn trong việc diễn giải kết quả: kết quả của t-SNE không dễ dàng diễn giải như PCA vì không có sự bảo toàn phương sai hay các thành phần chính rõ ràng. Cần phải có kinh nghiệm và đủ hiểu biết về t-SNE mới có thể đưa ra các nhận xét đúng đắn.
- Chi phí tính toán cao: đặc biệt đối với các bộ dữ liệu quá lớn, t-SNE gặp phải nhiều khó khăn do phải thực hiện nhiều lần và tính ngẫu nhiên của thuật toán.

6.1 Ví dụ

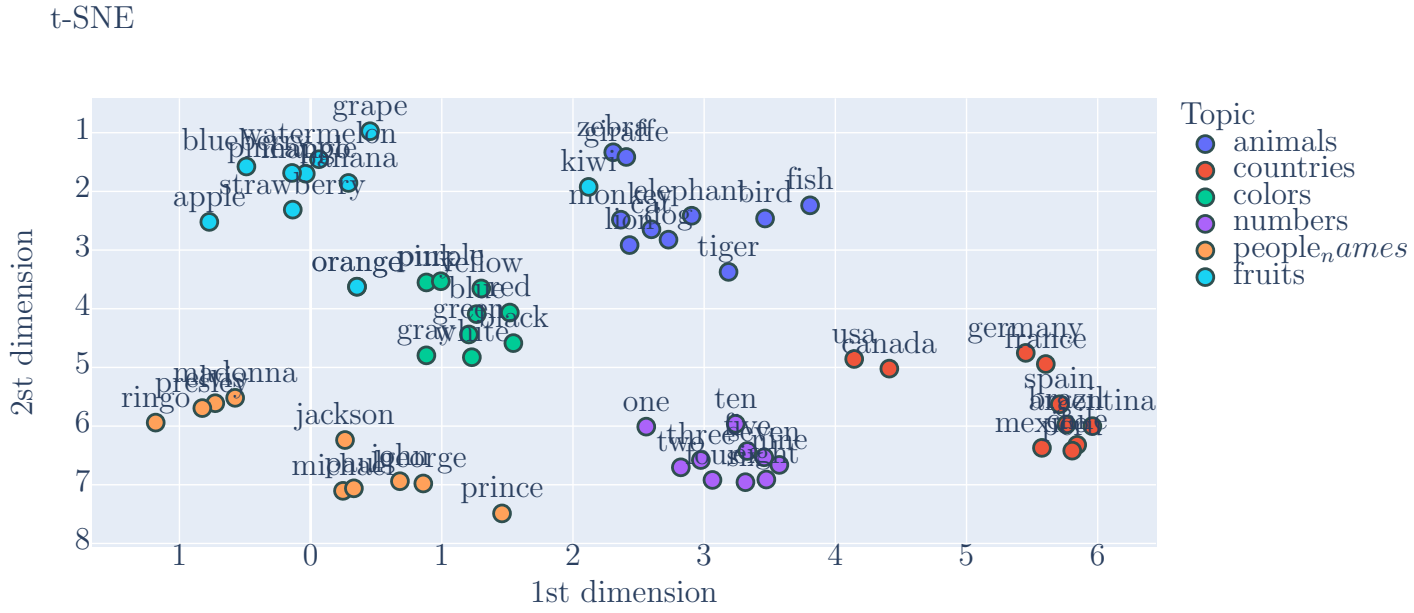
Xét bộ dữ liệu có chứa các vector word embedding gồm 50 chiều đại diện cho các từ



Hình 2: Trực quan dữ liệu bằng PCA

Nhận xét

- Dễ dàng thấy giữa các clusters có sự trùng lắp với nhau
- Chưa thể hiện mối quan hệ giữa các từ một cách rõ ràng
- PCA thể hiện tốt trong việc bảo toàn cấu trúc tổng thể của dữ liệu, giữ lại được nhiều thông tin nhất có thể. Tuy nhiên không thể hiện rõ nét mối quan hệ gi các cụm dữ liệu.



Hình 3: Trực quan dữ liệu bằng t-SNE

Nhận xét

- Các clusters được phân tách rõ ràng, tình trạng overlap giữa các cluster đã được giảm đi đáng kể.
- Có thể thấy rằng, các từ thuộc về một chủ đề nhất định thường nằm trong cùng một cluster. Bởi vì mục tiêu của t-sne là cố gắng duy trì độ tương tự của các điểm dữ liệu trong không gian mới so với ban đầu.
- Ví dụ: các từ thuộc chủ đề "chữ số" như : one, two, three... có xu hướng nằm gần nhau. Tương tự, các từ có liên quan tới chủ đề "đất nước" nằm rất gần nhau trong không gian mới.

6.2 Bảng tóm tắt sơ lược

STT.	PCA	t-SNE
1.	Kỹ thuật giảm chiều tuyến tính.	Kỹ thuật giảm chiều phi tuyến
2.	Cố gắng giữ lại nhiều thông tin nhất có thể	Cố gắng duy trì cấu trúc cục bộ của dữ liệu.
3.	Khả năng trực quan không hiệu quả bằng t-SNE.	Trực quan hoá được các bộ dữ liệu phức tạp.
4.	Triển khai đơn giản do không có siêu tham số.	Chỉ định siêu tham số như: perplexity, learning rate và số lần lặp.
5.	Ảnh hưởng bởi outlier.	Có cơ chế xử lý outlier.
6.	Có tính xác định.	Không có tính xác định (ngẫu nhiên)
7.	Có thể quyết định lượng phương sai cần giữ lại.	Không thể duy trì phương sai nhưng duy trì được khoảng cách.
8.	Chi phí tính toán thấp hơn, đặc biệt với datasets lớn. EVD: $O(n^3)$,SVD: $O(mn^2)$	t-SNE chi phí tính toán cao, đặc biệt với datasets lớn và có nhiều chiều. $O(n^2)$, Barnes Hut: $O(nlog(n))$
9.	Có thể sử dụng cho việc trích xuất đặc trưng.	Chủ yếu dùng cho trực quan và khám phá dữ liệu.

Bảng 1: So sánh giữa PCA và t-SNE

Tài liệu

[1] Visualizing Data using t-SNE (2008) Laurens van der Maaten, Geoffrey Hinton