DR_tSNE

May 22, 2024

0.0.1 Ý tưởng chính và bài toán t-SNE

1. **t-SNE là gì và tại sao nó được sử dụng?**
2. **Những bài toán nào thường được giải quyết bằng t-SNE?**
3. **t-SNE có thể giúp giảm chiều dữ liệu như thế nào?**
4. **Ý tưởng chính của t-SNE là gì?**
5. **Tại sao t-SNE được coi là một kỹ thuật giảm chiều phi tuyến tính?**
6. **Trong t-SNE, "t-distributed Stochastic Neighbor Embedding" có nghĩa là gì?**
7. **Làm thế nào t-SNE có thể giúp trong việc trực quan hóa dữ liệu?**

0.0.2 Input và Output của t-SNE

8. **Dữ liệu đầu vào cho t-SNE yêu cầu những gì?**
9. **Dữ liệu đầu vào có cần phải chuẩn hóa không? Tại sao?**
10. **t-SNE tạo ra những output gì từ dữ liệu đầu vào?**
11. **Làm thế nào để chọn số lượng dimensions cần giữ lại trong t-SNE?**
12. **Tại sao t-SNE thường được sử dụng để giảm chiều dữ liệu xuống 2 hoặc 3 dimensions?**
13. **Perplexity là gì và nó ảnh hưởng như thế nào đến kết quả của t-SNE?**

0.0.3 Thuật toán triển khai t-SNE

14. **Các bước chính của thuật toán t-SNE là gì?**
15. **Tại sao chúng ta cần tính toán các xác suất có điều kiện giữa các điểm trong không gian gốc?**
16. **Làm thế nào để tính toán các xác suất có điều kiện giữa các điểm trong không gian giảm chiều?**
17. **Làm thế nào để tối thiểu hóa hàm mất mát Kullback-Leibler divergence trong t-SNE?**
18. **Tại sao t-SNE sử dụng phân phối t với một bậc tự do (Student's t-distribution)?**
19. **Làm thế nào để chọn giá trị cho các siêu tham số như perplexity và learning rate?**
20. **Sự khác biệt giữa t-SNE và PCA là gì?**

0.0.4 Ưu điểm và nhược điểm của t-SNE

21. **Những ưu điểm chính của t-SNE là gì?**
22. **t-SNE có những nhược điểm gì?**
23. **t-SNE có thể gặp vấn đề gì khi dữ liệu đầu vào không được chuẩn hóa?**
24. **Tại sao t-SNE không thể làm việc tốt với dữ liệu rất lớn?**
25. **t-SNE có thể làm giảm thông tin quan trọng nào trong dữ liệu?**

26. **Ưu điểm của việc sử dụng t-SNE trong việc trực quan hóa dữ liệu là gì?**
27. **Tại sao kết quả của t-SNE có thể không ổn định khi chạy lại nhiều lần?**
28. **t-SNE có thể được cải thiện bằng cách sử dụng các kỹ thuật nào khác?**

0.0.5 Ứng dụng và ví dụ của t-SNE

29. **Một số ứng dụng thực tế của t-SNE trong phân tích dữ liệu là gì?**
30. **Làm thế nào t-SNE có thể được sử dụng trong lĩnh vực nhận dạng mẫu?**
31. **t-SNE có thể giúp gì trong việc phát hiện và xử lý dữ liệu ngoại lai?**
32. **Làm thế nào t-SNE có thể giúp trong việc trực quan hóa dữ liệu?**
33. **Một ví dụ cụ thể về việc sử dụng t-SNE trong xử lý ảnh là gì?**

0.0.6 Đánh giá output thu được từ t-SNE

34. **Làm thế nào để đánh giá chất lượng của output thu được từ t-SNE?**

0.0.7 Ý tưởng chính và bài toán t-SNE

1. **t-SNE là gì và tại sao nó được sử dụng?** t-SNE (t-Distributed Stochastic Neighbor Embedding) là một kỹ thuật giảm chiều dữ liệu phi tuyến tính được sử dụng để trực quan hóa dữ liệu có chiều cao trong không gian hai hoặc ba chiều. Nó được sử dụng để phát hiện cấu trúc dữ liệu tiềm ẩn và cụm (clusters) trong dữ liệu một cách trực quan.

2. **Những bài toán nào thường được giải quyết bằng t-SNE?** t-SNE thường được sử dụng trong các bài toán trực quan hóa dữ liệu, phát hiện cụm (clustering), nhận dạng mẫu, và phân tích dữ liệu đa chiều trong lĩnh vực học máy và khoa học dữ liệu.

3. **t-SNE có thể giúp giảm chiều dữ liệu như thế nào?** t-SNE giảm chiều dữ liệu bằng cách tối thiểu hóa sự khác biệt giữa các xác suất có điều kiện của các điểm dữ liệu trong không gian gốc và không gian giảm chiều. Điều này giúp giữ lại các mối quan hệ cục bộ và toàn cục giữa các điểm dữ liệu.

4. **Ý tưởng chính của t-SNE là gì?** Ý tưởng chính của t-SNE là duy trì các quan hệ cục bộ giữa các điểm dữ liệu trong không gian giảm chiều bằng cách sử dụng xác suất có điều kiện để biểu diễn các mối quan hệ này, và sau đó tối thiểu hóa hàm mất mát để giữ cho các xác suất này tương đồng trong không gian giảm chiều.

5. **Tại sao t-SNE được coi là một kỹ thuật giảm chiều phi tuyến tính?** t-SNE là một kỹ thuật giảm chiều phi tuyến tính vì nó sử dụng các hàm phi tuyến để duy trì các mối quan hệ cục bộ và toàn cục giữa các điểm dữ liệu trong quá trình chuyển đổi từ không gian cao chiều xuống không gian thấp chiều.

6. **Trong t-SNE, "t-distributed Stochastic Neighbor Embedding" có nghĩa là gì?** "t-distributed Stochastic Neighbor Embedding" đề cập đến việc sử dụng phân phối t với một bậc tự do (Student's t-distribution) để biểu diễn các xác suất có điều kiện trong không gian giảm chiều. Phân phối t giúp giảm thiểu vấn đề crowding problem khi dữ liệu được nén vào không gian thấp chiều.

7. **Làm thế nào t-SNE có thể giúp trong việc trực quan hóa dữ liệu?** t-SNE giúp trực quan hóa dữ liệu bằng cách giảm chiều dữ liệu xuống 2 hoặc 3 dimensions, cho phép dễ dàng

quan sát các cụm, xu hướng và mối quan hệ trong dữ liệu mà không thể thấy được trong không gian cao chiều.

0.0.8 Input và Output của t-SNE

8. **Dữ liệu đầu vào cho t-SNE yêu cầu những gì?** Dữ liệu đầu vào cho t-SNE yêu cầu phải là một ma trận dữ liệu, trong đó các hàng là các mẫu và các cột là các biến. Dữ liệu cần phải được chuẩn hóa để đảm bảo tính chính xác của các xác suất có điều kiện.
9. **Dữ liệu đầu vào có cần phải chuẩn hóa không? Tại sao?** Có, dữ liệu đầu vào cần phải được chuẩn hóa để đảm bảo rằng tất cả các biến có cùng mức độ quan trọng và để tránh việc một biến có ảnh hưởng quá lớn đến kết quả của t-SNE.
10. **t-SNE tạo ra những output gì từ dữ liệu đầu vào?** t-SNE tạo ra một ma trận dữ liệu mới có số chiều thấp hơn (thường là 2 hoặc 3 dimensions), trong đó các hàng vẫn là các mẫu gốc nhưng các cột là các tọa độ của các điểm dữ liệu trong không gian giảm chiều.
11. **Làm thế nào để chọn số lượng dimensions cần giữ lại trong t-SNE?** Số lượng dimensions cần giữ lại trong t-SNE thường được chọn là 2 hoặc 3 để dễ dàng trực quan hóa. Tuy nhiên, có thể chọn số lượng dimensions khác tùy theo mục đích cụ thể của phân tích.
12. **Tại sao t-SNE thường được sử dụng để giảm chiều dữ liệu xuống 2 hoặc 3 dimensions?** t-SNE thường được sử dụng để giảm chiều dữ liệu xuống 2 hoặc 3 dimensions để trực quan hóa dữ liệu một cách dễ dàng và hiệu quả. Không gian 2D hoặc 3D giúp con người dễ dàng quan sát và hiểu được cấu trúc dữ liệu.
13. **Perplexity là gì và nó ảnh hưởng như thế nào đến kết quả của t-SNE?** Perplexity là một siêu tham số trong t-SNE, xác định số lượng hàng xóm gần nhất mà t-SNE sẽ xem xét khi tính toán các xác suất có điều kiện. Perplexity ảnh hưởng đến mức độ nhạy bén của t-SNE đối với cấu trúc dữ liệu cục bộ. Giá trị quá nhỏ hoặc quá lớn của perplexity có thể dẫn đến kết quả không tối ưu.

0.0.9 Thuật toán triển khai t-SNE

14. **Các bước chính của thuật toán t-SNE là gì?** Các bước chính của thuật toán t-SNE bao gồm:
    1. Chuẩn hóa dữ liệu.
    2. Tính toán các xác suất có điều kiện giữa các điểm trong không gian gốc.
    3. Sử dụng phân phối t để tính các xác suất có điều kiện giữa các điểm trong không gian giảm chiều.
    4. Tối thiểu hóa hàm mất mát Kullback-Leibler divergence giữa các xác suất có điều kiện trong không gian gốc và không gian giảm chiều.
    5. Cập nhật các tọa độ của các điểm dữ liệu trong không gian giảm chiều để

tối thiểu hóa hàm mất mát.

15. **Tại sao chúng ta cần tính toán các xác suất có điều kiện giữa các điểm trong không gian gốc?** Chúng ta cần tính toán các xác suất có điều kiện giữa các điểm trong không gian gốc để biểu diễn mối quan hệ giữa các điểm dữ liệu theo cách xác suất, giúp giữ lại cấu trúc cục bộ khi chuyển đổi sang không gian giảm chiều.

16. **Làm thế nào để tính toán các xác suất có điều kiện giữa các điểm trong không gian giảm chiều?** Trong không gian giảm chiều, t-SNE sử dụng phân phối t để tính toán các xác suất có điều kiện giữa các điểm. Điều này giúp giảm thiểu vấn đề crowding problem và duy trì khoảng cách giữa các điểm dữ liệu.

17. **Làm thế nào để tối thiểu hóa hàm mất mát Kullback-Leibler divergence trong t-SNE?** t-SNE tối thiểu hóa hàm mất mát Kullback-Leibler divergence thông qua các thuật toán tối ưu hóa như gradient descent. Hàm mất mát đo lường sự khác biệt giữa các xác suất có điều kiện trong không gian gốc và không gian giảm chiều, và việc tối thiểu hóa nó giúp duy trì các mối quan hệ cục bộ và toàn cục.

18. **Tại sao t-SNE sử dụng phân phối t với một bậc tự do (Student's t-distribution)?** t-SNE sử dụng phân phối t với một bậc tự do để tính toán các xác suất có điều kiện trong không gian giảm chiều vì phân phối này có đuôi dài, giúp giảm thiểu crowding problem và giữ khoảng cách giữa các điểm dữ liệu khi chúng được nén vào không gian thấp chiều.

19. **Làm thế nào để chọn giá trị cho các siêu tham số như perplexity và learning rate?** Giá trị của các siêu tham số như perplexity và learning rate thường được chọn thông qua thử nghiệm và đánh giá. Perplexity thường nằm trong khoảng từ 5 đến 50, trong khi learning rate thường bắt đầu từ giá trị mặc định và điều chỉnh dựa trên kết quả quan sát được.

20. **Sự khác biệt giữa t-SNE và PCA là gì?** t-SNE là một kỹ thuật giảm chiều phi tuyến tính, trong khi PCA là một kỹ thuật giảm chiều tuyến tính. t-SNE giữ lại các mối quan hệ cục bộ giữa các điểm dữ liệu, trong khi PCA giữ lại các biến có phương sai lớn nhất. t-SNE thường được sử dụng để trực quan hóa dữ liệu, trong khi PCA thường được sử dụng để giảm chiều và trích xuất đặc trưng.

0.0.10 Ưu điểm và nhược điểm của t-SNE

21. **Những ưu điểm chính của t-SNE là gì?**

    - Khả năng trực quan hóa dữ liệu phức tạp trong không gian 2D hoặc 3D.
    - Giữ lại các mối quan hệ cục bộ giữa các điểm dữ liệu.
    - Hiệu quả trong việc phát hiện cụm và cấu trúc tiềm ẩn trong dữ liệu.

22. **t-SNE có những nhược điểm gì?**

    - Tốn nhiều thời gian tính toán và bộ nhớ với dữ liệu lớn.
    - Kết quả có thể không ổn định và phụ thuộc vào các siêu tham số như perplexity và learning rate.
    - Khó khăn trong việc diễn giải các principal components do tính phi tuyến tính.

23. **t-SNE có thể gặp vấn đề gì khi dữ liệu đầu vào không được chuẩn hóa?** Khi dữ liệu đầu vào không được chuẩn hóa, các biến có quy mô lớn hơn có thể ảnh hưởng quá mức đến kết quả, dẫn đến các principal components không phản ánh đúng mối quan hệ giữa các điểm dữ liệu.

24. **Tại sao t-SNE không thể làm việc tốt với dữ liệu rất lớn?** t-SNE có độ phức tạp tính toán cao và yêu cầu nhiều bộ nhớ, làm cho việc xử lý dữ liệu rất lớn trở nên chậm chạp và không hiệu quả. Các phương pháp gần đúng như Barnes-Hut t-SNE có thể cải thiện nhưng vẫn có giới hạn.

25. **t-SNE có thể làm giảm thông tin quan trọng nào trong dữ liệu?** Nếu không chọn đúng các siêu tham số, t-SNE có thể không giữ lại đầy đủ thông tin toàn cục, dẫn đến việc mất mát thông tin quan trọng về cấu trúc tổng thể của dữ liệu.

26. **Ưu điểm của việc sử dụng t-SNE trong việc trực quan hóa dữ liệu là gì?** t-SNE giúp phát hiện và trực quan hóa các cụm và cấu trúc tiềm ẩn trong dữ liệu một cách rõ ràng và dễ hiểu, ngay cả khi dữ liệu có chiều cao và cấu trúc phức tạp.

27. **Tại sao kết quả của t-SNE có thể không ổn định khi chạy lại nhiều lần?** Kết quả của t-SNE có thể không ổn định do thuật toán sử dụng các yếu tố ngẫu nhiên trong quá trình tối ưu hóa. Điều này dẫn đến các khác biệt nhỏ trong kết quả khi chạy lại thuật toán với cùng một bộ dữ liệu và siêu tham số.

28. **t-SNE có thể được cải thiện bằng cách sử dụng các kỹ thuật nào khác?**
    - Barnes-Hut t-SNE: một phiên bản t-SNE cải tiến giúp tăng tốc độ tính toán.
    - Uniform Manifold Approximation and Projection (UMAP): một phương pháp giảm chiều phi tuyến tính tương tự t-SNE nhưng thường nhanh hơn và có thể cho kết quả ổn định hơn.
    - Sử dụng các kỹ thuật học sâu như autoencoders để giảm chiều dữ liệu trước khi áp dụng t-SNE.

0.0.11 Ứng dụng và ví dụ của t-SNE

29. **Một số ứng dụng thực tế của t-SNE trong phân tích dữ liệu là gì?**
    - Trực quan hóa dữ liệu trong học máy và khoa học dữ liệu.
    - Phân tích gene và phát hiện cụm trong dữ liệu sinh học.
    - Nhận dạng mẫu và phân loại dữ liệu.
    - Phân tích thị trường tài chính và khám phá cấu trúc dữ liệu trong dữ liệu tài chính.

30. **Làm thế nào t-SNE có thể được sử dụng trong lĩnh vực nhận dạng mẫu?** t-SNE có thể giảm chiều dữ liệu đầu vào, giúp trực quan hóa và xác định các cụm trong dữ liệu, từ đó cải thiện hiệu suất của các thuật toán nhận dạng mẫu như SVM, KNN, và neural networks.

31. **t-SNE có thể giúp gì trong việc phát hiện và xử lý dữ liệu ngoại lai?** t-SNE giúp phát hiện dữ liệu ngoại lai bằng cách giảm chiều dữ liệu và xác định các điểm nằm xa so với các cụm dữ liệu chính. Những điểm này có thể là các ngoại lai trong dữ liệu.

32. **Làm thế nào t-SNE có thể giúp trong việc trực quan hóa dữ liệu?** t-SNE giúp giảm chiều dữ liệu xuống 2 hoặc 3 dimensions, cho phép dễ dàng trực quan hóa dữ liệu trong không gian thấp chiều, giúp phát hiện các cụm, xu hướng và mối quan hệ trong dữ liệu.

33. **Một ví dụ cụ thể về việc sử dụng t-SNE trong xử lý ảnh là gì?** Trong xử lý ảnh, t-SNE có thể được sử dụng để giảm chiều dữ liệu của các đặc trưng trích xuất từ ảnh, giúp phát hiện và trực quan hóa các cụm ảnh có đặc điểm tương tự, từ đó hỗ trợ các nhiệm vụ như phân loại ảnh, nhận dạng đối tượng và phát hiện ngoại lai.

0.0.12 Đánh giá output thu được từ t-SNE

34. **Làm thế nào để đánh giá chất lượng của output thu được từ t-SNE?**
    - **Visual Inspection**: Trực quan hóa kết quả để xem các cụm và cấu trúc có rõ ràng và hợp lý không.

- **K-Nearest Neighbors (KNN) Preservation**: Kiểm tra xem các hàng xóm gần nhất trong không gian gốc có được bảo toàn trong không gian giảm chiều hay không.
- **Trustworthiness and Continuity**: Đo lường mức độ bảo toàn của các mối quan hệ cục bộ và toàn cục giữa các điểm dữ liệu trong không gian giảm chiều.
- **Silhouette Score**: Đánh giá mức độ tách biệt của các cụm trong không gian giảm chiều.
- **Perplexity and Learning Rate Tuning**: Thử nghiệm và điều chỉnh các giá trị của perplexity và learning rate để đảm bảo kết quả ổn định và hợp lý.
- **Stability Analysis**: Chạy t-SNE nhiều lần với cùng bộ dữ liệu và các siêu tham số để kiểm tra tính ổn định của kết quả.

Ví dụ về mã R để kiểm tra trustworthiness của t-SNE:

```r
library(Rtsne)
library(cluster)

# Tạo dữ liệu mẫu
set.seed(123)
data <- matrix(rnorm(100*5), ncol=5)

# Chuẩn hóa dữ liệu
data_scaled <- scale(data)

# Thực hiện t-SNE
tsne_result <- Rtsne(data_scaled, perplexity = 30, dims = 2)

# Tính toán trustworthiness
trustworthiness <- function(X, Y, k) {
  n <- nrow(X)
  D_X <- as.matrix(dist(X))
  D_Y <- as.matrix(dist(Y))

  rank_X <-

 apply(D_X, 1, rank)
  rank_Y <- apply(D_Y, 1, rank)

  TW <- 0
  for (i in 1:n) {
    for (j in 1:n) {
      if (rank_X[i, j] <= k && rank_Y[i, j] > k) {
        TW <- TW + (rank_Y[i, j] - k)
      }
    }
  }
  TW <- 1 - (2 / (n * k * (2 * n - 3 * k - 1))) * TW
  return(TW)
}
```

```
trustworthiness_score <- trustworthiness(data_scaled, tsne_result$Y, k = 10)
cat("Trustworthiness Score:", trustworthiness_score, "\n")
```

Mã trên tính toán trustworthiness score để đánh giá mức độ bảo toàn của các mối quan hệ cục bộ sau khi áp dụng t-SNE. Trustworthiness score càng gần 1 thì chất lượng giảm chiều càng tốt.