Trong bài viết này chúng ta cùng tìm hiểu về NMF và ứng dụng cho bài toán phân cụm văn bản.

Trước hết ta xét bài toán phân cụm văn bản như sau:

Cho một dataset D trong đó có n văn bản. Sau khi tiền xử lý ta có ma trận X gồm n dòng và m cột, m cột này tương ứng với tập tất cả các từ xuất hiện trong dataset. Lưu ý để có được ma trận này dataset thường đã qua bước tiền xử lý bao gồm chuyển câu thành từ, loại bỏ các dấu, các từ không quan trọng, ... Bạn có thể tìm hiểu thêm về tiền xử lý dữ liệu văn bản tại [https://towardsdatascience.com/machine-learning-text-processing-1d5a2d638958], [https://www.kdnuggets.com/2017/12/general-approach-preprocessing-text-data.html].

Lưu ý khác là với dữ liệu văn bản, thường sẽ thưa (có nhiều giá trị 0) và có số chiều rất lớn, do đó trực tiếp áp dụng giải thuật phân cụm như K-means có thể sẽ không hiệu quả. Vì vậy, ta cùng tìm hiểu giải thuật hiệu quả hơn, đó là giải quyết bài toán phân cụm bằng NMF, ta cùng tìm hiểu kỹ thuật NMF nhé.

1. **Non-negative matrix factorization** (**NMF)**

Một cách ngắn gọn, có thể hiểu NMF là một kỹ thuật giảm số chiều (dimensionality reduction) của dữ liệu mà không làm mất các đặc trưng chính của dữ liệu.

1. Định nghĩa NMF

Cho ma trận X với n dòng, m cột. Kỹ thuật NMF sẽ tìm 2 (hoặc 3) ma trận W, H sao cho, tích của HW sẽ cho trở lại giá trị xấp xỉ với X, nghĩa là, với ma trận X được cho trước, tìm H, W sao cho,

X ≈ HW.

H nxk, W k x m

Lưu ý rằng, k có giá trị rất nhỏ so với n và m. Các ma trận X, H, W đều có ràng buộc không âm, nghĩa là giá trị của các phần tử (element) trong các ma trận này >=0.

1. Làm sao để tìm được hai ma trận H và W như vậy?

Ta cần giải thuật tối ưu hóa hàm mục tiêu sau:

min |X - HW|

Khi hàm mục tiêu này được tối ưu hóa, nghĩa là tiến dần về 0, khi đó ta có X ~ HW.

Để tối ưu hóa hàm mục tiêu J, có thể dùng gradient descent hoặc multiplicative update rule.

Bản chất cả hai phương pháp tối ưu này giống nhau ở chỗ, giải thuật sẽ bắt đầu bằng việc khởi tạo giá trị không âm ngẫu nhiên cho W, H rồi lặp lại quá trình cập nhật giá trị cho H và W cho đến khi giá trị hàm mục tiêu hội tụ (hội tụ nghĩa là giá trị hàm mục tiêu không thay đổi hoặc thay đổi rất rất nhỏ sau mỗi lần lặp).

H và W sẽ được cập nhật theo công thức sau:

Minh họa giải thuật NMF như sau:

Lưu ý rằng:

Hai ma trận đầu ra H (n x k) và W (k x m) được giải thích như sau: Ma trận W được gọi là ma trận cơ sở, H được gọi là ma trận trọng số. Trong kỹ thuật NMF, ma trận H cũng được chứng minh là ma trận biểu diễn X qua ma trận cơ sở W, nghĩa là thông qua giải thuật NMF, ta tìm được ma trận H, có đặc trưng của xấp xỉ ma trận X nhưng với số chiều nhỏ hơn rất rất nhiều, ví dụ trong thực tế m có thể có giá trị hàng triệu, trong khi giá trị của k chỉ khoảng 10, 20 hoặc nhiều nhất là 100, 200 cho một số dataset.

**Áp dụng NMF cho bài toán phân cụm**

Sau khi đã áp dụng NMF trên ma trận đầu vào X, ta có đầu ra là 2 ma trận H (n x k) và W (k x m). Như bạn có thể đoán, lúc này ta có được ma trận H là ma trận biểu diễn X. Lúc này thay vì thực hiện phân cụm trên dữ liệu gốc với số chiều cao và do đó việc phân cụm rất khó khăn và không đảm bảo, ta thực hiện việc phân cụm trên H bằng cách áp dụng k-means tại bước này để thu được các cụm.

@inproceedings{example1,

title={ A Novel Approach to Learning Consensus and Complementary Information for Multi-View Data Clustering },

author={ Luong, Khanh and Nayak, Richi },

booktitle={2020 IEEE 36th International Conference on Data Engineering (ICDE)},

pages={1--6},

year={2020},

organization={IEEE}

}

@inproceedings{9101762,

author={Luong, Khanh and Nayak, Richi},

booktitle={2020 IEEE 36th International Conference on Data Engineering (ICDE)},

title={A Novel Approach to Learning Consensus and Complementary Information for Multi-View Data Clustering},

year={2020},

volume={},

number={},

pages={865-876},

doi={10.1109/ICDE48307.2020.00080}}