

Hide menu

Lecture: Vector Space Models

- Video: Week Introduction47 sec
- Video: Vector Space Models2 min
- Reading: Vector Space Models10 min
- Video: Word by Word and Word by Doc.4 min

Reading: Word by Word and Word by Doc.10 min

Lab: Linear algebra in Python with Numpy1h

Video: Euclidean Distance3 min

Reading: Euclidian Distance10 min

Video: Cosine Similarity: Intuition2 min

Reading: Cosine Similarity: Intuition10 min

Video: Cosine Similarity3 min

Reading: Cosine Similarity10 min

Video: Manipulating Words in Vector Spaces3 min

Reading: Manipulating Words in Vector Spaces10 min

Lab: Manipulating word embeddings1h

Video: Visualization and PCA3 min

Reading: Visualization and PCA10 min

Video: PCA Algorithm3 min

Reading: PCA algorithm10 min

Lab: Another explanation about PCA1h

Reading: The Rotation Matrix (Optional Reading)10 min

Video: Week Conclusion46 sec

Lecture Notes (Optional)

Practice Quiz

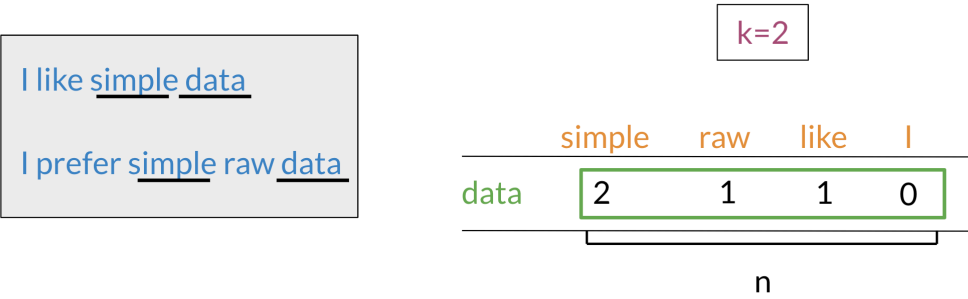
Assignment: Vector Space Models

Week 3 > Word by Word and Word by Doc.

Word by Word and Word by Doc.

Word by Word Design

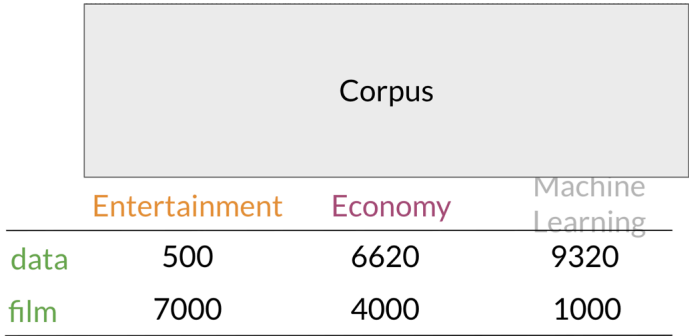
We will start by exploring the word by word design. Assume that you are trying to come up with a vector that will represent a certain word. One possible design would be to create a matrix where each row and column corresponds to a word in your vocabulary. Then you can iterate over a document and see the number of times each word shows up next each other word. You can keep track of the number in the matrix. In the video I spoke about a parameter  $K$ . You can think of  $K$  as the bandwidth that decides whether two words are next to each other or not.



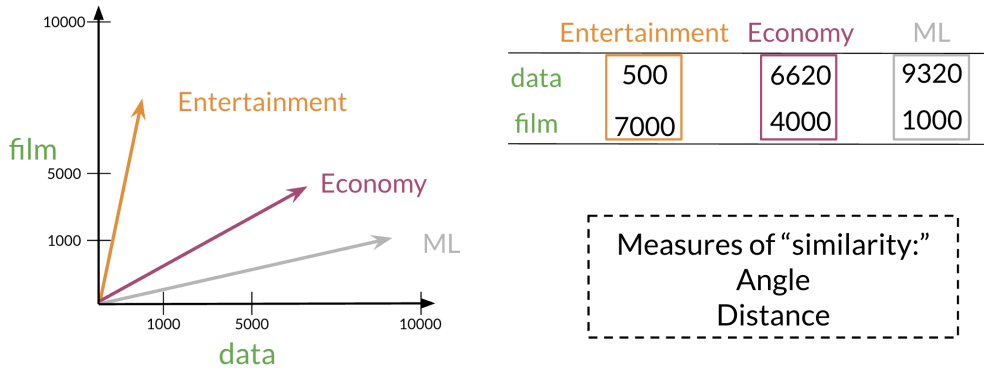
In the example above, you can see how we are keeping track of the number of times words occur together within a certain distance  $k$ . At the end, you can represent the word data, as a vector  $v = [2, 1, 1, 0]$ .

Word by Document Design

You can now apply the same concept and map words to documents. The rows could correspond to words and the columns to documents. The numbers in the matrix correspond to the number of times each word showed up in the document.



You can represent the entertainment category, as a vector  $v = [500, 7000]$ . You can then also compare categories as follows by doing a simple plot.



Later this week, you will see how you can use the angle between two vectors to measure similarity.

Previous Next

