Lecture: Vector Space Models

- **Video:** Week Introduction 47 sec
- **Video:** Vector Space Models 2 min
- **Reading:** Vector Space Models 10 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 Numpy 1h
- **Video:** Euclidean Distance 3 min
- **Reading:** Euclidian Distance 10 min
- **Video:** Cosine Similarity: Intuition 2 min
- **Reading:** Cosine Similarity: Intuition 10 min
- **Video:** Cosine Similarity 3 min
- **Reading:** Cosine Similarity 10 min
- **Video:** Manipulating Words in Vector Spaces 3 min
- **Reading:** Manipulating Words in Vector Spaces 10 min
- **Lab:** Manipulating word embeddings 1h
- **Video:** Visualization and PCA 3 min
- **Reading:** Visualization and PCA 10 min
- **Video:** PCA Algorithm 3 min
- **Reading:** PCA algorithm 10 min
- **Lab:** Another explanation about PCA 1h
- **Reading:** The Rotation Matrix (Optional Reading) 10 min
- **Video:** Week Conclusion 46 sec

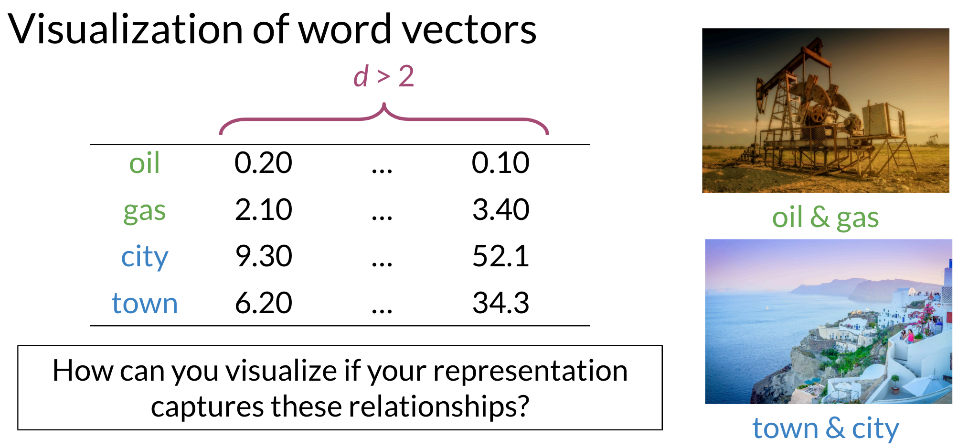
Lecture Notes (Optional)

Practice Quiz

Assignment: Vector Space Models

Visualization and PCA

Principal component analysis is an unsupervised learning algorithm which can be used to reduce the dimension of your data. As a result, it allows you to visualize your data. It tries to combine variances across features. Here is a concrete example of PCA:



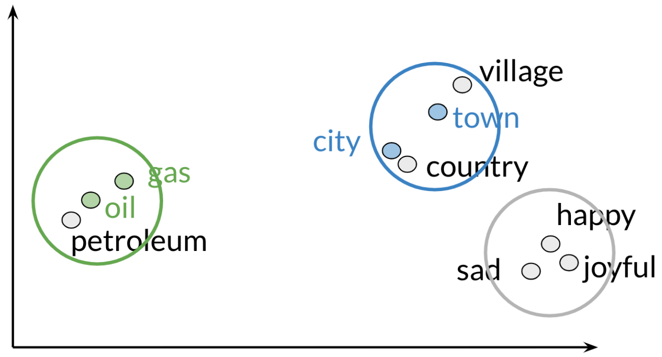
Visualization of word vectors

$\overbrace{\qquad\qquad\qquad}^{d > 2}$

oil	0.20	...	0.10
gas	2.10	...	3.40
city	9.30	...	52.1
town	6.20	...	34.3

How can you visualize if your representation captures these relationships?

Note that when doing PCA on this data, you will see that oil & gas are close to one another and town & city are also close to one another. To plot the data you can use PCA to go from $d > 2$ dimensions to $d = 2$.



Those are the results of plotting a couple of vectors in two dimensions. Note that words with similar part of speech (POS) tags are next to one another. This is because many of the training algorithms learn words by identifying the neighboring words. Thus, words with similar POS tags tend to be found in similar locations. An interesting insight is that synonyms and antonyms tend to be found next to each other in the plot. Why is that the case?

Mark as completed

Like Dislike Report an issue