Topic Modeling:

Non-Negative Matrix Factorization (NMF)

Frank Burkholder (credit T. Heilman, E. Wellinger, C. Goltermann)

Objectives

- Review PCA
- Review SVD
- Compare hard and soft clustering techniques
- Define topic modeling
- Explain what NMF is
- Describe how the non-negativity constraint distinguishes NMF from PCA and SVD
- Describe how Alternating Least Squares Method solves for matrix values in NMF
- Provide use case examples of NMF

PCA

- What does it stand for?
- Conceptually, what is it?
- Why do you do it?
- What is the process you generally follow when performing PCA?
- What are eigenvectors?
- What are eigenvalues?
- How do you pick the number of principal components?

SVD

- What does it stand for?
- Conceptually, what is it?
- Why do you do it?
- SVD is called a Matrix Factorization technique. Why?
- Describe the matrices that result from SVD, and how rows and columns in an original matrix X are related to latent topics in the factored matrices
- How do you perform dimensionality reduction in SVD?
- Are SVD and PCA related? How? Here's help.

Clustering - hard and soft

Clustering involves assigning data points to clusters such that items in the same cluster are as similar as possible, while items belonging to different clusters are as dissimilar as possible.

Hard clustering: Each observation (row of data) can belong to only one cluster, e.g. article is sports, or politics, or finance.

Soft or “fuzzy” clustering: Each observation (row of data) can belong to multiple clusters, e.g. article is a mixture of sports, politics, and finance.

What kind of clustering do you get with Kmeans?

How about SVD?

Clustering - hard and soft

Clustering involves assigning data points to clusters such that items in the same cluster are as similar as possible, while items belonging to different clusters are as dissimilar as possible.

Hard clustering: Each observation (row of data) can belong to only one cluster.
*Example: **Kmeans***

Soft or "fuzzy" clustering: Each observation (row of data) can belong to multiple clusters.
*Examples: singular value decomposition (**SVD**), non-negative matrix factorization (**NMF**), latent dirichlet allocation (**LDA**), gaussian mixture models (**GMM**)*

Clustering results often become additional features in matrices for supervised learning. This is an example of model stacking.

Topic Modeling

Especially in the case of text, **clusters** are referred to as **topics**.

Paraphrased from Wikipedia:
...a topic model is a type of statistical model for discovering the abstract "topics" that occur in a collection of documents.

...given that a document is about a topic, some words appear in the document more or less frequently: "dog" and "bone" will appear more often in documents about dogs, "cat" and "meow" will appear in documents about cats, and "the" and "is" will appear equally in both…

...a document typically concerns multiple topics in different proportions…

...the "topics" produced by topic modeling techniques are clusters of words that tend to occur together in the topic. The topic is a probability distribution of words.

Topic Modeling

Algorithms typically used in topic modeling:

- Singular Value Decomposition (SVD)
- Latent Semantic Analysis (LSA)
- Non-negative Matrix Factorization (NMF)
- Latent Dirichlet Allocation

Topic Modeling Examples

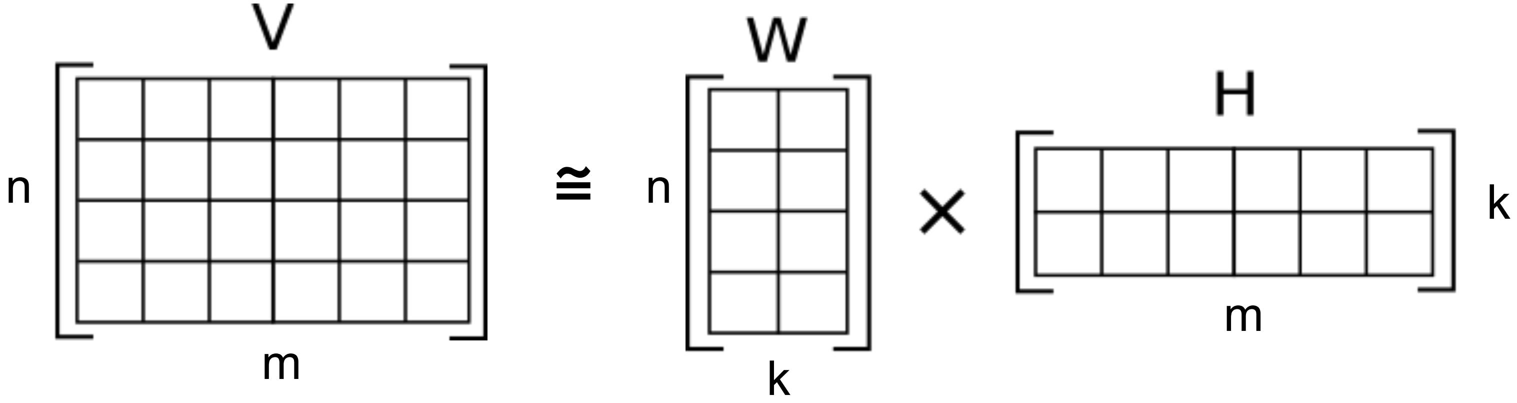
Past capstones:

- Spice Blends (a spice blend is a topic formed by a combination of spices occuring in recipe documents - NMF)
- News Topics in the 2016 U.S. Election Cycle (topics are combinations of words found in news articles - NMF)
- Topic Analysis of the Enron Email Corpus (topics are combinations of words found in the emails - LDA)
- DonorsChoose Recommender (topics are combinations of words found in school projects - LDA)

NMF (Non-Negative Matrix Factorization)

It’s a matrix factorization technique where a large matrix V is factored into two smaller matrices W and H, where **all matrices are constrained to contain only zero or positive values**. This is typically **true in ratings matrices and tf-idf matrices**.

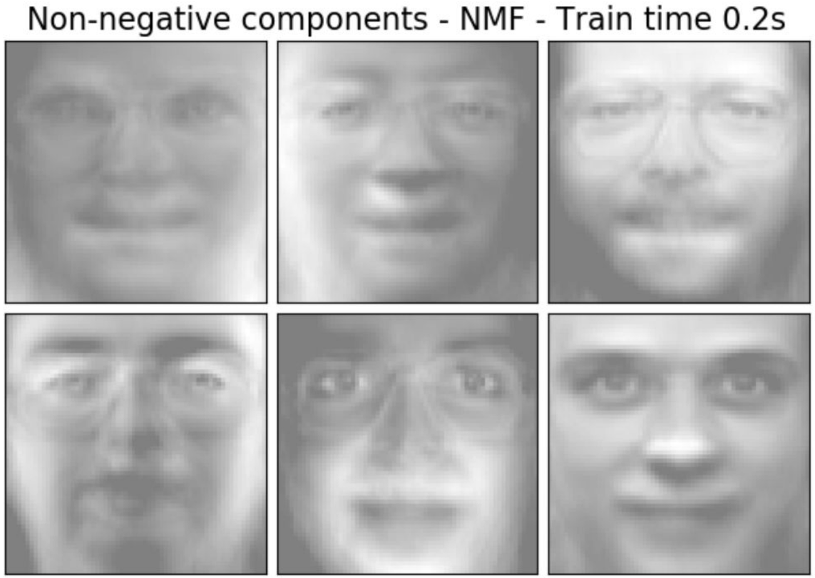
$$V_{n*m} \cong W_{n*k}H_{k*m}$$



Wikipedia

Benefits of NMF non-negativity constraint

- Leads to a parts-based representation of the items of interest where the additive combinations of latent topics combine to make the whole.
- Applications: Facial recognition, linguistics, topic modeling, recommenders



source

Words in NMF discovered topics:

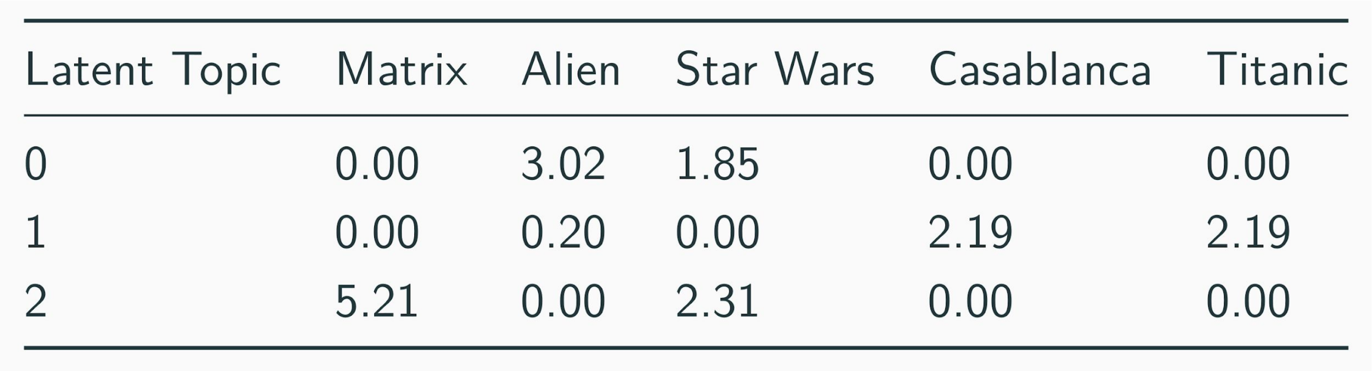
Topic 1:
[“ruler”, “monarch”, “king”, “president”]

Topic 2:
[“ruler”, “protractor”, “calculator”, “abacus”]

In the first case “ruler” corresponds to a leadership topic, but in the second case the ruler is used to measure things.

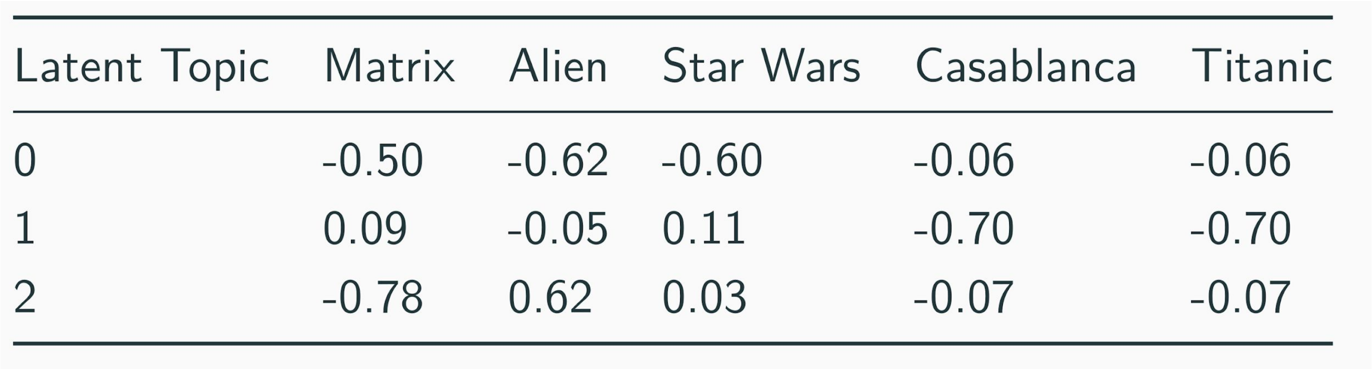
NMF representations are often sparse

NMF



Latent Topic	Matrix	Alien	Star Wars	Casablanca	Titanic
0	0.00	3.02	1.85	0.00	0.00
1	0.00	0.20	0.00	2.19	2.19
2	5.21	0.00	2.31	0.00	0.00

SVD



Latent Topic	Matrix	Alien	Star Wars	Casablanca	Titanic
0	-0.50	-0.62	-0.60	-0.06	-0.06
1	0.09	-0.05	0.11	-0.70	-0.70
2	-0.78	0.62	0.03	-0.07	-0.07

Let’s find movie topics!

Please fill out your movie preferences (1-10).

Leave unseen movies blank.

See NMF in action

`NMF_notebook.ipynb`

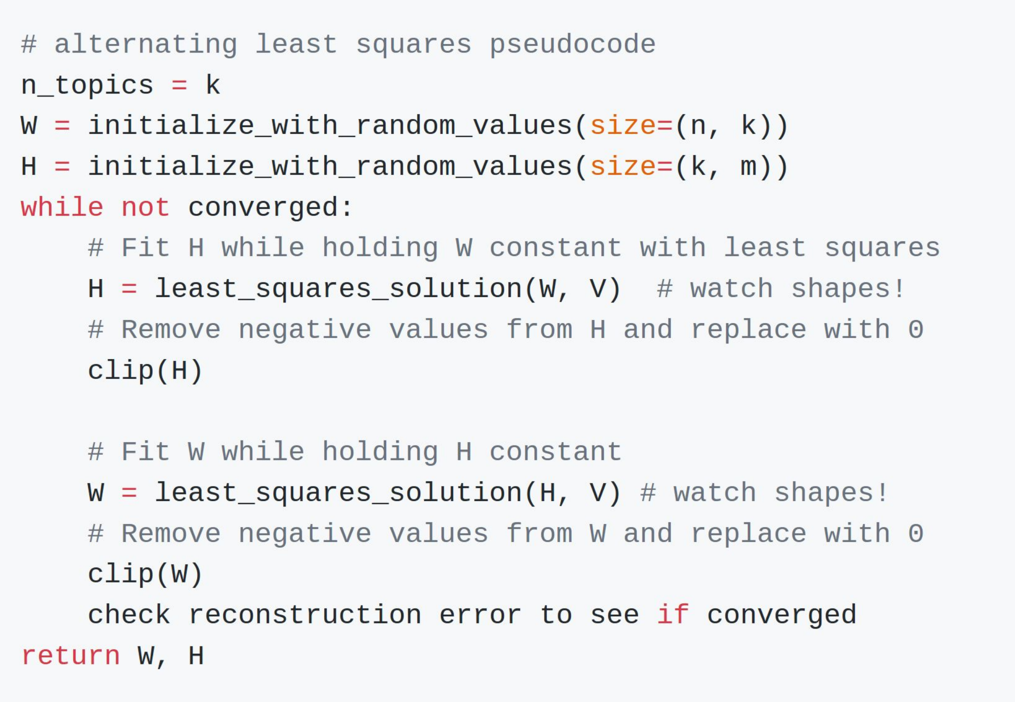
NMF algorithm

$$V_{n*m} \cong W_{n*k}H_{k*m}$$

- All elements of V, W, and H must be non-negative.
- k <= min(n, m)
- Solution will always be approximate (solution method stochastic!), unlike PCA/SVD which will give exact, orthogonal topic vectors. The vectors in W and H are not guaranteed to be orthogonal.
- Goal is to find values in W and H so that reconstruction error $||V - WH||^2$ is minimized. (See Frobenius norm, Numpy `np.linalg.norm`)

NMF algorithm - reducing reconstruction error

Will use Alternating Least Squares to step-by-step solve for H, then W, then H, then W, etc., until a desired number of iterations have occurred, or the reconstruction error has been minimized to some desired value.



```
# alternating least squares pseudocode
n_topics = k
W = initialize_with_random_values(size=(n, k))
H = initialize_with_random_values(size=(k, m))
while not converged:
    # Fit H while holding W constant with least squares
    H = least_squares_solution(W, V)  # watch shapes!
    # Remove negative values from H and replace with 0
    clip(H)

    # Fit W while holding H constant
    W = least_squares_solution(H, V) # watch shapes!
    # Remove negative values from W and replace with 0
    clip(W)
    check reconstruction error to see if converged
return W, H
```

See code at end of jupyter notebook to help you code this.

Choosing the number of topics, k

- Plot the reconstruction error for different values of k (elbow plot).
- Look at the cosine similarity of items within topics (should be similar) and between topics (should be dissimilar).
- Experience / domain knowledge / consulting the literature.
- What value of k is useful to you? Try it.

Objectives

- Review PCA
- Review SVD
- Compare hard and soft clustering techniques
- Define topic modeling
- Explain what NMF is
- Describe how the non-negativity constraint distinguishes NMF from PCA and SVD
- Describe how Alternating Least Squares Method solves for matrix values in NMF
- Provide use case examples of NMF