Non-Negative Matrix Factorization

Clayton W. Schupp, Galvanize

June 2015

NMF Overview

What is NMF

- A group of algorithms that takes a large dimensional matrix and factors into 2 (or more) smaller dimensional matrices
- The matrices are all non-negative
  - Often inherent in the data being considered
  - Makes the resulting matrices easier to inspect and interpret
- This smaller dimension makes the matrices easier to store

NMF Overview

Matrix $V_{m \times n}$ where each entry $v_{ij} \geq 0$

$$\underset{m \times r}{W} * \underset{r \times n}{H} = \underset{m \times n}{V}$$

also $w_{ij} \geq 0$

$h_{ij} \geq 0$

r is set by the user and can be substantially smaller than either (m, n)

NMF Overview

Matrix $V_{m \times n}$ where each entry $v_{ij} \geq 0$

$$\underset{m \times r}{W} * \underset{r \times n}{H} = \underset{m \times n}{V}$$

also $w_{ij} \geq 0$
$h_{ij} \geq 0$

The columns of V are linear combinations of the columns of W weighted by the corresponding column in H

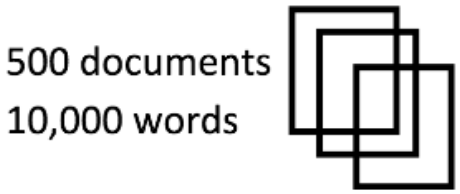
$$v_i = W * h_i$$

where v_i and h_i are the i^{th} column vectors of V and H respectively

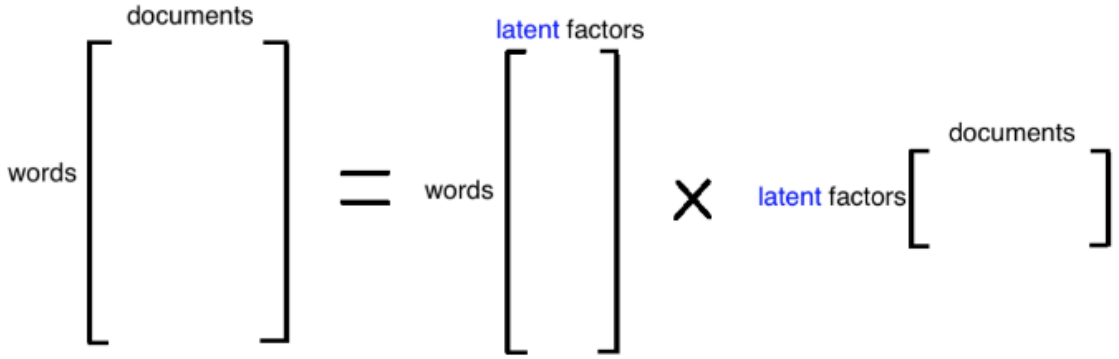
Text Mining Example

- Let V be the input matrix with 10000 rows and 500 columns, where
  - words are rows
  - documents are columns
- Assume we want to find 10 hidden features. Want to generate:
  - W with 10000 rows and 10 columns
  - H with 10 rows and 500 columns

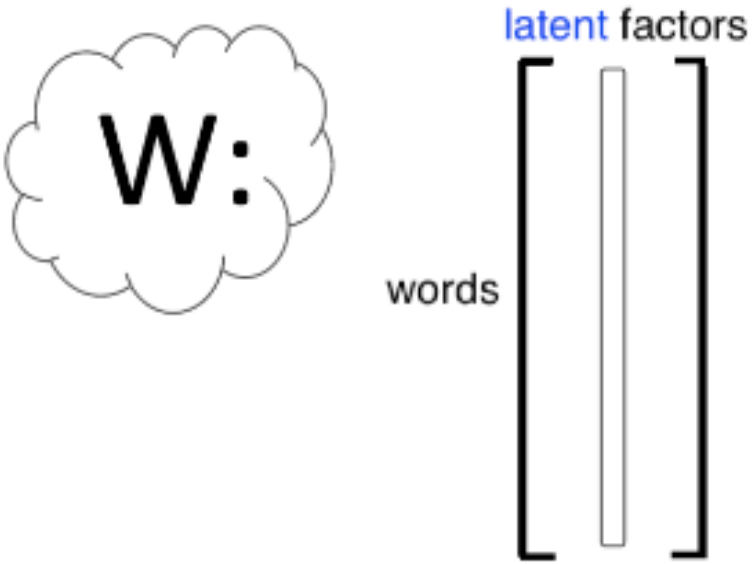
Document Clustering with NMF



$$V = W * H$$



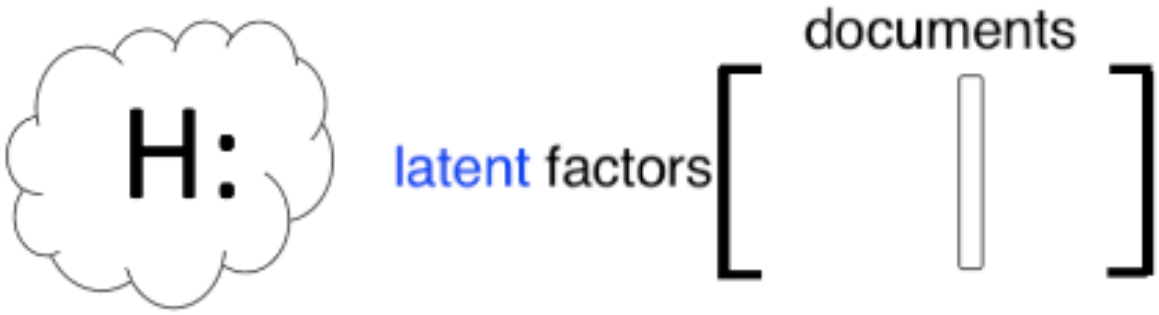
Document Clustering with NMF



Column in W:

- document archetype
- word's cell value establishes it's rank for that latent feature

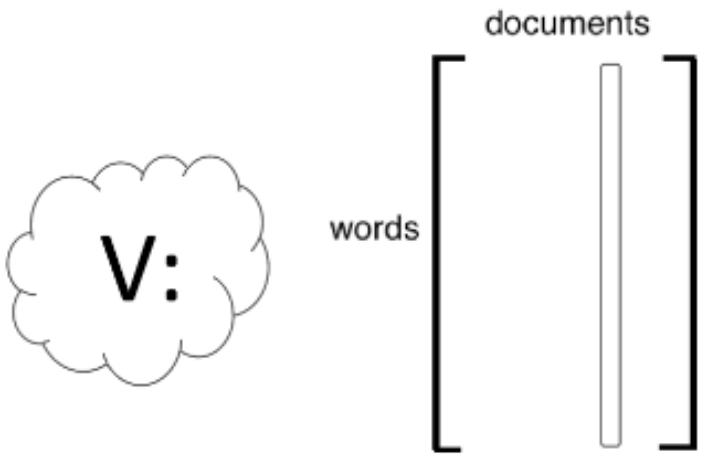
Document Clustering with NMF



Column in H:

- Original document
- Cell value is document's rank for that latent feature

Document Clustering with NMF



Recall $v_i = W * h_i$, so v_i:

- is rebuilding a particular document
- linear combination of document archetypes weighted by how important they are

Algorithm

Want to minimize RMSE

Minimize $||V - WH||^2$ with respect to W and H
subject to W, H ≥ 0

- Lee and Seung's multiplicative update rules
  - good compromise between speed and ease of implementation
- Additive update rules such as gradient descent
  - easiest to implement but convergence can be slow
  - convergence is sensitive to choice of step size

Algorithm

Minimize $||V - WH||^2$ with respect to W and H
subject to W, H ≥ 0

Multiplicative Update Steps:

- Start with some random W and H
- Repeatedly adjust W and H to make RMSE smaller

$$H_{a\mu} \longleftarrow H_{a\mu} \frac{(W^T V)_{a\mu}}{(W^T WH)_{a\mu}} \qquad W_{ia} \longleftarrow W_{ia} \frac{(VH^T)_{ia}}{(WHH^T)_{ia}}$$

- Stop when some threshold is met