Non-negative matrix factorization for analysing high-dimensional datasets

The 3rd Machine Learning & AI in Quantitative Finance Conference
22nd March 2019

Edvin Hopkins
Technical Consultant
edvin.hopkins@nag.co.uk
NAG Library

- Over 1700 mathematical and statistical routines
  - Used from Python, Java, C++, Fortran, .NET, R, MATLAB, ...
- Decades of experience in statistics, data mining, ML
- Most major investment banks have global NAG licences
- Machine learning at NAG:
  - Clustering, PCA, times series, optimization, regression ...
- HPC consulting
Motivation

https://www.bbc.co.uk/news/uk-politics-47031312
https://www.bbc.co.uk/news/world-us-canada-47477727
https://www.bbc.co.uk/news/uk-politics-46393399
https://www.bbc.co.uk/news/world-us-canada-47633940
https://www.bbc.co.uk/news/uk-politics-47627744
https://www.bbc.co.uk/news/uk-politics-parliaments-47653160
https://www.bbc.co.uk/news/world-us-canada-47642335
https://www.bbc.co.uk/news/uk-politics-47660019
https://www.bbc.co.uk/news/world-middle-east-47657843
https://www.bbc.co.uk/news/uk-politics-47659410
https://www.bbc.co.uk/news/uk-politics-47652071
Motivation

https://www.bbc.co.uk/news/uk-politics-47031312

![Brexit: EU draft plans propose Brexit delay until May](https://www.bbc.co.uk/news/uk-politics-47031312)

© 21 March 2019

[https://www.bbc.co.uk/news/uk-politics-47031312](https://www.bbc.co.uk/news/uk-politics-47031312)
Motivation

Trump: I didn't get a thank you for McCain funeral


Vietnam War

http://www.bbc.co.uk/worldservice/vietnam-war
Motivation

'Cancel Brexit' petition passes 1m signatures on Parliament site

https://www.bbc.com

Motivation

Trump spooks markets with China trade tariffs warning

20 March 2019

Sign this petition

https://www.bbc.com

47651013?intlink_from_inlink
Motivation

https://www.bbc.co.uk/news/uk-politics-47031312
https://www.bbc.co.uk/news/world-us-canada-47477727
https://www.bbc.co.uk/news/uk-politics-46393399
https://www.bbc.co.uk/news/world-us-canada-47633940
https://www.bbc.co.uk/news/uk-politics-47627744
https://www.bbc.co.uk/news/uk-politics-parliaments-47653160
https://www.bbc.co.uk/news/world-us-canada-47642335
https://www.bbc.co.uk/news/uk-politics-47660019
https://www.bbc.co.uk/news/world-middle-east-47657843
https://www.bbc.co.uk/news/uk-politics-47659410
https://www.bbc.co.uk/news/uk-politics-47652071
Motivation

https://www.bbc.co.uk/news/uk-politics-47031312
https://www.bbc.co.uk/news/world-us-canada-47477727
https://www.bbc.co.uk/news/uk-politics-46393399
https://www.bbc.co.uk/news/world-us-canada-47633940
https://www.bbc.co.uk/news/uk-politics-47627744
https://www.bbc.co.uk/news/uk-politics-parliaments-47653160
https://www.bbc.co.uk/news/world-us-canada-47642335
https://www.bbc.co.uk/news/uk-politics-47660019
https://www.bbc.co.uk/news/world-middle-east-47657843
https://www.bbc.co.uk/news/uk-politics-47659410
https://www.bbc.co.uk/news/uk-politics-47652071

Can we automatically categorise based on page content?
### Word counts

<table>
<thead>
<tr>
<th></th>
<th>link 1</th>
<th>link 2</th>
<th>link 3</th>
<th>link 4</th>
<th>link 5</th>
<th>link 6</th>
<th>link 7</th>
<th>link 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>aberconwy</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>aberdeenshire</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>abide</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>able</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>above</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>absolutely</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>abstain</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>abuse</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>accorded</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- Complete matrix of word counts is $1800 \times 15$
## Data matrix

<table>
<thead>
<tr>
<th></th>
<th>Alex</th>
<th>Bobby</th>
<th>Charlie</th>
<th>Dana</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>58.1</td>
<td>19.5</td>
<td>66.3</td>
<td>48.3</td>
</tr>
<tr>
<td>Math(s)</td>
<td>84.2</td>
<td>85.1</td>
<td>90.7</td>
<td>81.4</td>
</tr>
<tr>
<td>History</td>
<td>96.1</td>
<td>61.2</td>
<td>85.6</td>
<td>52.1</td>
</tr>
<tr>
<td>Geography</td>
<td>45.3</td>
<td>97.8</td>
<td>75.0</td>
<td>81.2</td>
</tr>
</tbody>
</table>
### Data matrix

<table>
<thead>
<tr>
<th></th>
<th>58.1</th>
<th>19.5</th>
<th>66.3</th>
<th>48.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>84.2</td>
<td>85.1</td>
<td>90.7</td>
<td>81.4</td>
<td></td>
</tr>
<tr>
<td>96.1</td>
<td>61.2</td>
<td>85.6</td>
<td>52.1</td>
<td></td>
</tr>
<tr>
<td>45.3</td>
<td>97.8</td>
<td>75.0</td>
<td>81.2</td>
<td></td>
</tr>
</tbody>
</table>

- **columns = observations**
- **rows = variables**
## Observations and variables

<table>
<thead>
<tr>
<th>Observations</th>
<th>Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>People</td>
<td>Exam grades</td>
</tr>
<tr>
<td></td>
<td>Medical test scores</td>
</tr>
<tr>
<td>Companies</td>
<td>Share prices</td>
</tr>
<tr>
<td>Pixels/voxels</td>
<td>Concentration of chemicals</td>
</tr>
<tr>
<td></td>
<td>Spectroscopy data</td>
</tr>
<tr>
<td></td>
<td>Light intensities</td>
</tr>
<tr>
<td>News articles</td>
<td>Word frequencies</td>
</tr>
<tr>
<td>Plants or animals</td>
<td>Anatomical lengths</td>
</tr>
</tbody>
</table>
Characteristics of data matrices

- Real world example from seismic tomography:
  - **Big**: 87,616 x 67,320 matrix
  - **Sparse**: 13,734,559 non-zeros (0.23%)
  - **Low rank**: 26,137
Characteristics of data matrices

- Real world example from seismic tomography:
  - **Big**: 87,616 x 67,320 matrix
  - **Sparse**: 13,734,559 non-zeros (0.23%)
  - **Low rank**: 26,137

*Can we encapsulate data using a smaller matrix?*
Characteristics of data matrices

- Real world example from seismic tomography:
  - Big: 87,616 x 67,320 matrix
  - Sparse: 13,734,559 non-zeros (0.23%)
  - Low rank: 26,137

  Can we encapsulate data using a smaller matrix?

- Reduce number of variables:
  - Principal component analysis
  - Linear discriminant analysis
  - Various nonlinear methods
Characteristics of data matrices

- Real world example from seismic tomography:
  - Big: 87,616 x 67,320 matrix
  - Sparse: 13,734,559 non-zeros (0.23%)
  - Low rank: 26,137

  *Can we encapsulate data using a smaller matrix?*

- Reduce number of variables:
  - Principal component analysis
  - Linear discriminant analysis
  - Various nonlinear methods

  *But our data is non-negative!*
Non-negative matrix factorization

\[ m \times n \approx m \times k \times n \]
Non-negative matrix factorization

\[ A \approx WH \]

Data matrix \( A \) \( m \times n \)

All elements are non-negative

\( W \) \( m \times k \)

\( H \) \( k \times n \)

You choose \( k \ll \min(m, n) \)
Interpreting NMF – example

\[ A \approx W \times H \]

- \( A, m \times n \):
  - \( n \) Netflix users each rate \( m \) shows. Each column of \( A \) represents one user

- \( W, m \times k \):
  - Each column is a genre, or group of films

- \( H, k \times n \):
  - Each column represents how well the user fits into the different groupings
Interpreting NMF – in general

\[ A \approx WH \]

- **A**, data matrix:
  - \( m \) variables, \( n \) observations

- **W**, features matrix:
  - Each column is a feature, a combination of the variables

- **H**, coefficients matrix:
  - Each column represents how an observation is made up of features
Problem statement

- Exact NMF may not exist

- So try to solve: \[ \text{argmin}_{W \in \mathbb{R}^{m \times k}^+, H \in \mathbb{R}^{k \times n}^+} \| A - WH \|_F \]

- NMF is NP hard:
  - Seek acceptable rather than optimal solutions
  - Iterative algorithms, converging to local minima

- Not unique: e.g. \( WD \) and \( D^{-1}H \), if non-negative
NMF algorithms

- Exclusively two-block coordinate descent schemes
- Alternate between optimising for $W$ and $H$, keeping the other fixed:
  1. Find initial $W$ and $H$
  2. Solve $\min_{W \geq 0} \|A - WH\|_F$, $H$ fixed
  3. Solve $\min_{H \geq 0} \|A - WH\|_F$, $W$ fixed
  4. Repeat until some stopping criterion is met

- Stopping criterion:
  - Maximum number of iterations
  - Compute the gradient of $\|A - WH\|_F$ and compare with original gradient
NMF algorithms

- So how do we solve $\min_{W \geq 0} \|A - WH\|_F$?
  - **ALS**: solve $\min_{W} \|A - WH\|_F$ then project onto non-negative orthant
  - **ANLS**: Attempt to solve $\min_{W \geq 0} \|A - WH\|_F$ using e.g. active set methods, quasi-newton
  - **MU**: $W \leftarrow W \circ \begin{bmatrix} AH^T \\ WHH^T \end{bmatrix}$ ensures that objective function decreases
NMF algorithms

- So how do we solve $\min_{W \geq 0} \| A - WH \|_F$?
  - **ALS**: solve $\min_W \| A - WH \|_F$ then project onto non-negative orthant
    - May not converge
  - **ANLS**: Attempt to solve $\min_{W \geq 0} \| A - WH \|_F$ using e.g. active set methods, quasi-newton
    - Expensive iterations; difficult to implement
  - **MU**: $W \leftarrow W \cdot \frac{[AHT]}{[WHT]}$ ensures that objective function decreases
    - Slow convergence

- Or ...
Hierarchical alternating least squares

- Update one column of $W$ at a time with exact solution to a rank-1 version of the problem

$$W(:,j) \leftarrow \max \left( 0, \frac{AH(j,:)^T - \sum_{k \neq j} W(:,k)(H(k,:)H(j,:)^T)}{\|H(j,:)\|^2_2} \right)$$

- Guaranteed to converge to a local minimum
- Appears to have been rediscovered at least 6 times!

Non-negative matrix factorization of real non-negative matrix.

**Parameters:**
- **a:** float, array-like, shape \((m, n)\)
  - The \(m\) by \(n\) non-negative matrix \(A\).
- **w:** float, array-like, shape \((m, k)\)
  - if seed \(\leq 0\), \(w\) should be set to an initial iterate for the non-negative matrix factor, \(W\).
  - If seed \(\geq 1\), \(w\) need not be set. **real nmf** will generate a random initial iterate.
- **h:** float, array-like, shape \((k, n)\)
  - if seed \(\leq 0\), \(h\) should be set to an initial iterate for the non-negative matrix factor, \(H\).
  - If seed \(\geq 1\), \(h\) need not be set. **real nmf** will generate a random initial iterate.
- **seed:** int
  - if seed \(\leq 0\), the supplied values of \(W\) and \(H\) are used for the initial iterate.
  - If seed \(\geq 1\), the value of seed is used to seed a random number generator for the initial iterates \(W\) and \(H\). See [Generating Random Initial Iterates] for further details.
- **errtol:** float
  - The convergence tolerance for when the Hierarchical Alternating Least Squares iteration has reached a stationary point. If errtol \(\leq 0.0\) then \(\max(m, n) \times \sqrt{\text{machine precision}}\) is used.
- **maxit:** int
  - Specifies the maximum number of iterations to be used. If maxit \(\leq 0\), 200 is used.

**Returns:**
- **w:** float, ndarray, shape \((m, k)\)
  - The non-negative matrix factor, \(W\).
- **h:** float, ndarray, shape \((k, n)\)
  - The non-negative matrix factor, \(H\).

**Raises:**
- **NagValueError**
  - **errno 1**
    - On entry, \(m = \{\text{value}\}\).
    - Constraint: \(m \geq 2\).
  - **errno 2**
    - On entry, \(n = \{\text{value}\}\).
    - Constraint: \(n \geq 2\).
  - **errno 3**
    - On entry, \(k = \{\text{value}\}\), \(m = \{\text{value}\}\) and \(n = \{\text{value}\}\).
    - Constraint: \(1 \leq k < \min(m, n)\).
  - **errno 8**
    - An internal error occurred when generating initial values for \(w\) and \(h\). Please contact NAG.
  - **errno 9**
    - On entry, one of more of the elements of \(a\), \(w\) or \(h\) were negative.

**Warns:**
- **NagAlgorithmicWarning**
  - **errno 7**
    - The function has failed to converge after \(\{\text{value}\}\) iterations. The factorization given by \(w\) and \(h\) may still be a good enough approximation to be useful. Alternatively an improved factorization may be obtained by increasing maxit or using different initial choices of \(w\) and \(h\).
Non-negative matrix factorization of real non-negative matrix (reverse communication).

**Parameters:**

irevcm : int
On initial entry: must be set to 0

w : float, array-like, shape (m, k)
On initial entry:
- if seed \( \leq 0 \), w should be set to an initial iterate for the non-negative matrix factor, \( W \).
- If seed \( \geq 1 \), w need not be set. real_nmf_rcomm will generate a random initial iterate.

On intermediate entry: if irevcm = 3, w must contain \( AH^T \), where \( H^T \) is stored in \( ht \)

h : float, array-like, shape (k, n)
On initial entry:
- if seed \( \leq 0 \), h should be set to an initial iterate for the non-negative matrix factor, \( H \).
- If seed \( \geq 1 \), h need not be set. real_nmf_rcomm will generate a random initial iterate.

On intermediate entry: h must not be changed

ht : float, array-like, shape (n, k)
On initial entry: ht need not be set

On intermediate entry: if irevcm = 2, ht must contain \( A^TW \)

seed : int
On initial entry:
- if seed \( \leq 0 \), the supplied values of \( W \) and \( H \) are used for the initial iterate.
- If seed \( \geq 1 \), the value of seed is used to seed a random number generator for the initial iterates \( W \) and \( H \). See [Generating Random Initial Iterates](#) for further details.

**Returns:**

irevcm : int
On intermediate exit: specifies what action you must take before re-entering real_nmf_rcomm with irevcm unchanged

w : float, ndarray, shape (m, k)
On intermediate exit: if irevcm = 1 or 2, w contains the current iterate of the \( m \times k \) non-negative matrix \( W \)

On final exit: w contains the \( m \times k \) non-negative matrix \( W \)

h : float, ndarray, shape (k, n)
On intermediate exit: if irevcm = 1, h contains the current iterate of the \( k \times n \) non-negative matrix \( H \)

On final exit: h contains the \( k \times n \) non-negative matrix \( H \)

ht : float, ndarray, shape (n, k)
On intermediate exit: if irevcm = 3, ht contains the \( n \times k \) non-negative matrix \( H^T \), which is required in order to from \( AH^T \)

On final exit: ht is undefined

**Raises:**

NagValueError

(errno 1)

On intermediate re-entry, irevcm = \{value\}.

Constraint: irevcm = 1, 2 or 3.
### Word counts

<table>
<thead>
<tr>
<th></th>
<th>link 1</th>
<th>link 2</th>
<th>link 3</th>
<th>link 4</th>
<th>link 5</th>
<th>link 6</th>
<th>link 7</th>
<th>link 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>aberconwy</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>aberdeenshire</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>abide</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>able</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>above</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>absolutely</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>abstain</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>abuse</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>accorded</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- Complete data matrix is $1800 \times 15$
- Computed NMF using `real_nmf` with $k = 2$
Features matrix, $W$

- $1800 \times 2$, each column is a ‘category’
- Size of entries gives most important words in categories:

<table>
<thead>
<tr>
<th>Category 1</th>
<th>Category 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>deal</td>
<td>trump</td>
</tr>
<tr>
<td>brexit</td>
<td>president</td>
</tr>
<tr>
<td>extension</td>
<td>women</td>
</tr>
<tr>
<td>vote</td>
<td>nielsen</td>
</tr>
<tr>
<td>could</td>
<td>border</td>
</tr>
<tr>
<td>parliament</td>
<td>mccain</td>
</tr>
<tr>
<td>government</td>
<td>security</td>
</tr>
<tr>
<td>europe</td>
<td>senator</td>
</tr>
<tr>
<td>delay</td>
<td>class</td>
</tr>
<tr>
<td>referendum</td>
<td>administration</td>
</tr>
</tbody>
</table>
Features matrix, $W$

- **1800 \times 2**, each column is a ‘category’
- **Size of entries gives most important words in categories:**

<table>
<thead>
<tr>
<th>Category 1</th>
<th>Category 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>deal</td>
<td>trump</td>
</tr>
<tr>
<td>brexit</td>
<td>president</td>
</tr>
<tr>
<td>extension</td>
<td>women</td>
</tr>
<tr>
<td>vote</td>
<td>nielsen</td>
</tr>
<tr>
<td>could</td>
<td>border</td>
</tr>
<tr>
<td>parliament</td>
<td>mccain</td>
</tr>
<tr>
<td>government</td>
<td>security</td>
</tr>
<tr>
<td>europe</td>
<td>senator</td>
</tr>
<tr>
<td>delay</td>
<td>class</td>
</tr>
<tr>
<td>referendum</td>
<td>administration</td>
</tr>
</tbody>
</table>
Coefficients matrix, $H$

- Size $2 \times 15$
- Each row represents a category
- Each column represents a webpage:

<table>
<thead>
<tr>
<th>link 1</th>
<th>link 2</th>
<th>link 3</th>
<th>link 4</th>
<th>link 5</th>
<th>link 6</th>
<th>link 7</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>30.13</td>
<td>0.00</td>
<td>1.56</td>
<td>15.78</td>
<td>0.00</td>
<td>32.92</td>
<td>1.87</td>
<td>...</td>
</tr>
<tr>
<td>0.46</td>
<td>36.05</td>
<td>2.36</td>
<td>2.06</td>
<td>1.52</td>
<td>0.00</td>
<td>2.36</td>
<td>...</td>
</tr>
</tbody>
</table>

- Entries represent how well the page matches that category
- Used largest entry to assign category for each page
<table>
<thead>
<tr>
<th>Topic</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brexit delay: How can Article 50 be extended?</td>
<td></td>
</tr>
<tr>
<td>Kirstjen Nielsen: Walking a tightrope working for Trump</td>
<td></td>
</tr>
<tr>
<td>Trump homes plan at Menie being recommended for approval</td>
<td></td>
</tr>
<tr>
<td>Theresa May at her worst during Brexit speech - Mark Drakeford</td>
<td></td>
</tr>
<tr>
<td>President Trump shows map of 'IS defeat'</td>
<td></td>
</tr>
<tr>
<td>Brexit: What happens now?</td>
<td></td>
</tr>
<tr>
<td>Trump spooks markets with China trade tariffs warning</td>
<td></td>
</tr>
<tr>
<td>A tale of two Trumps: Jair Bolsonaro goes to Washington</td>
<td></td>
</tr>
<tr>
<td>Brexit: Theresa May to formally ask for delay</td>
<td></td>
</tr>
<tr>
<td>Corbyn calls for compromise to avoid no-deal Brexit</td>
<td></td>
</tr>
<tr>
<td>Trump: I didn't get a thank you for McCain funeral</td>
<td></td>
</tr>
<tr>
<td>Brexit: EU draft plans propose Brexit delay until May</td>
<td></td>
</tr>
<tr>
<td>Trump: Time to recognise Golan Heights as Israeli territory</td>
<td></td>
</tr>
<tr>
<td>Brexit: MPs urged not to travel home alone as tensions rise</td>
<td></td>
</tr>
<tr>
<td>'Cancel Brexit' petition passes 1m signatures on Parliament site</td>
<td></td>
</tr>
<tr>
<td>Brexit delay: How can Article 50 be extended?</td>
<td></td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Kirstjen Nielsen: Walking a tightrope working for Trump</td>
<td></td>
</tr>
<tr>
<td>Trump homes plan at Menie being recommended for approval</td>
<td></td>
</tr>
<tr>
<td>Theresa May at her worst during Brexit speech - Mark Drakeford</td>
<td></td>
</tr>
<tr>
<td>President Trump shows map of 'IS defeat'</td>
<td></td>
</tr>
<tr>
<td>Brexit: What happens now?</td>
<td></td>
</tr>
<tr>
<td>Trump spooks markets with China trade tariffs warning</td>
<td></td>
</tr>
<tr>
<td>A tale of two Trumps: Jair Bolsonaro goes to Washington</td>
<td></td>
</tr>
<tr>
<td>Brexit: Theresa May to formally ask for delay</td>
<td></td>
</tr>
<tr>
<td>Corbyn calls for compromise to avoid no-deal Brexit</td>
<td></td>
</tr>
<tr>
<td>Trump: I didn't get a thank you for McCain funeral</td>
<td></td>
</tr>
<tr>
<td>Brexit: EU draft plans propose Brexit delay until May</td>
<td></td>
</tr>
<tr>
<td>Trump: Time to recognise Golan Heights as Israeli territory</td>
<td></td>
</tr>
<tr>
<td>Brexit: MPs urged not to travel home alone as tensions rise</td>
<td></td>
</tr>
<tr>
<td>'Cancel Brexit' petition passes 1m signatures on Parliament site</td>
<td></td>
</tr>
</tbody>
</table>
Further applications

- Astronomy - searching for exoplanets
- Text mining - splitting documents into topics
- Speech de-noising
- Image processing - face recognition
- Bioinformatics - finding patterns of mutations in cancers
- Recommender systems
Applying NMF to financial time series data

<table>
<thead>
<tr>
<th></th>
<th>BTG</th>
<th>Dechra</th>
<th>Derwent London</th>
<th>GlaxoSmithKline</th>
<th>Genus</th>
<th>Grainger</th>
<th>Helical Bar</th>
<th>Segro</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nov 23, 2018</td>
<td>826.0</td>
<td>2052.0</td>
<td>3044.0</td>
<td>1582.0</td>
<td>2300.0</td>
<td>234.9</td>
<td>325.0</td>
<td>608.0</td>
</tr>
<tr>
<td>Nov 22, 2018</td>
<td>828.0</td>
<td>2076.0</td>
<td>3022.0</td>
<td>1565.0</td>
<td>2232.0</td>
<td>234.0</td>
<td>313.0</td>
<td>600.0</td>
</tr>
<tr>
<td>Nov 21, 2018</td>
<td>828.0</td>
<td>2052.0</td>
<td>3010.0</td>
<td>1576.2</td>
<td>2208.0</td>
<td>235.4</td>
<td>318.5</td>
<td>605.4</td>
</tr>
<tr>
<td>Nov 20, 2018</td>
<td>824.5</td>
<td>2014.0</td>
<td>2951.0</td>
<td>1582.2</td>
<td>2166.0</td>
<td>234.5</td>
<td>320.0</td>
<td>601.8</td>
</tr>
<tr>
<td>Nov 19, 2018</td>
<td>615.0</td>
<td>2052.0</td>
<td>2969.0</td>
<td>1568.6</td>
<td>2124.0</td>
<td>239.2</td>
<td>318.5</td>
<td>612.4</td>
</tr>
<tr>
<td>Nov 16, 2018</td>
<td>617.5</td>
<td>2084.0</td>
<td>3018.0</td>
<td>1562.0</td>
<td>2140.0</td>
<td>240.0</td>
<td>315.0</td>
<td>614.0</td>
</tr>
<tr>
<td>Nov 15, 2018</td>
<td>629.0</td>
<td>2112.0</td>
<td>2988.0</td>
<td>1562.6</td>
<td>2136.0</td>
<td>247.0</td>
<td>324.0</td>
<td>615.0</td>
</tr>
<tr>
<td>Nov 14, 2018</td>
<td>658.5</td>
<td>2174.0</td>
<td>3097.0</td>
<td>1551.6</td>
<td>2152.0</td>
<td>251.6</td>
<td>319.0</td>
<td>634.4</td>
</tr>
<tr>
<td>Nov 13, 2018</td>
<td>660.0</td>
<td>2156.0</td>
<td>3098.0</td>
<td>1579.4</td>
<td>2150.0</td>
<td>263.5</td>
<td>312.0</td>
<td>632.4</td>
</tr>
<tr>
<td>Nov 12, 2018</td>
<td>596.0</td>
<td>2136.0</td>
<td>3034.0</td>
<td>1573.0</td>
<td>2150.0</td>
<td>260.4</td>
<td>319.5</td>
<td>630.2</td>
</tr>
</tbody>
</table>

- Complete data matrix is $4593 \times 8$
- Computed NMF using `real_nmf` with $k = 2$
Applying NMF to financial time series data

- NMF does not necessarily cluster into traditional sectors
- Why might this be useful?
  - Group assets that show similar responses to economic factors
  - Choose stocks to ensure a balanced portfolio and minimise risk
  - See: Drakakis et al. 2008, *Analysis of financial data using non-negative matrix factorization*
Further financial applications

- Bankruptcy prediction:
  - Obtain variables (e.g. solvency or liquidity ratios) that best predict financial distress

- Term structure of interest rates:
  - Takada & Stern (2015), *Non-negative matrix factorization and term structure of interest rates*

- Natural language processing:
  - Analysis of text from news organisations to guide trading
Summary

- Useful properties of NMF:
  - Easy to interpret
  - Preserves non-negativity
  - Reduces dimensionality of data

- Extensions:
  - Constrained NMF, sparse NMF, minimisation with respect to different norms

How useful is NMF in finance, outside of academia?

What other functionality would make it more useful?