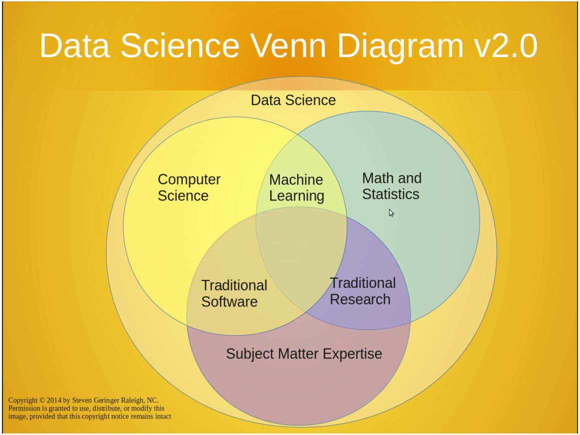
Non-Negative Matrix Factorization

Schwartz

August 14, 2017

How do you make a data scientist?



	Numerical Literacy	Scripting Coding	Inference Estimation	Predictive Modeling	Business Acumen	Creative Thinking	Problem Solving	? ?
Mathematics Degree								
Statistics Degree								
Economics Degree								
Computer Science Degree								
Data Science Immersive								
Independent Self Study								
Workshops and Lectures								
Community Engagement								
?								

Objectives

- NMF
  - versus SVD
  - non-negative
  - parts-based model
- Uses
  - learn/interpret latent reduced dimensionality features driving data
  - soft cluster samples by latent features
- Estimating NMF
  - gradient descent
  - alternating least squares (ALS)
  - multiplicative updating

What is NMF?

Singular Value Decomposition (SVD):

$$\underset{n\times p}{X} = \underset{n\times n}{U}\ \underset{n\times p}{\Sigma}\ \underset{p\times p}{V}^{T}$$
$$\approx \underset{n\times {\color{red}k}}{U}\ \underset{{\color{red}k\times k}}{\Sigma}\ \underset{{\color{red}k}\times p}{V}^{T}$$

What is NMF?

Singular Value Decomposition (SVD):

$$\underset{n\times p}{X} = \underset{n\times n}{U}\ \underset{n\times p}{\Sigma}\ \underset{p\times p}{V}^T$$
$$\approx \underset{n\times k}{U}\ \underset{k\times k}{\Sigma}\ \underset{k\times p}{V}^T$$

Non-Negative Matrix Factorization (NMF):

$$\underset{n\times p}{X} \approx \underset{n\times k}{W}\ \underset{k\times p}{H}$$

$$X_{ij}, W_{i'j'}, H_{i^*j^*} \geq 0$$

What is NMF?

Singular Value Decomposition (SVD):

$$\underset{n\times p}{X} = \underset{n\times n}{U} \; \underset{n\times p}{\Sigma} \; \underset{p\times p}{V}^{T}$$
$$\approx \underset{n\times k}{U} \; \underset{k\times k}{\Sigma} \; \underset{k\times p}{V}^{T}$$

Non-Negative Matrix Factorization (NMF):

$$\underset{n\times p}{X} \approx \underset{n\times k}{W} \; \underset{k\times p}{H}$$

$$X_{ij}, W_{i'j'}, H_{i^*j^*} \geq 0$$

So NMF is just SVD
– just drop the middle matrix and keep all the numbers positive

≥ 0

Keep all the numbers positive why?

Scores	item 1	item 2	$\cdots\cdots$	item p
user 1				
user 2				
$\vdots$				
user n				

$$\underset{n \times p}{X} = \underset{n \times k}{W} \; \underset{k \times p}{H}$$

≥ 0

Keep all the numbers positive why?

Scores	item 1	item 2	$\cdots\cdots$	item p
user 1				
user 2				
$\vdots$				
user n				

$$\underset{n \times p}{X} = \underset{n \times k}{W} \; \underset{k \times p}{H}$$

What are some recommender systems you know about, and what kind of numbers are used in those ratings, typically?

≥ 0

Keep all the numbers positive why? Because we can

Scores	item 1	item 2	$\cdots\cdots$	item p
user 1				
user 2				
$\vdots$				
user n				

$$\underset{n\times p}{X} = \underset{n\times k}{W} \; \underset{k\times p}{H}$$

What are some recommender systems you know about, and what kind of numbers are used in those ratings, typically?

≥ 0

Keep all the numbers positive why? Because we can

Scores	item 1	item 2	$\cdots\cdots$	item p
user 1				
user 2				
$\vdots$				
user n				

$$\underset{n\times p}{X} = \underset{n\times k}{W} \; \underset{k\times p}{H}$$

What are some recommender systems you know about, and what kind of numbers are used in those ratings, typically?

What about $\underset{n\times k}{W}$ and $\underset{k\times p}{H}$?

≥ 0

Keep all the numbers positive why? Because we can

Scores	item 1	item 2	$\cdots\cdots$	item p
user 1				
user 2				
$\vdots$				
user n				

$$\underset{n\times p}{X} = \underset{n\times k}{W}\ \underset{k\times p}{H}$$

What are some recommender systems you know about, and what kind of numbers are used in those ratings, typically?

What about $\underset{n\times k}{W}$ and $\underset{k\times p}{H}$?

"parts-based model"

≥ 0: what is the NMF "parts based model"?

$$\underset{n \times p}{X} \approx \underset{n \times k}{W} \underset{k \times p}{H} \qquad \hat{X}_{ij} = \overset{1 \times k}{W_{i\cdot}} \overset{k \times 1}{H_{\cdot j}}$$

$$X_{ij}, W_{i'j'}, H_{i^*j^*} \geq 0$$

$\big|$ protein carbs fat fiber vitamins $\big|$ (individuals diet implications)

(individuals food intake)

$= \big|$ food 1 food 2 $\cdots$ food k $\big| \times$

	protein	carbs	fat	fiber	vitamins
food 1					
food 2			(food contributions)		
$\vdots$					
food k					

≥ 0: what is the NMF "parts based model"?

$$\underset{n\times p}{X} \approx \underset{n\times k}{W} \underset{k\times p}{H} \qquad \hat{X}_{ij} = \overset{1\times k}{W_{i\cdot}} \overset{k\times 1}{H_{\cdot j}}$$

$$X_{ij}, W_{i'j'}, H_{i^*j^*} \geq 0$$

| protein carbs fat fiber vitamins | (individuals diet implications)

(individuals food intake)

$= |$ food 1 food 2 $\cdots$ food k $| \times$

	protein	carbs	fat	fiber	vitamins
food 1					
food 2					
⋮			(food contributions)		
food k					

- every user i gets "amounts" of k factors (food): W's i^{th} row
- factors k may contribute to item j (feature): H's j^{th} column
- "agreement" in *factors* for *user i* and *item j* determines X_{ij}

≥ 0: what is the NMF "parts based model"?

$$\underset{n \times p}{X} \approx \underset{n \times k}{W} \underset{k \times p}{H} \qquad \hat{X}_{ij} = \overset{1 \times k}{W_{i\cdot}} \overset{k \times 1}{H_{\cdot j}}$$

$$X_{ij}, W_{i'j'}, H_{i^*j^*} \geq 0$$

| protein carbs fat fiber vitamins | (individuals diet implications)

(individuals food intake)

= | food 1 food 2 $\cdots$ food k | $\times$

	protein	carbs	fat	fiber	vitamins
food 1					
food 2			(food contributions)		
$\vdots$					
food k					

- every user i gets "amounts" of k factors (food): W's i^{th} row
- factors k may contribute to item j (feature): H's j^{th} column
- "agreement" in *factors* for *user* i and *item* j determines X_{ij}
- everything being positive provides this "parts-based model"

Example: NMF Topic Modeling for NLP

$$
\begin{array}{c|cccc}
 & \text{word 1} & \text{word 2} & \cdots\cdots & \text{word p} \\
\hline
\text{doc 1} & & & & \\
\text{doc 2} & & & & \\
\vdots & & & & \\
\text{doc n} & & & &
\end{array}
$$

$$
= \begin{array}{c|ccc}
 & \text{topic 1} & \cdots & \text{topic } k \\
\hline
\text{doc 1} & & & \\
\text{doc 2} & & & \\
\vdots & & & \\
\text{doc n} & & &
\end{array}
\times
\begin{array}{c|ccc}
 & \text{word 1} & \cdots & \text{word } p \\
\hline
\text{topic 1} & & & \\
\vdots & & & \\
\text{topic } k & & &
\end{array}
$$

- Identifies latent "topics" or features driving word appearance
- Says what words each of the topics are comprised of (cool!)
- Gives topic similarity (how?) for documents (soft clustering)

How good is this model?

$$\underset{n\times p}{X} \approx \underset{n\times k}{W} \; \underset{k\times p}{H}$$

How good is this model?

$$\underset{n\times p}{X} \approx \underset{n\times k}{W} \;\; \underset{k\times p}{H}$$

$$\sum_{i,j}(X_{ij} - \hat{X}_{ij})^2 = \sum_{i,j}(X_{ij} - W_{i.}\; H_{.j})^2$$

How good is this model?

$$\underset{n \times p}{X} \approx \underset{n \times k}{W} \quad \underset{k \times p}{H}$$

$$\sum_{i,j} (X_{ij} - \hat{X}_{ij})^2 = \sum_{i,j} (X_{ij} - W_{i.} \, H_{.j})^2$$

- What i and j do we use to evaluate this on?

Scores	item 1	item 2	$\cdots\cdots$	item p
user 1				
user 2				
$\vdots$				
user n				

How good is this model?

$$\underset{n\times p}{X} \approx \underset{n\times k}{W} \quad \underset{k\times p}{H}$$

$$\sum_{i,j}(X_{ij} - \hat{X}_{ij})^2 = \sum_{i,j}(X_{ij} - W_{i.}\ H_{.j})^2$$

- What i and j do we use to evaluate this on?

Scores	item 1	item 2	$\cdots\cdots$	item p
user 1				
user 2				
$\vdots$				
user n				

- What's interesting about this compared to SVD?

How good is this model?

$$\underset{n \times p}{X} \approx \underset{n \times k}{W} \;\; \underset{k \times p}{H}$$

$$\sum_{i,j} (X_{ij} - \hat{X}_{ij})^2 = \sum_{i,j} (X_{ij} - W_{i\cdot} \, H_{\cdot j})^2$$

- What i and j do we use to evaluate this on?

Scores	item 1	item 2	$\cdots\cdots$	item p
user 1				
user 2				
$\vdots$				
user n				

- What's interesting about this compared to SVD?

  Hint: what does SVD do with NA values?

How do we learn H and W?

How do we learn H and W?

- **Gradient Descent**

$$\frac{\partial}{\partial W_{i'k}} \sum_{i,j} (X_{ij} - W_{i\cdot} \; H_{\cdot j})^2 = \sum_{j} -2(X_{i'j} - W_{i'k} \; H_{kj}) H_{kj}$$

$$\frac{\partial}{\partial H_{kj'}} \sum_{i,j} (X_{ij} - W_{i\cdot} \; H_{\cdot j})^2 = \sum_{i} -2(X_{ij'} - W_{ik} \; H_{kj'}) W_{ik}$$

$$\text{subject to } W_{ij}, H_{i'j'} \geq 0$$

How do we learn H and W?

- **Gradient Descent**

$$\frac{\partial}{\partial W_{i'k}} \sum_{i,j} (X_{ij} - W_{i\cdot} \;\; H_{\cdot j})^2 = \sum_{j} -2(X_{i'j} - W_{i'k} \;\; H_{kj}) H_{kj}$$

$$\frac{\partial}{\partial H_{kj'}} \sum_{i,j} (X_{ij} - W_{i\cdot} \;\; H_{\cdot j})^2 = \sum_{i} -2(X_{ij'} - W_{ik} \;\; H_{kj'}) W_{ik}$$

$$\text{subject to } W_{ij}, H_{i'j'} \geq 0$$

- **Alternating Lease Squares (ALS)**

0. Initialize H and W with $H_{ij}, W_{i'j'} > 0$
1. Update $H_{\cdot j}$ using OLS: $X_{\cdot j} = \mathbf{W} H_{\cdot j} + \epsilon_{\cdot j}$

$$\begin{bmatrix} & | & \\ & | & \\ & | & \end{bmatrix} = \begin{bmatrix} \longrightarrow\longrightarrow \\ \longrightarrow\longrightarrow \\ \longrightarrow\longrightarrow \end{bmatrix} \begin{bmatrix} & \downarrow & \\ & \downarrow & \\ & \downarrow & \end{bmatrix}$$

2. Update $W_{i\cdot}$ using OLS: $X_{i\cdot}^T = \mathbf{H^T} W_{i\cdot}^T + \epsilon_{i\cdot}^T$

$$\begin{bmatrix} - & - \\ & \\ & \end{bmatrix}^T = \left(\begin{bmatrix} \longrightarrow\longrightarrow \\ \\ \\ \end{bmatrix} \begin{bmatrix} \downarrow & \downarrow & \downarrow & \downarrow \\ \downarrow & \downarrow & \downarrow & \downarrow \\ \downarrow & \downarrow & \downarrow & \downarrow \end{bmatrix} \right)^T$$

3. If $H_{ij} < 0$ set $H_{ij} = 0$; if $W_{i'j'} < 0$ set $W_{i'j'} = 0$
4. Evaluate stopping criterion: return to step 1 if check fails

   (What stopping criterion might we use?)

How do we learn H and W?

- **Lee and Seung's "multiplicative update rules"**
  0. Initialize H and W with $H_{ij}, W_{i'j'} > 0$
  1. Update W and H with

$$W'_{ik} = W_{ik} \frac{(XH^T)_{ik}}{(WHH^T)_{ik}} \qquad H'_{kj} = H_{kj} \frac{(W^T X)_{kj}}{(W^T WH)_{kj}}$$

  2. Evaluate stopping criterion: return to step 1 if check fails
     (What stopping criterion might we use?)

How do we learn H and W?

- **Lee and Seung's "multiplicative update rules"**
  0. Initialize H and W with $H_{ij}, W_{i'j'} > 0$
  1. Update W and H with

$$W'_{ik} = W_{ik} \frac{(XH^T)_{ik}}{(WHH^T)_{ik}} \qquad H'_{kj} = H_{kj} \frac{(W^T X)_{kj}}{(W^T WH)_{kj}}$$

  2. Evaluate stopping criterion: return to step 1 if check fails
     (What stopping criterion might we use?)

What is doing this?

- Recall the OLS estimate $\hat{\beta} = (X^T X)^{-1} X^T Y$
  Which looks a lot like, e.g., $\hat{H}_{kj} = \frac{(W^T X)_{kj}}{(W^T W)_{kj}}$
- So the update looks a lot like $H'_{kj} = H_{kj} \frac{\hat{H}_{kj}}{H}$
  so if the new estimate is increased/decreased relative to the current value then the estimate is increase/decrease by that proportion; but this change in H will result in a change in $\hat{W}$ the next time, which will again change $\hat{H}$ the *next* next time...

Wrap Up

Both do Unsupervised Dimensionality Reduction, but

	SVD/PCA	NMF
NA's	Nope	Yep
Estimation	Non-iterative	Iterative
Coefficients	Orthogonal, $+/-$ coefficients	only $+$ coefficients
Interpretation	Linear combination	"Parts-based"
k	Skree plot	Cross-validation
	All of those as well $\rightarrow$	Permutation testing
		Cophenetic correlation
		Interpretability

<u>Go look at all the other instructors jupyter notebooks demoing this all</u>