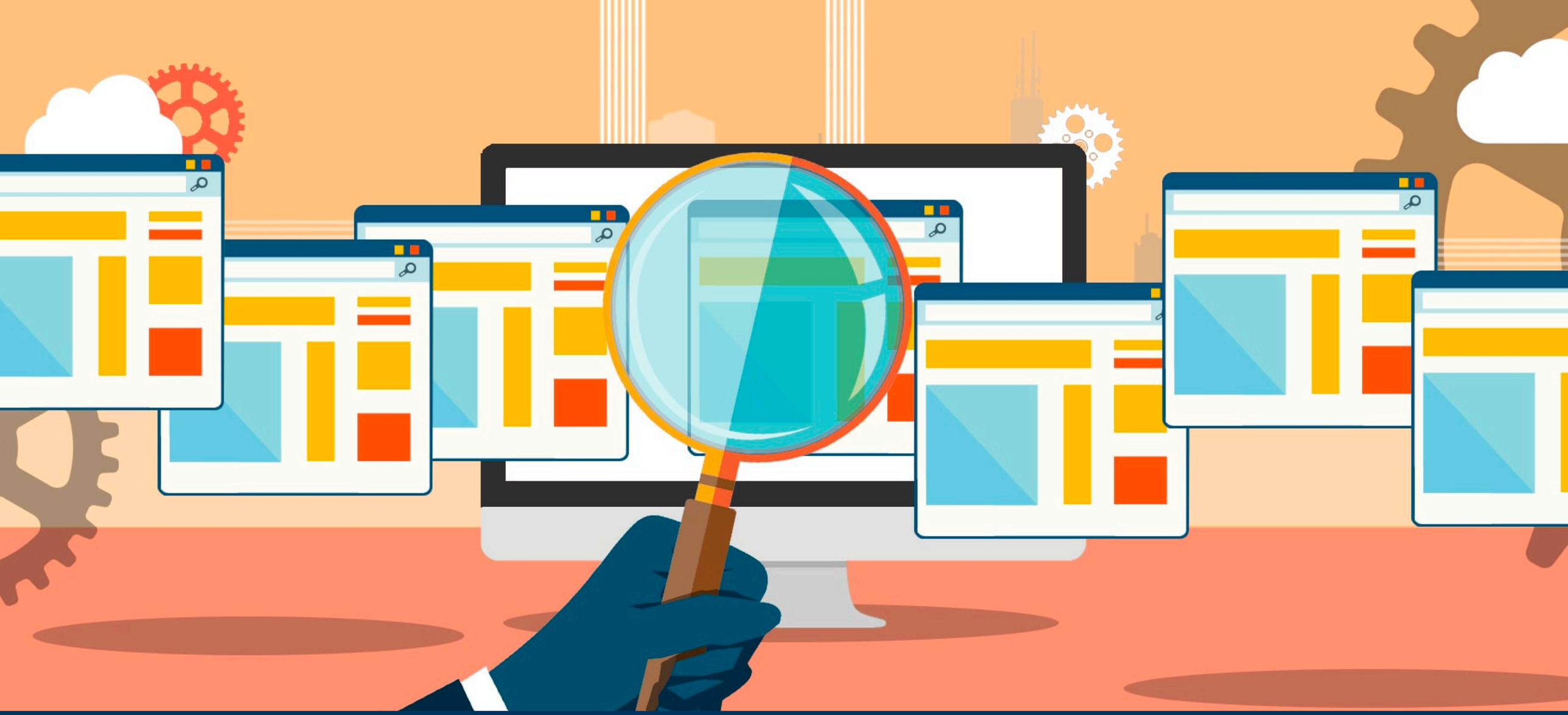


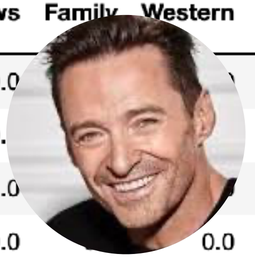
Latent Factors and Dimensionality Reduction

INST414 - Data Science Techniques

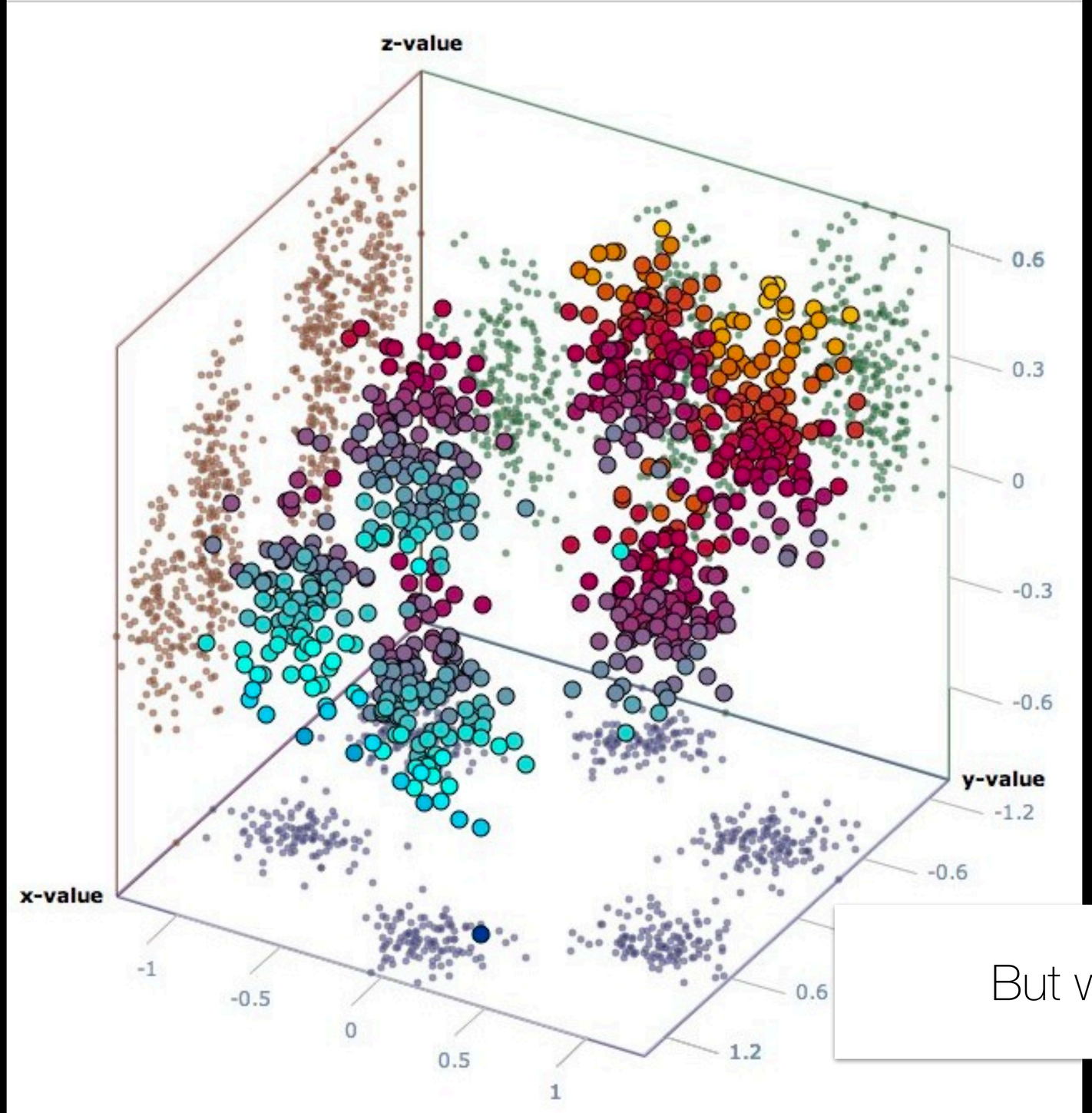

IMDb

```
actor_genre_df.head(20)
```

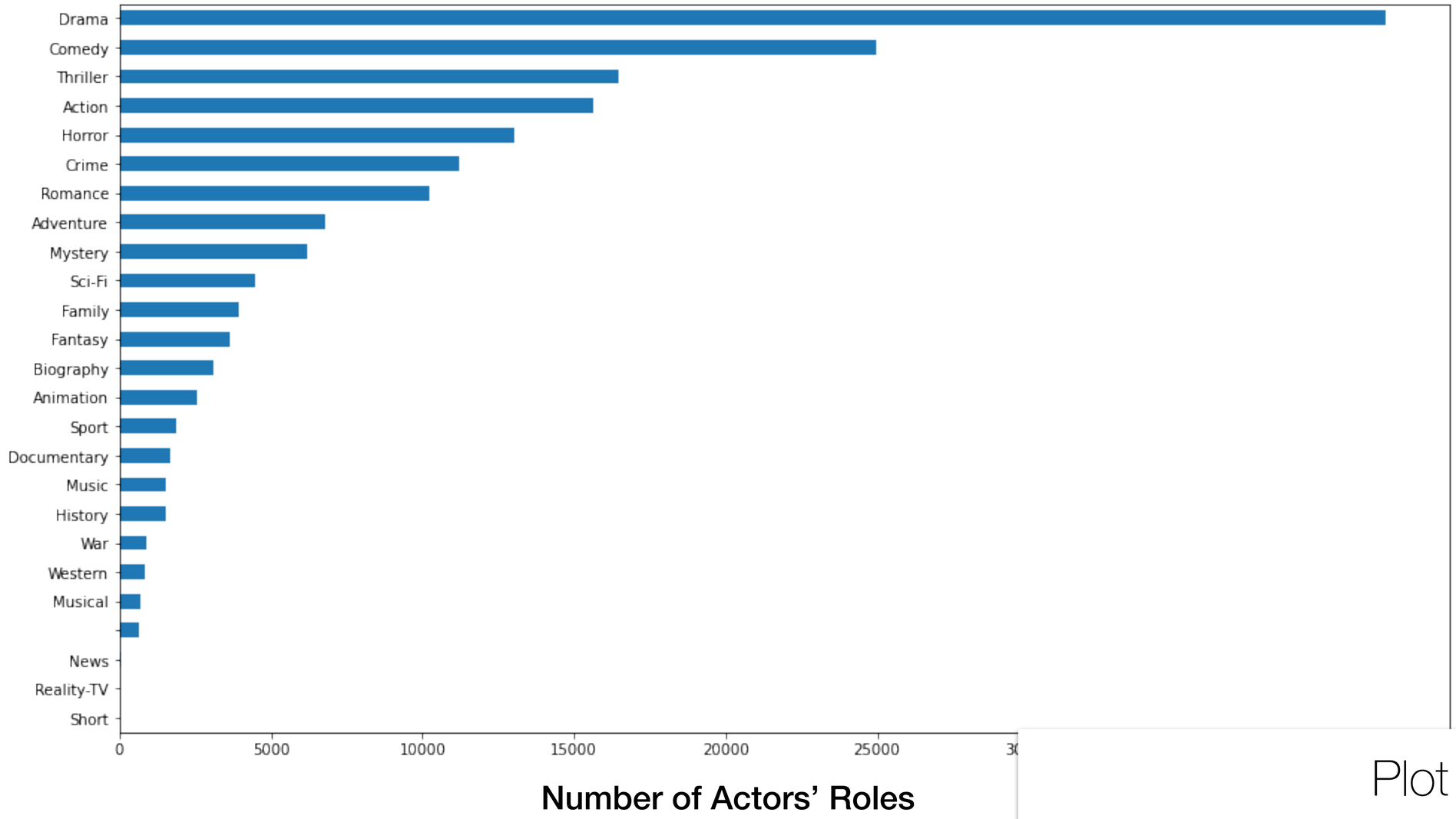
	Comedy	Fantasy	Romance	Action	Crime	Adventure	Mystery	Thriller	Drama	Biography	...	Sport		News	Family	Western	Short
nm0000212	16.0	3.0	16.0	5.0	4.0	2.0	5.0	3.0	16.0	2.0	...	0.0	0.0	0.0			0.0
nm0413168	8.0	3.0	6.0	14.0	6.0	11.0	5.0	2.0	13.0	5.0	...	0.0	0.0	0.			0.0
nm0000630	10.0	2.0	6.0	4.0	1.0	2.0	2.0	4.0	17.0	6.0	...	4.0	1.0	1.0			0.0
nm0005227	12.0	1.0	3.0	2.0	0.0	3.0	0.0	1.0	5.0	1.0	...	1.0	0.0	0.0		0.0	0.0
nm0697338	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
nm1300519	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
nm0940707	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
nm0625977	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
nm0792032	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
nm0496571	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
nm2868805	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
nm2866192	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
nm0001379	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	1.0	...	1.0	0.0	0.0	1.0	0.0	0.0
nm0462648	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	2.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
nm0000953	6.0	0.0	0.0	1.0	3.0	0.0	0.0	2.0	9.0	7.0	...	0.0	0.0	0.0	0.0	0.0	0.0
nm0001782	0.0	0.0	0.0	1.0	0.0	1.0	0.0	1.0	1.0	1.0	...	0.0	0.0	0.0	0.0	0.0	0.0
nm0005077	1.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	1.0	0.0	...	0.0	0.0	0.0	1.0	0.0	0.0
nm0550626	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0								
nm0177016	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0								
nm0907480	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0								



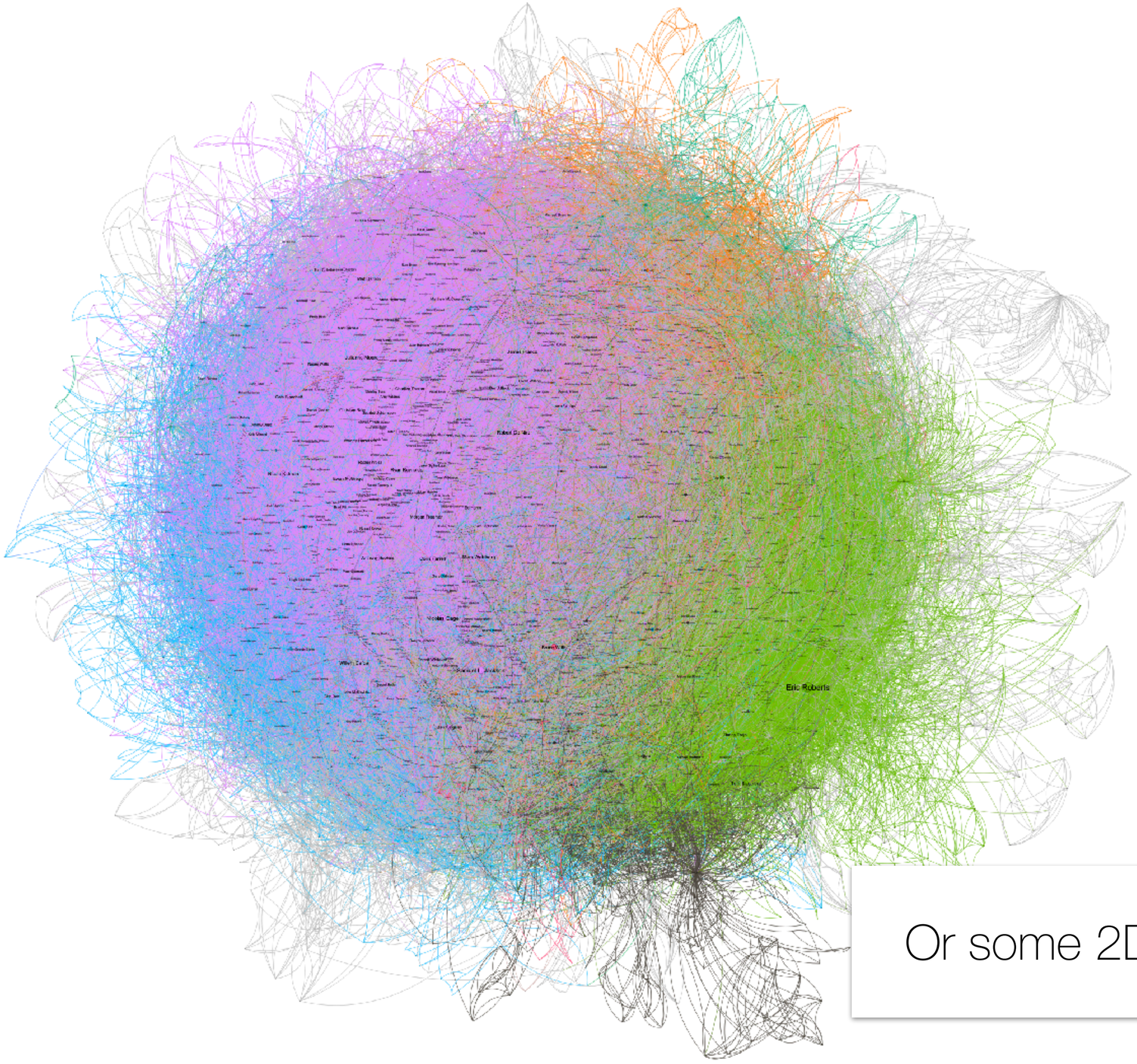
How might you visualize these rows?



But we have 25 genres…



Plot only the top two most prolific genres?



Or some 2D space of similarity

Today's Learning Objectives

Define the connection between latent factors and dimensionality reduction

Describe at least two methods for dimensionality reduction

Today's Learning Objectives

Define the connection between latent factors and dimensionality reduction

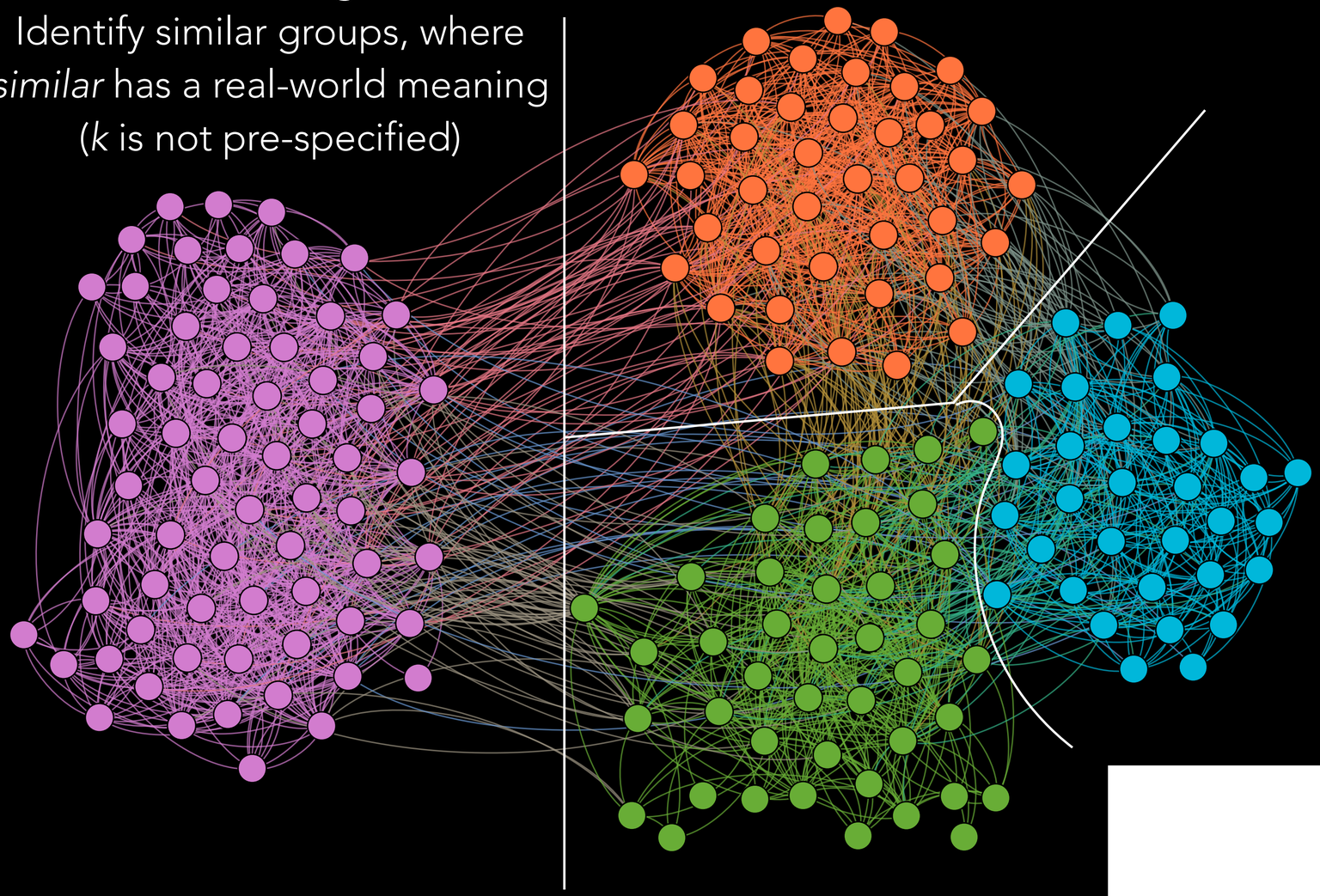
Describe at least two methods for dimensionality reduction

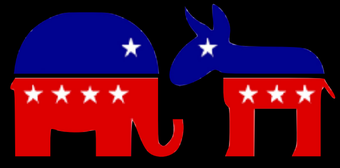
Core Motivation: We want to find similar content in low-dimensional spaces

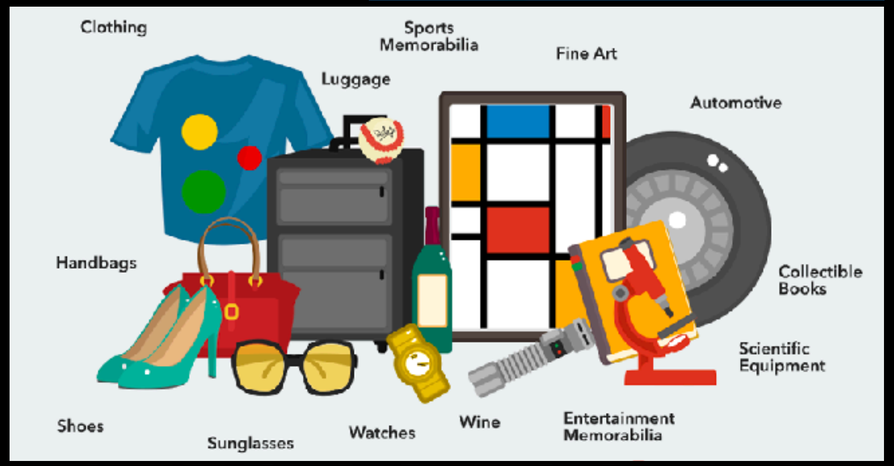
One solution is to group elements or dimensions

What do these “groups” mean?

Clustering:
Identify similar groups, where *similar* has a real-world meaning (*k* is not pre-specified)







Groups are “similar” things

Retweets

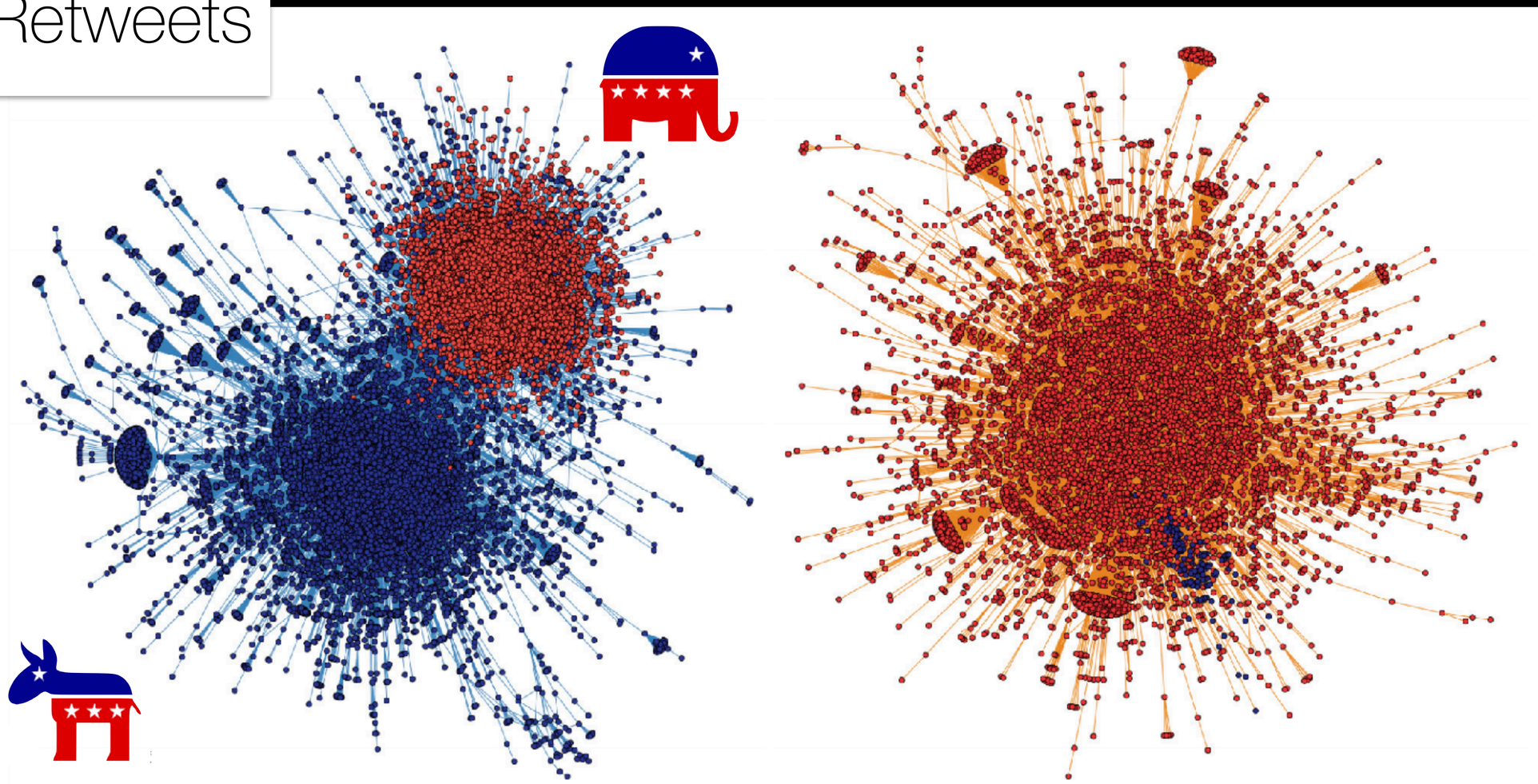
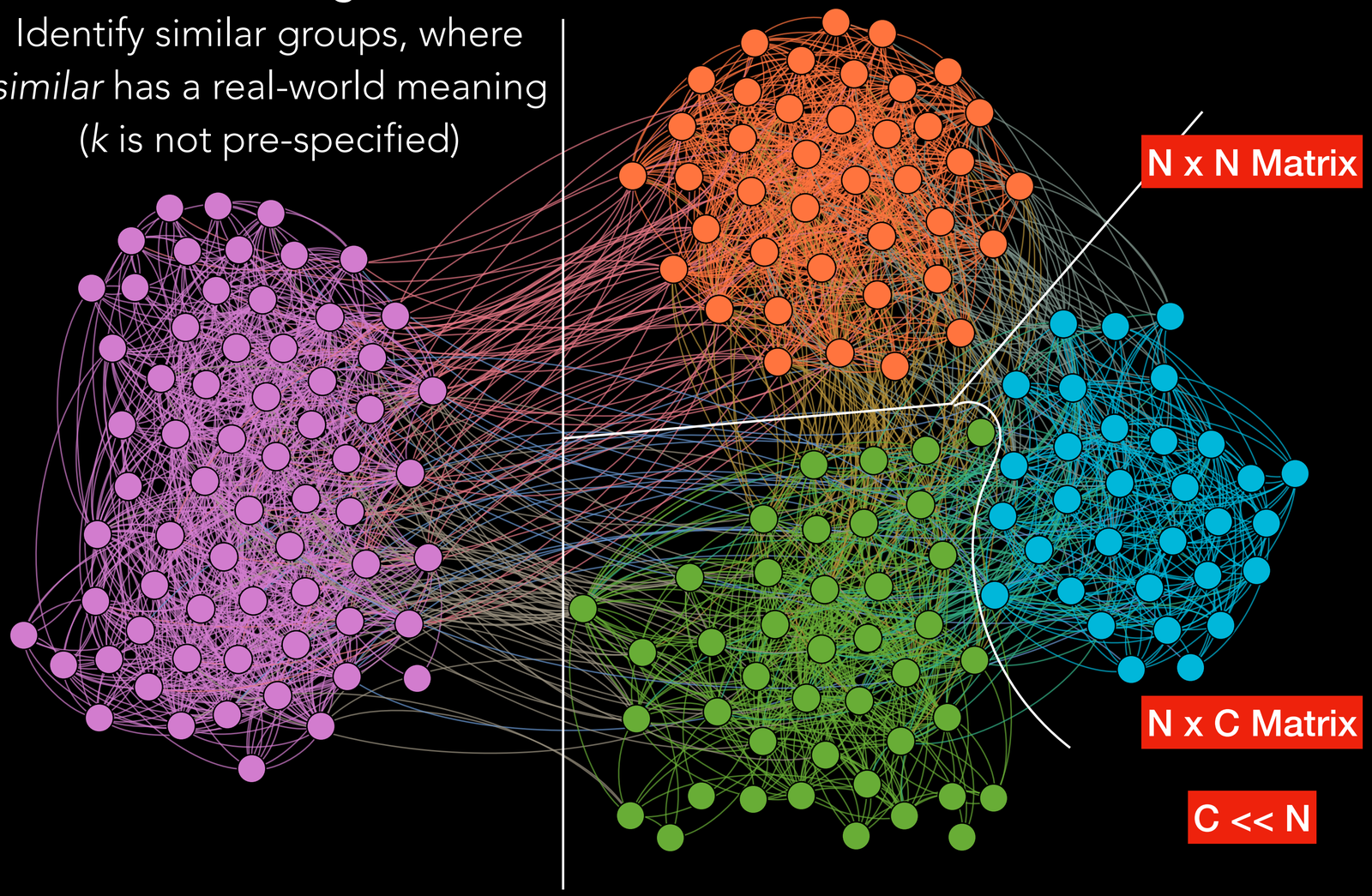


Figure 1: The political retweet (left) and mention (right) networks, laid out using a force-directed algorithm. Node colors reflect cluster assignments (see § 3.1). Community structure is evident in the retweet network, but less so in the mention network. We show in § 3.3 that in the retweet network, the red cluster A is made of 93% right-leaning users, while the blue cluster B is made of 80% left-leaning users.

Conover, Michael D., Jacob Ratkiewicz, Matthew R. Francisco, Bruno Gonçalves, Filippo Menczer, and Alessandro Flammini. "Political polarization on twitter." Icwsm 133, no. 26 (2011): 89-96.


Automotive
Animals
Software
JAGUAR
Mac OS X v10.2

Clustering:
Identify similar groups, where *similar* has a real-world meaning (*k* is not pre-specified)



Adjacency Matrix *A*

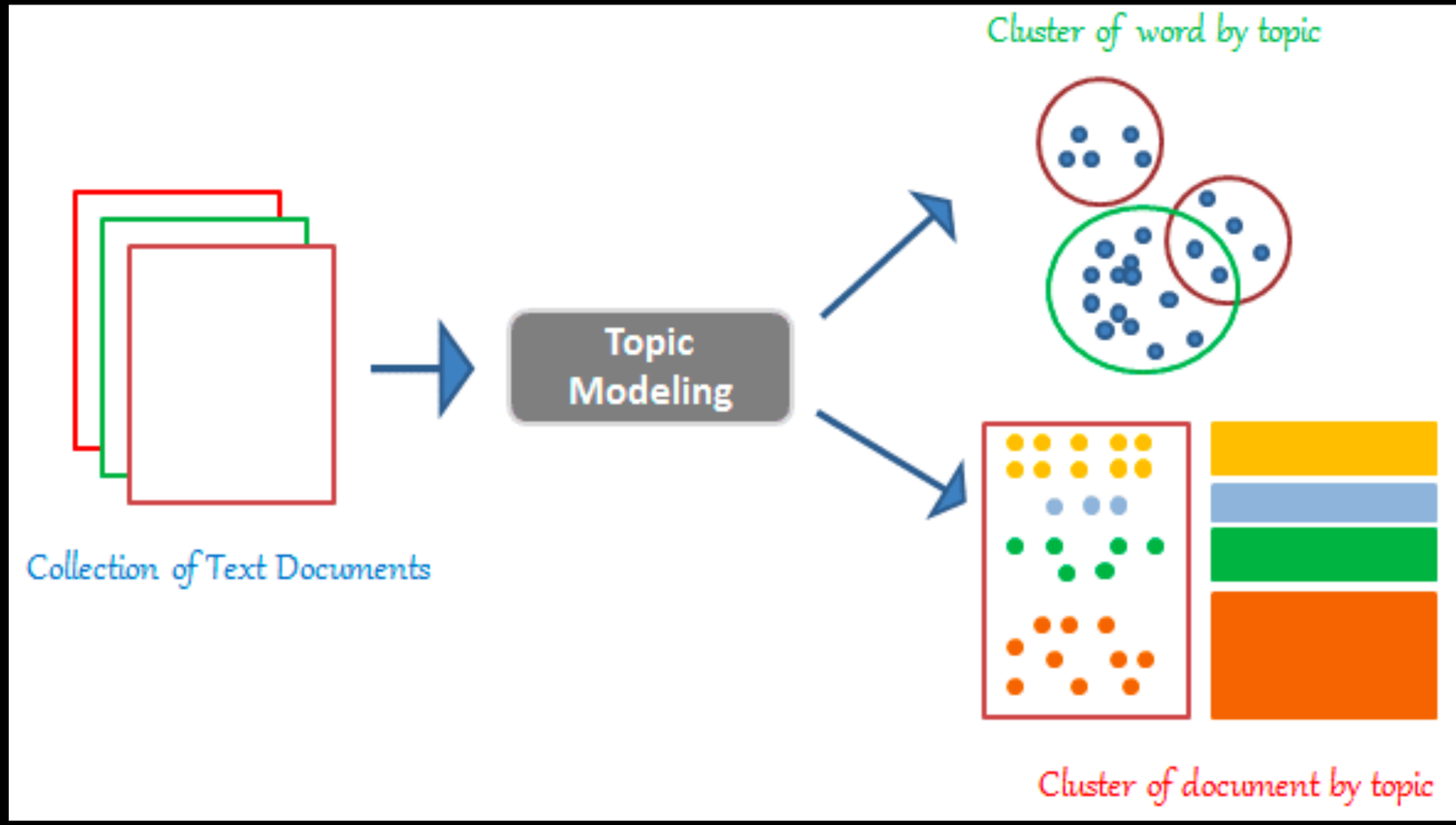
	v_1	v_2	v_3	…	v_n
v_1	0	0	1		1
v_2	1	0	0		0
v_3	0	1	0		0
…					
v_n	0	0	0		0

	C_1	C_2	C_3	C_4
v_1	0	0	1	0
v_2	1	0	0	0
v_3	0	1	0	0
…				
v_n	0	0	0	1

Bag-of-Words/Shingle Matrix

	W_1	W_2	W_3	…	W_n
D_1	0	0	1		1
D_2	1	0	0		0
D_3	0	1	0		0
…					
D_n	0	0	0		0

	T_1	T_2	T_3	T_4
D_1	0	0	1	0
D_2	1	0	0	0
D_3	0	1	0	0
…				
D_n	0	0	0	1



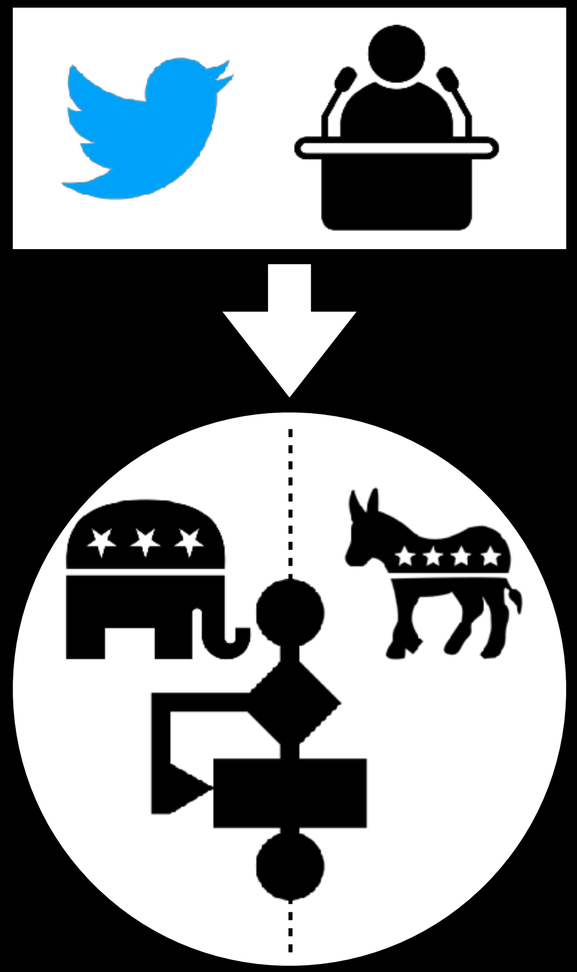
Topic modeling and Latent Dirichlet Allocation

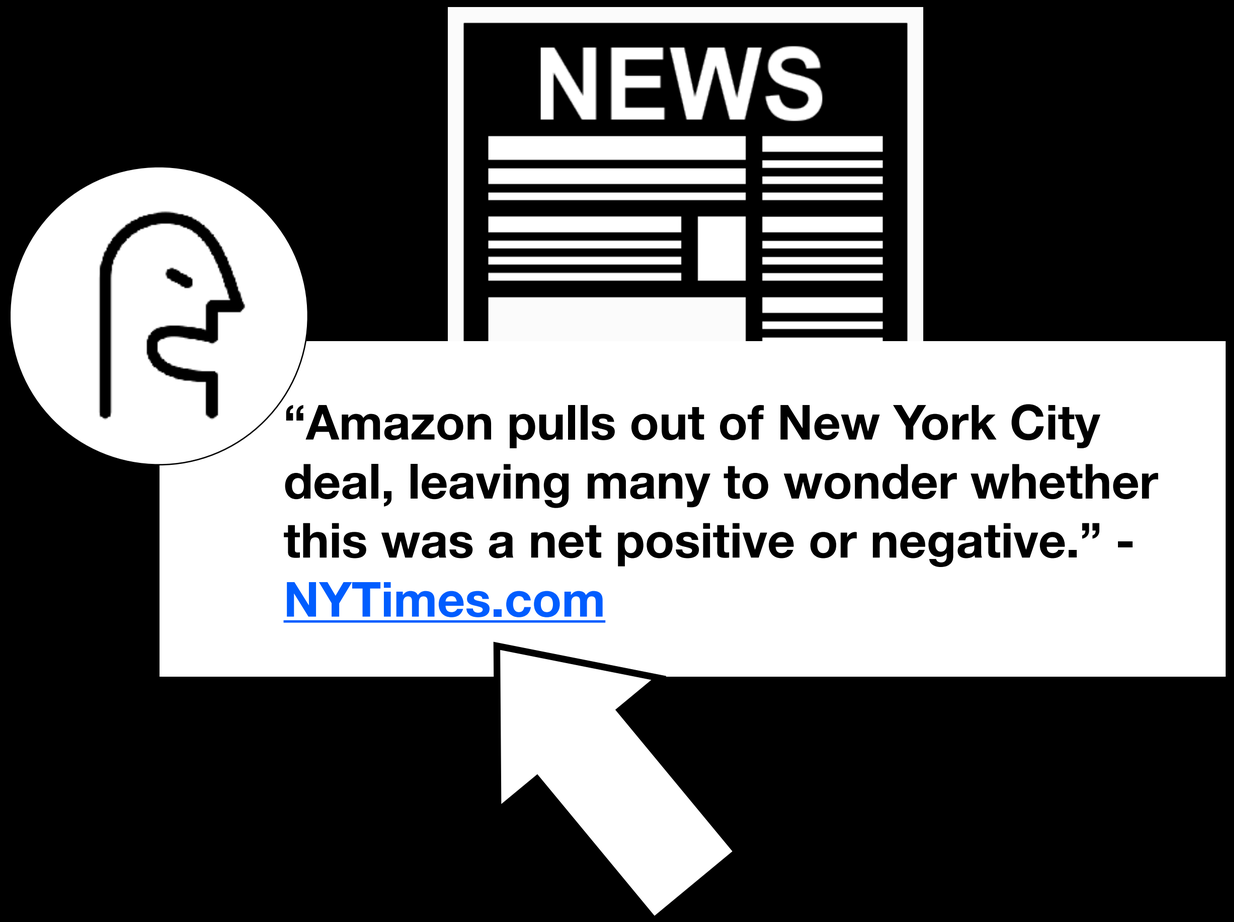
In some sense, you're familiar with dimensionality reduction already

These groups capture hidden ("latent") relationships

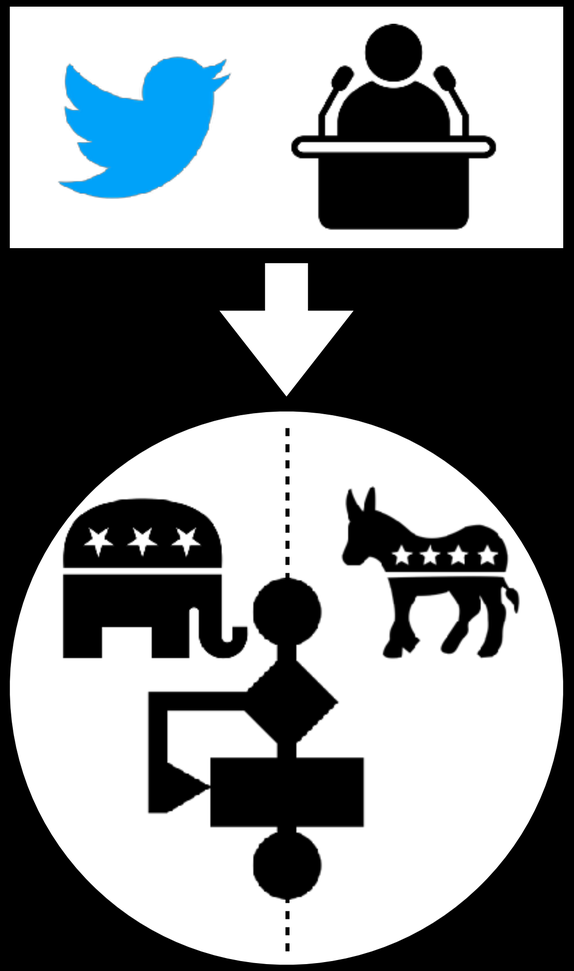
May not know what these latent factors are a priori

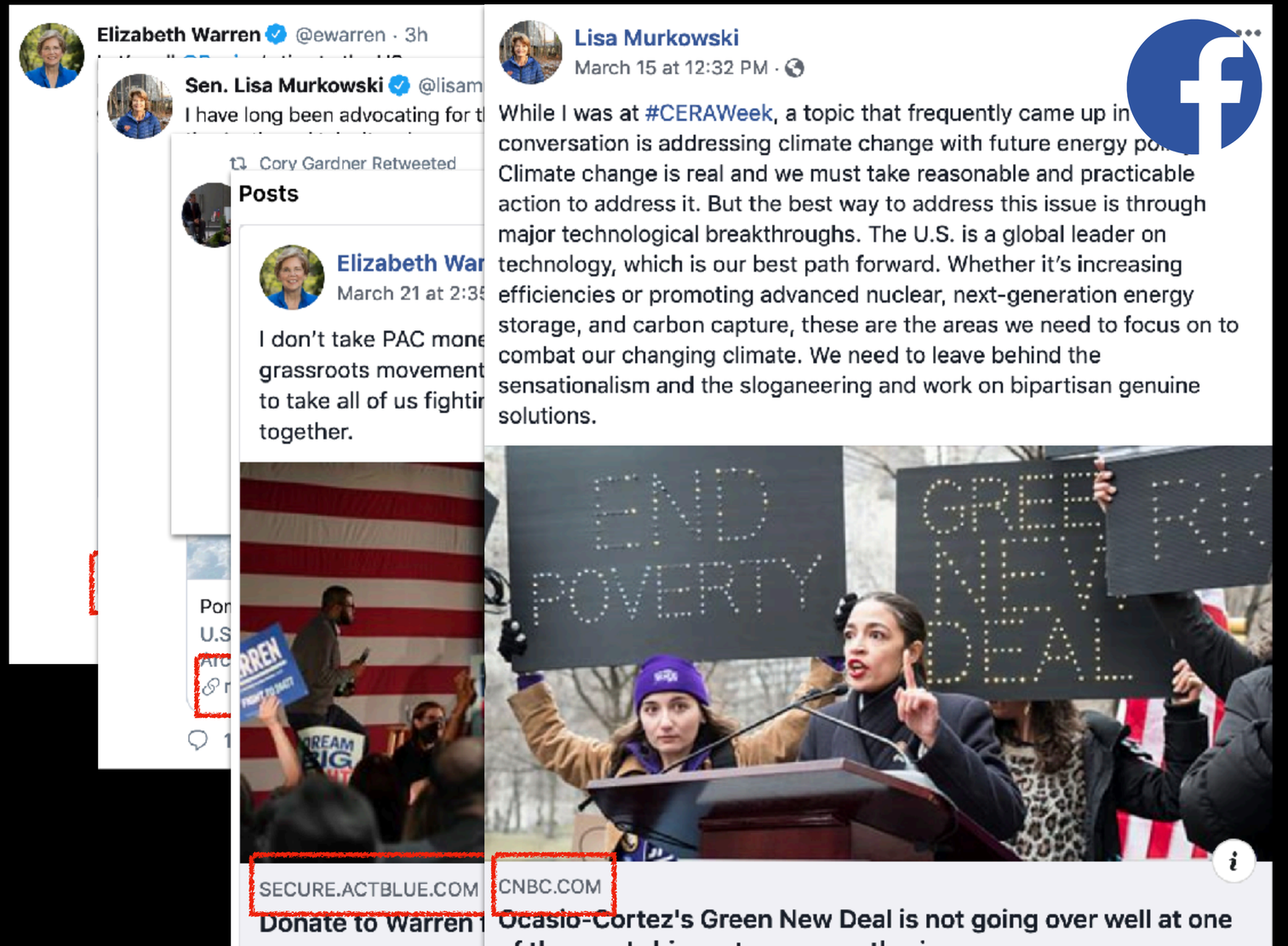
Link Sharing and Ideology





Link Sharing and Ideology





Link Sharing and Ideology

Elizabeth Warren

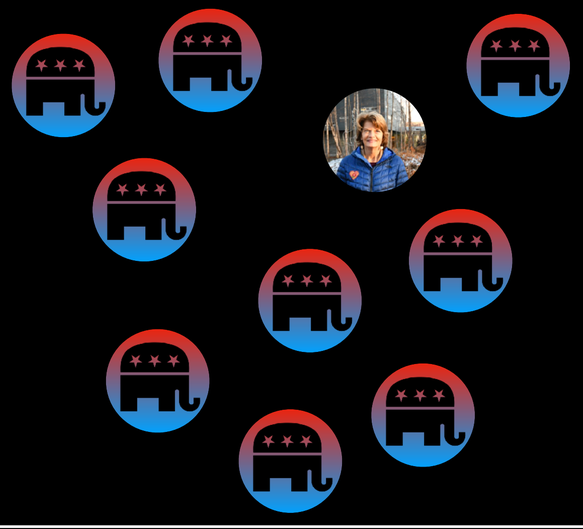
Lisa Murkowski

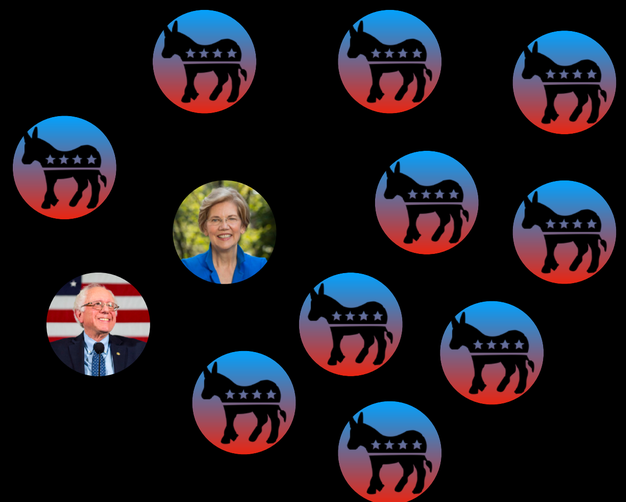
Bernie Sanders

nytimes.com	reuters.com	bbc.co.uk	vox.com	HuffPost.com	breitbart.com	...	foxnews.com	npr.org
23	7	9	37	22	0	...	0	20
3	11	10	0	1	5	...	21	3
31	23	17	40	39	0	...	0	17

Link Sharing and Ideology

One dimension for each web domain you share



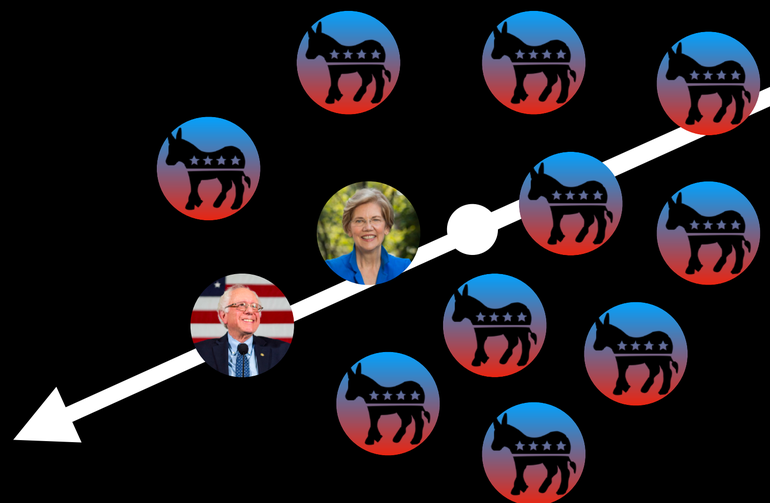


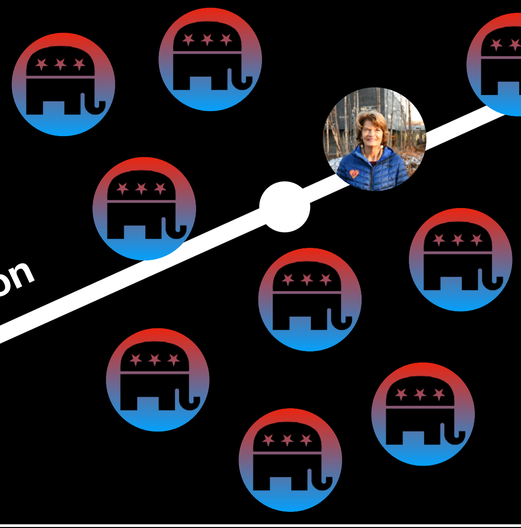
Point-clouds for various politicians' domain sharing

Link Sharing and Ideology

One dimension
for each web domain
you share

One-Dimensional Projection

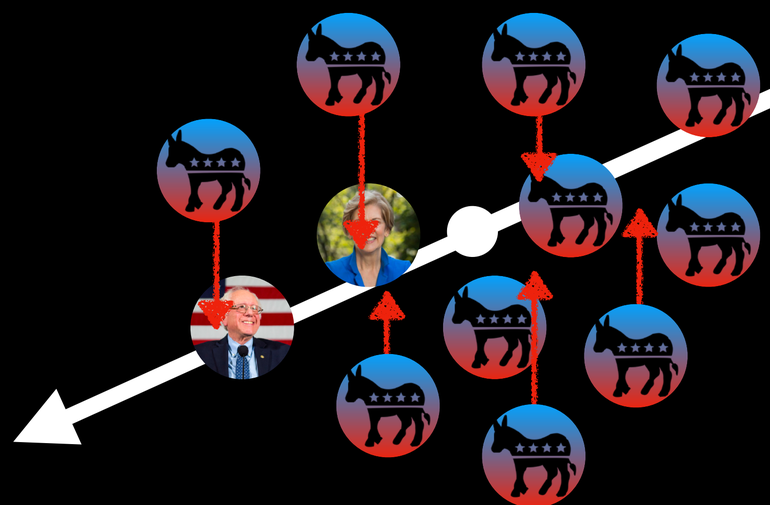


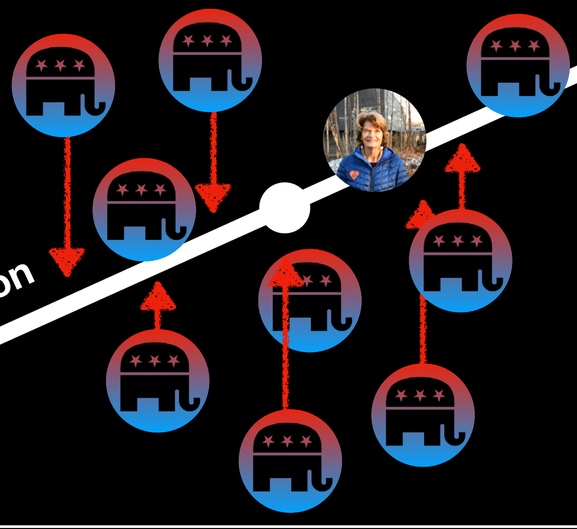


Link Sharing and Ideology

One dimension
for each web domain
you share

One-Dimensional Projection

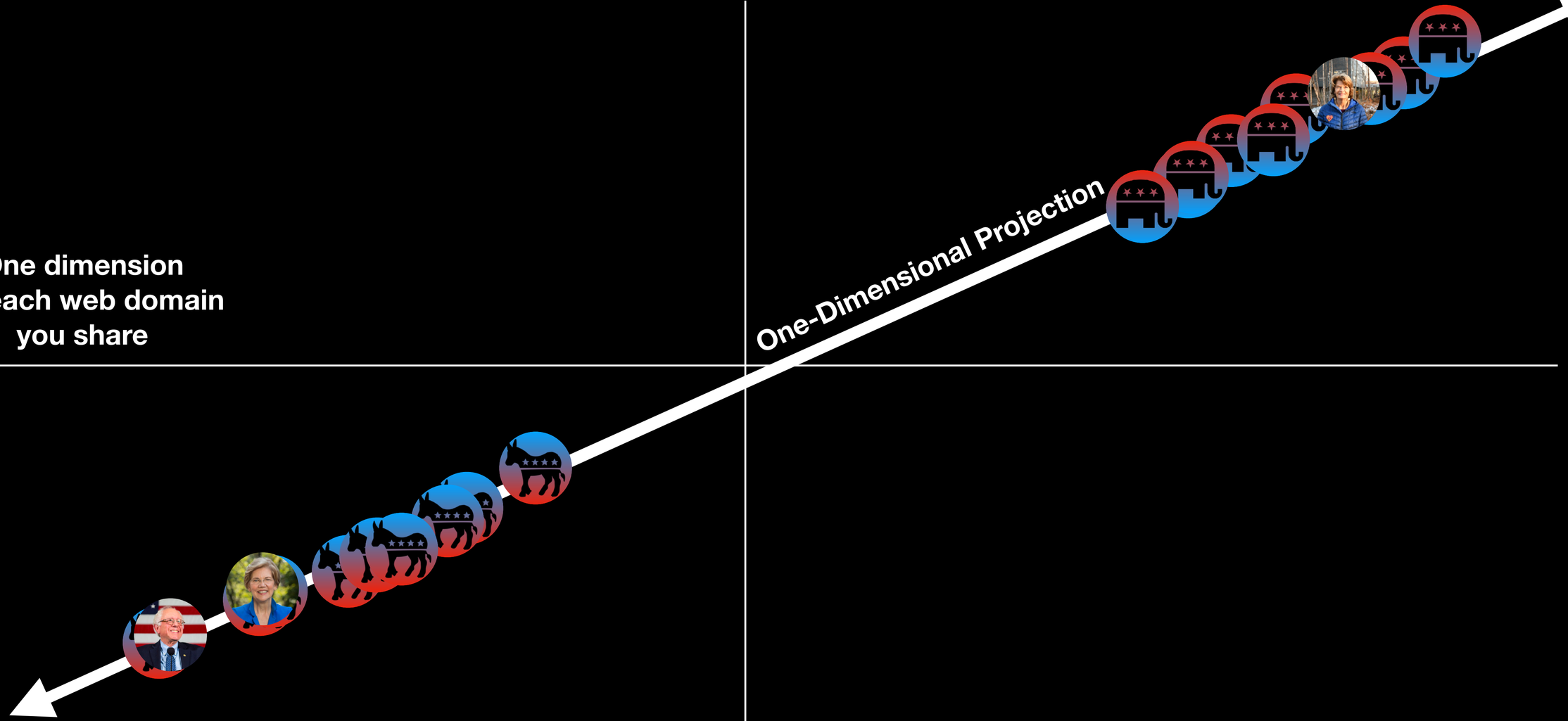




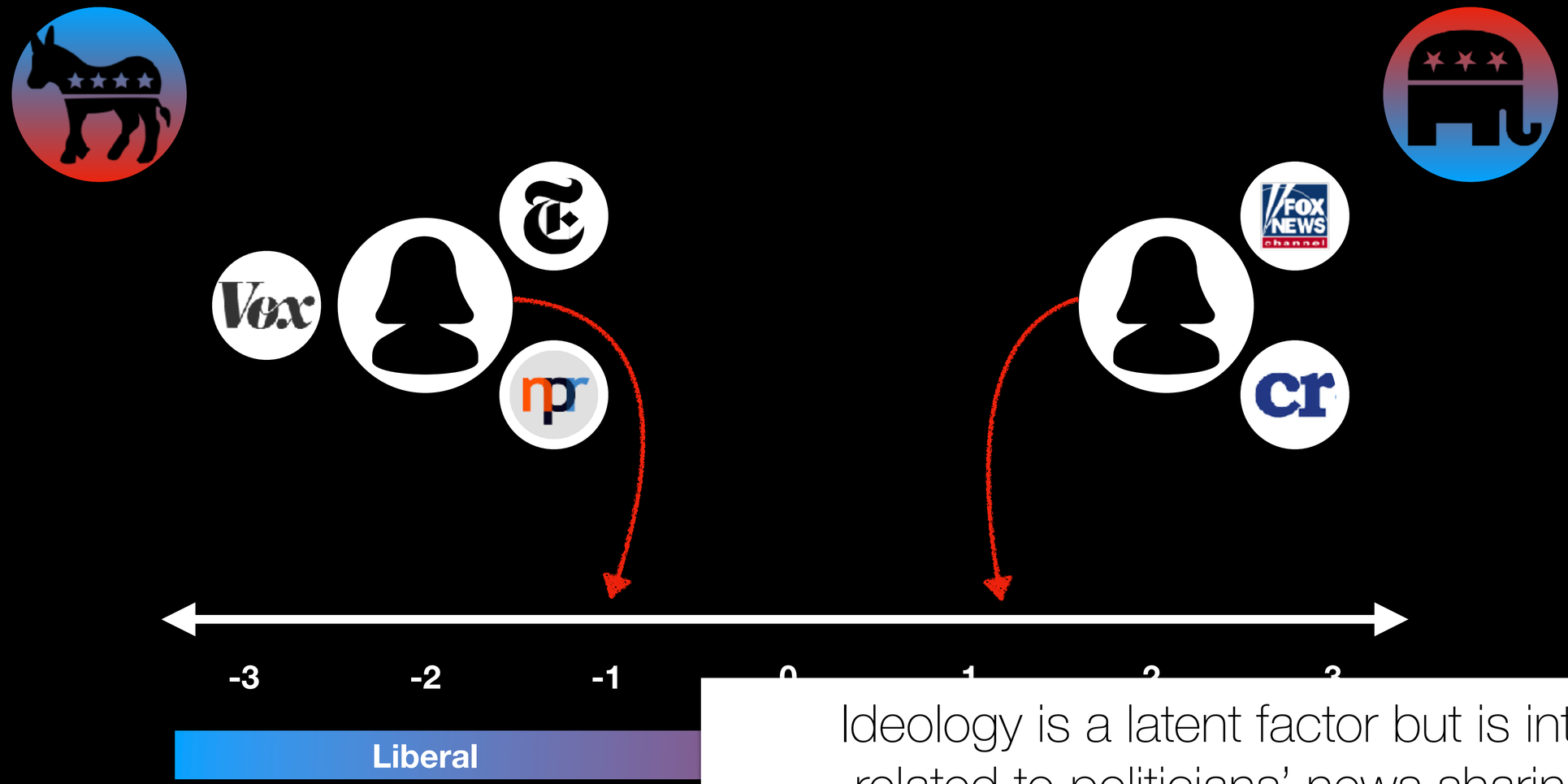
Link Sharing and Ideology

One dimension
for each web domain
you share

One-Dimensional Projection



Link Sharing and Ideology



Ideology is a latent factor but is intrinsically related to politicians' news sharing activity

1. thing, bad, guy, happen, lot, worry, imagine, good, forget, shit -> (UNKNOWN)
3. work, great, team, hard, amazing, time, idea, part, proud, job-> Appreciation
4. health, care, patient, hospital, public, doctor, medical, system, nurse, treat-> Healthcare
5. virus, people, die, corona, kill, infect, person, thousand, black, dead-> Deaths during covid
6. case, death, confirm, report, number, total, rise, record, recover, late-> Number of cases/deaths
8. mask, face, wear, hand, wash, water, touch, cover, clean, glove->Precautions
10. time, change, learn, human, important, experience, manage, create, covid, tip->Survival tips
11. pay, money, job, company, lose, cut, month, employee, leave, cost-> Layoffs/Pay-cut
12. covid, call, state, pandemic, medium, situation, social, lead, current, issue->
13. watch, live, play, video, talk, game, start, season, set, win->Sports on TV
14. time, year, life, back, long, month, save, wait, normal, spend-> Wait to go back to normal
15. give, people, agree, police, reason, fine, chance, power, mind, understand->(UNKNOWN)
16. people, covid, man, woman, group, chinese, warn, affect, issue, concern->(UNKNOWN)
17. trump, lie, wrong, fail, blame, stupid, stop, truth, vote, dangerous-> Blaming Trump
18. covid, today, feel, day, sign, good, bit, call, hope, finally-> Family Occasions
19. lockdown, day, isolation, quarantine, music, busy, happy, coronalockdown, single, dog->Schedule
20. people, put, follow, place, rule, law, break, lock, continue, avoid-> (Language Barrier)
21. covid, share, listen, post, fact, important, message, check, explain, list-> Sharing opinions
22. government, response, plan, measure, public, action, canadian, official, announce, release-> Govt. response
23. read, hear, story, word, write, sad, thought, book, article, full-> Reading books
24. buy, big, hit, market, panic, fear, store, demand, sell, shop->Demand and supply
25. home, stay, family, safe, friend, send, healthy, strong, distance, hope-> Safety Messages
26. news, question, travel, quarantine, return, answer, flight, citizen, break, leave-> Travel restrictions
27. make, good, thing, point, happen, decision, big, easy, light, sense->(UNKNOWN)
28. support, business, community, local, small, provide, donate, fund, service, program-> Support loc
29. treatment, vaccine, find, interesting, base, study, research, cure, scientist, science-> Vaccine
30. pandemic, crisis, global, economy, impact, economic, future, opportunity, industry, threat->Econo
31. spread, stop, child, school, close, kid, student, parent, risk, slow->Closing of Schools In UK
32. run, walk, city, hour, open, exercise, drive, sit, close, space->Transportation/ Exercise
33. food, order, deliver, eat, delivery, drink, pick, animal, local, restaurant->Food
34. test, symptom, positive, show, testing, covid, result, contact, cough, flu->Test results/ Symptoms
35. covid, high, risk, death, infection, number, rate, low, population, datum->
36. free, update, late, information, link, check, visit, resource, advice, online-> Information links
37. worker, fight, covid, staff, continue, protect, essential, resident, service, key-> Medical staff appre
38. world, country, pandemic, leader, leadership, problem, real, show, poor, rest->Leadership opinion

Topic models puts “words” in groups

These groups may not always make sense

1. thing, bad, guy, happen, lot, worry, imagine, good, forget, shit -> (UNKNOWN)
2. love, staysafe, stayhome, good, hope, flattenthecurve, morning, stayathome, night, beautiful-> Safe to stay home.
3. work, great, team, hard, amazing, time, idea, part, proud, job-> Appreciation
4. health, care, patient, hospital, public, doctor, medical, system, nurse, treat-> Healthcare
5. virus, people, die, corona, kill, infect, person, thousand, black, dead-> Deaths during covid
6. case, death, confirm, report, number, total, rise, record, recover, late-> Number of cases/deaths
7. week, lockdown, start, end, level, move, restriction, early, covid, begin->Extension of lockdown
8. mask, face, wear, hand, wash, water, touch, cover, clean, glove->Precautions
9. due, today, join, cancel, live, pm, tonight, tomorrow, event, covid-> Cancellation of events
10. time, change, learn, human, important, experience, manage, create, covid, tip->Survival tips
11. pay, money, job, company, lose, cut, month, employee, leave, cost-> Layoffs/Pay-cut
12. covid, call, state, pandemic, medium, situation, social, lead, current, issue->
13. watch, live, play, video, talk, game, start, season, set, win->Sports on TV
14. time, year, life, back, long, month, save, wait, normal, spend-> Wait to go back to normal
15. give, people, agree, police, reason, fine, chance, power, mind, understand->(UNKNOWN)
16. people, covid, man, woman, group, chinese, warn, affect, issue, concern->(UNKNOWN)
17. trump, lie, wrong, fail, blame, s
18. covid, today, feel, day, sign, go
19. lockdown, day, isolation, quara
20. people, put, follow, place, rule,
21. covid, share, listen, post, fact, i
22. government, response, plan, m
23. read, hear, story, word, write, sad, thought, book, article, full-> Reading books
24. buy, big, hit, market, panic, fear, store, demand, sell, shop->Demand and supply
25. home, stay, family, safe, friend, send, healthy, strong, distance, hope-> Safety Messages
26. news, question, travel, quarantine, return, answer, flight, citizen, break, leave-> Travel restrictions
27. make, good, thing, point, happen, decision, big, easy, light, sense->(UNKNOWN)
28. support, business, community, local, small, provide, donate, fund, service, program-> Support local businesses
29. treatment, vaccine, find, interesting, base, study, research, cure, scientist, science-> Vaccine
30. pandemic, crisis, global, economy, impact, economic, future, opportunity, industry, threat->Economy threat
31. spread, stop, child, school, close, kid, student, parent, risk, slow->Closing of Schools In UK
32. run, walk, city, hour, open, exercise, drive, sit, close, space->Transportation/ Exercise
33. food, order, deliver, eat, delivery, drink, pick, animal, local, restaurant->Food
34. test, symptom, positive, show, testing, covid, result, contact, cough, flu->Test results/ Symptoms
35. covid, high, risk, death, infection, number, rate, low, population, datum->
36. free, update, late, information, link, check, visit, resource, advice, online-> Information links
37. worker, fight, covid, staff, continue, protect, essential, resident, service, key-> Medical staff appreciation
38. world, country, pandemic, leader, leadership, problem, real, show, poor, rest->Leadership opinions(Good/ Bad)

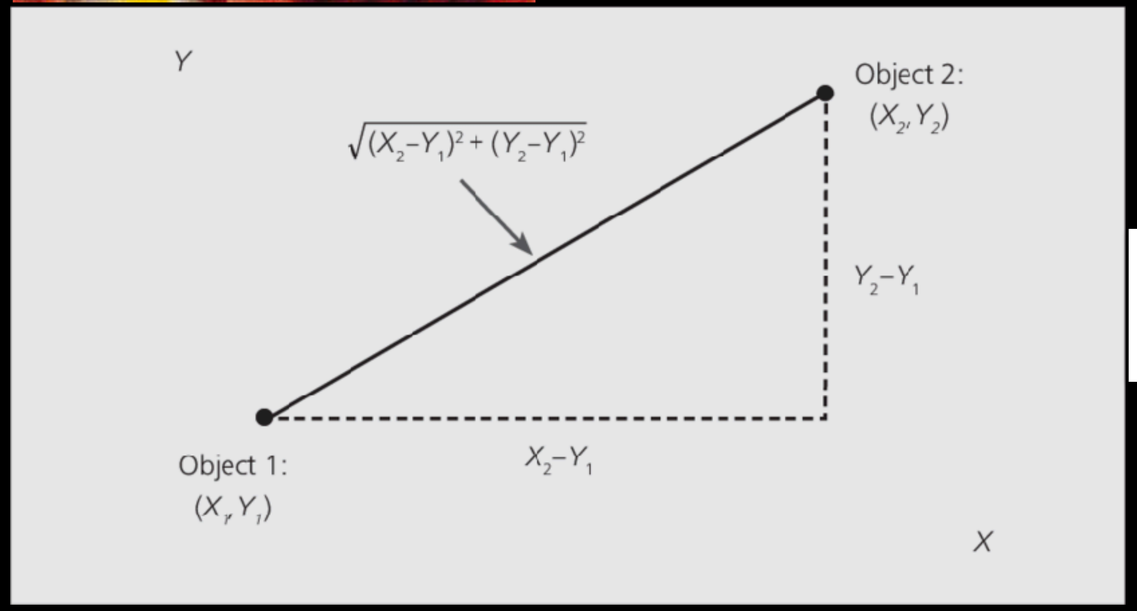
No guarantee that lower-dimensional projections capture a real-world relationship

Why might we want to reduce a dataset into fewer dimensions?

Euclidean Distance d(x,y):

Similarity: $1/d(r_A, r_B)$

Similarity $\rightarrow \infty$ as $d(r_A, r_B) \rightarrow 0$



review articles

DOI:10.1145/2347736.2347755

Tapping into the "folk knowledge" needed to advance machine learning applications.

BY PEDRO DOMINGOS

A Few Useful Things to Know About Machine Learning

is needed to successfully develop machine learning applications is not readily available in them. As a result, many machine learning projects take much longer than necessary or wind up producing less-than-ideal results. Yet much of this folk knowledge is fairly easy to communicate. This is the purpose of this article.

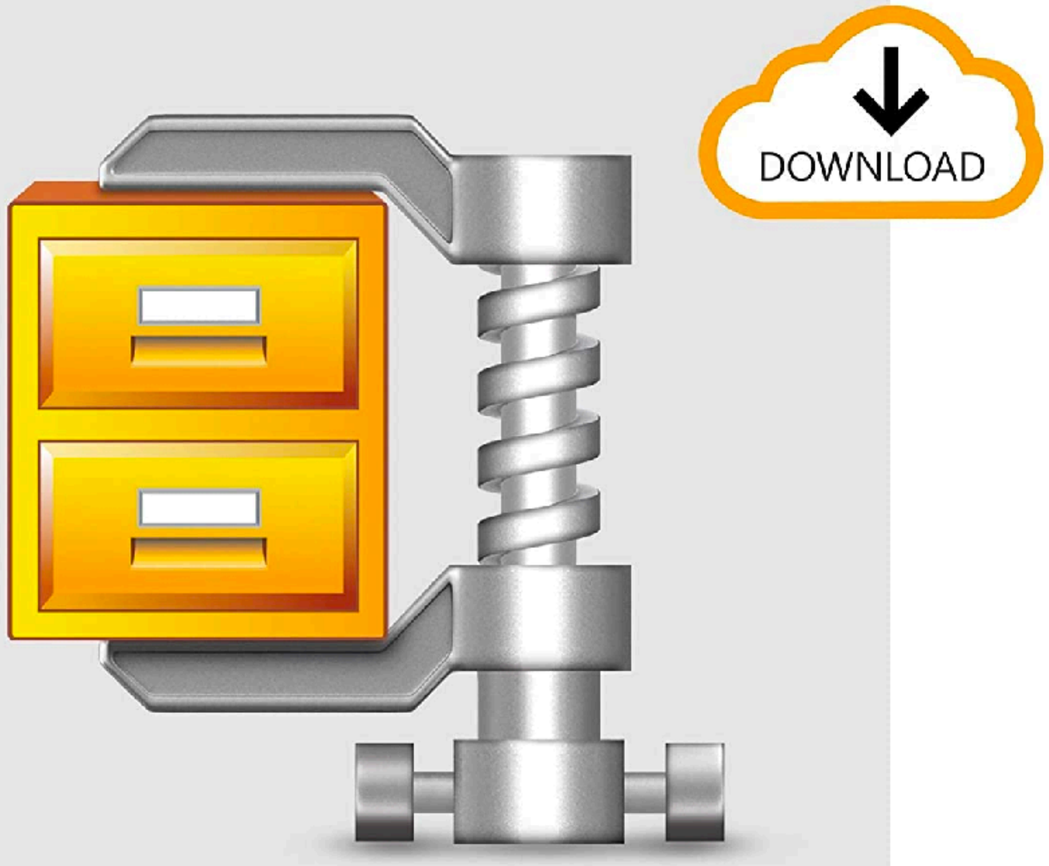
» key insights

- Machine learning is widely used in computer science and other fields. However, developing successful machine learning applications requires a substantial amount of "black art" that is difficult to find in textbooks.
- This article summarizes 12 key lessons that machine learning researchers and practitioners have learned. These include pitfalls to avoid, important issues to focus on, and answers to common questions.

MACHINE LEARNING SYSTEMS automatically learn programs from data. This is often a very attractive

fraud detection, stock trading, drug design, and many other applications. A recent report from the McKinsey Global Institute asserts that machine learning (a.k.a. data mining or predictive analytics) will be the driver of the next big wave of innovation.[15] Several fine textbooks are available to interested practitioners and researchers (for example, Mitchell[16] and Witten et al.[24]). However, much of the "folk knowledge" that

Uninformative for points in higher dimensions



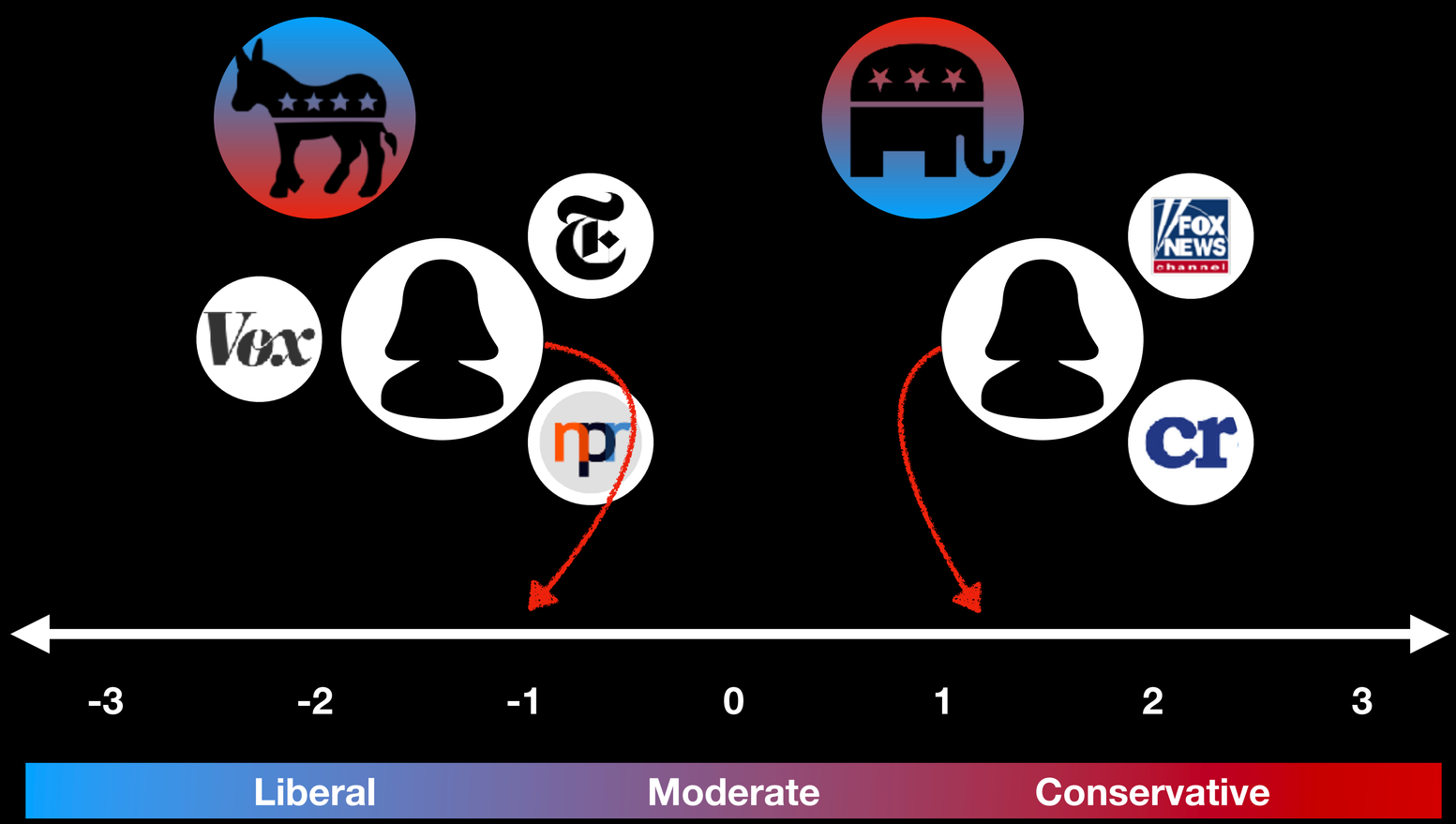
120KB
Uncompressed → Compressed
68KB

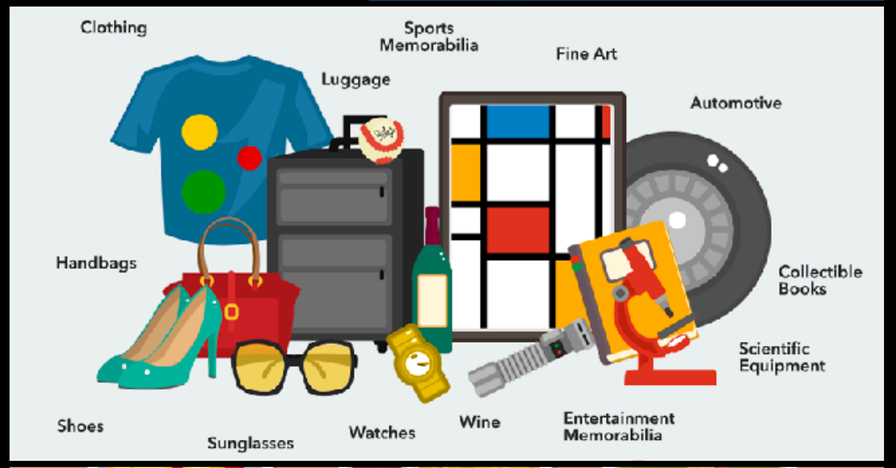
What Is Image Compression?

Uncompressed Waveform	Compressed Waveform

Dimensionality reduction with a different goal

Less space at expense of noise/error, regardless of latent factors

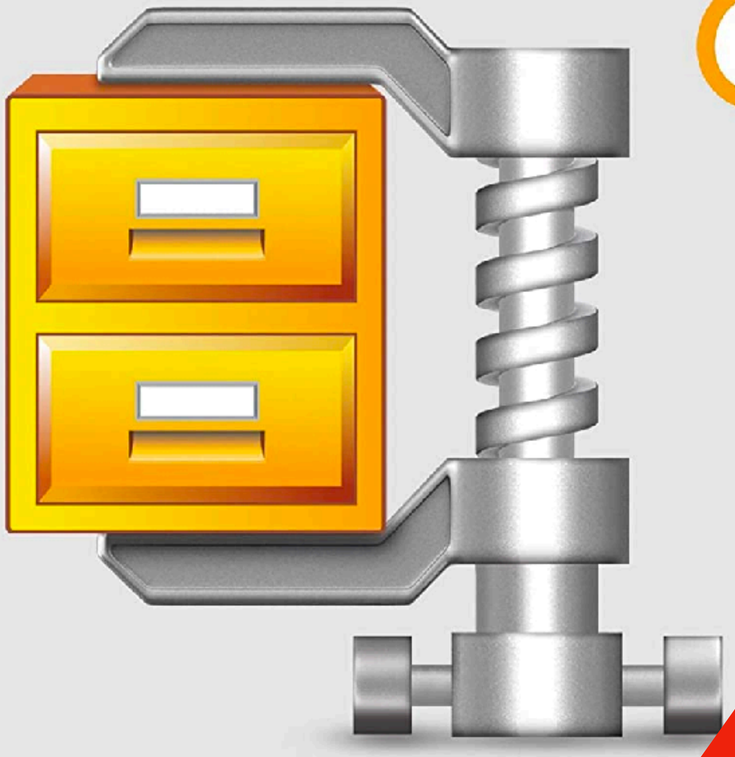




All recover some form of underlying structure

Do you need to know the latent factors to compress data?

No, you do not

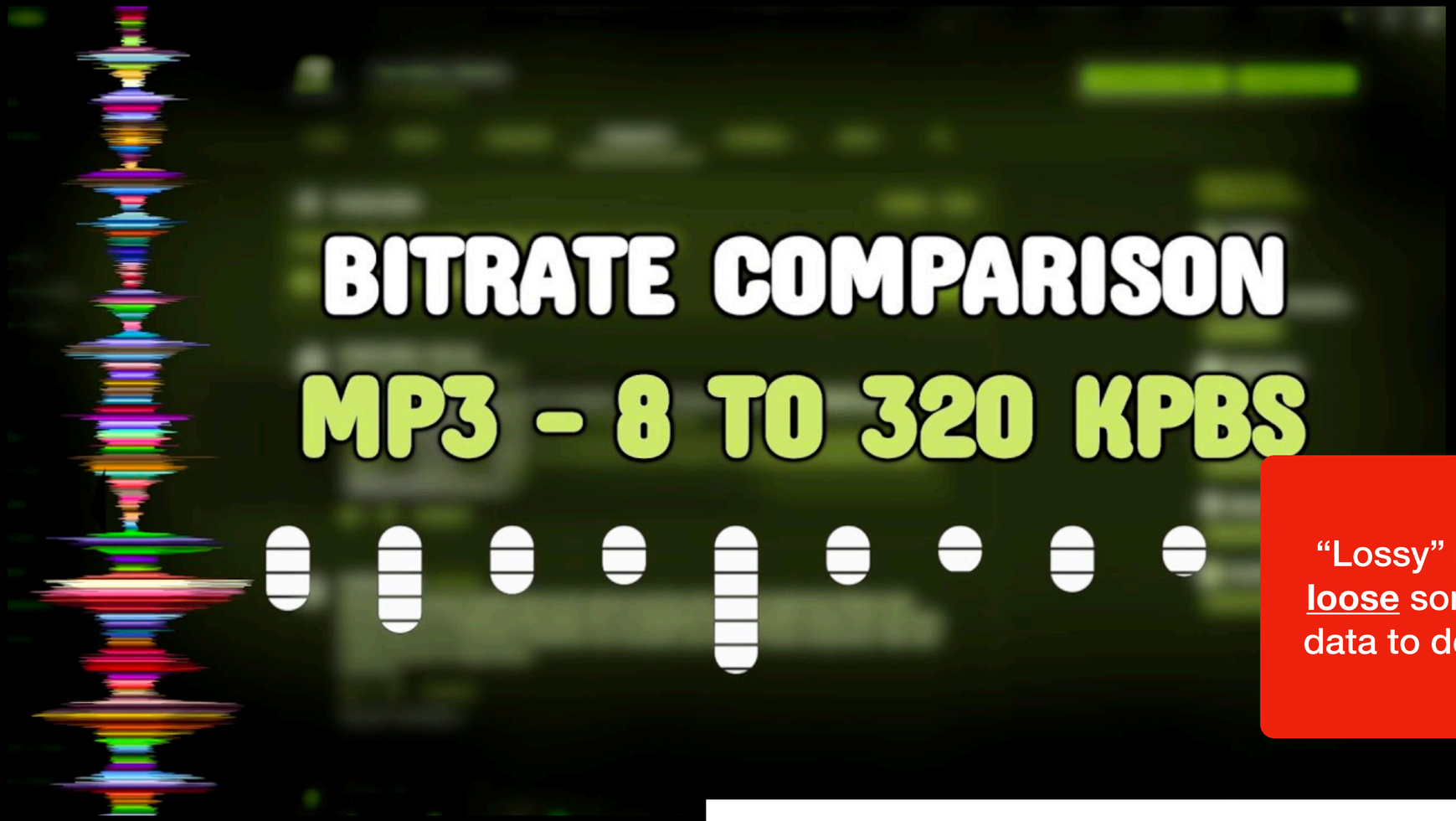


WinZip is a “lossless” compression scheme

WinZip 8 Mac

Don’t need to know underlying structure

Only need to recover the “original” data



"Lossy" means we **loose** some original data to decompress

MP3 compresses audio in a "lossy" way

3 KB

PNG

JPEG is lossy

3 KB

JPEG

Quality 20

24 KB

JPEG

Quality 100

Compression prioritizes reduced size while minimizing “reconstruction error”

This Module's Learning Objectives

Define the connection between latent factors and dimensionality reduction

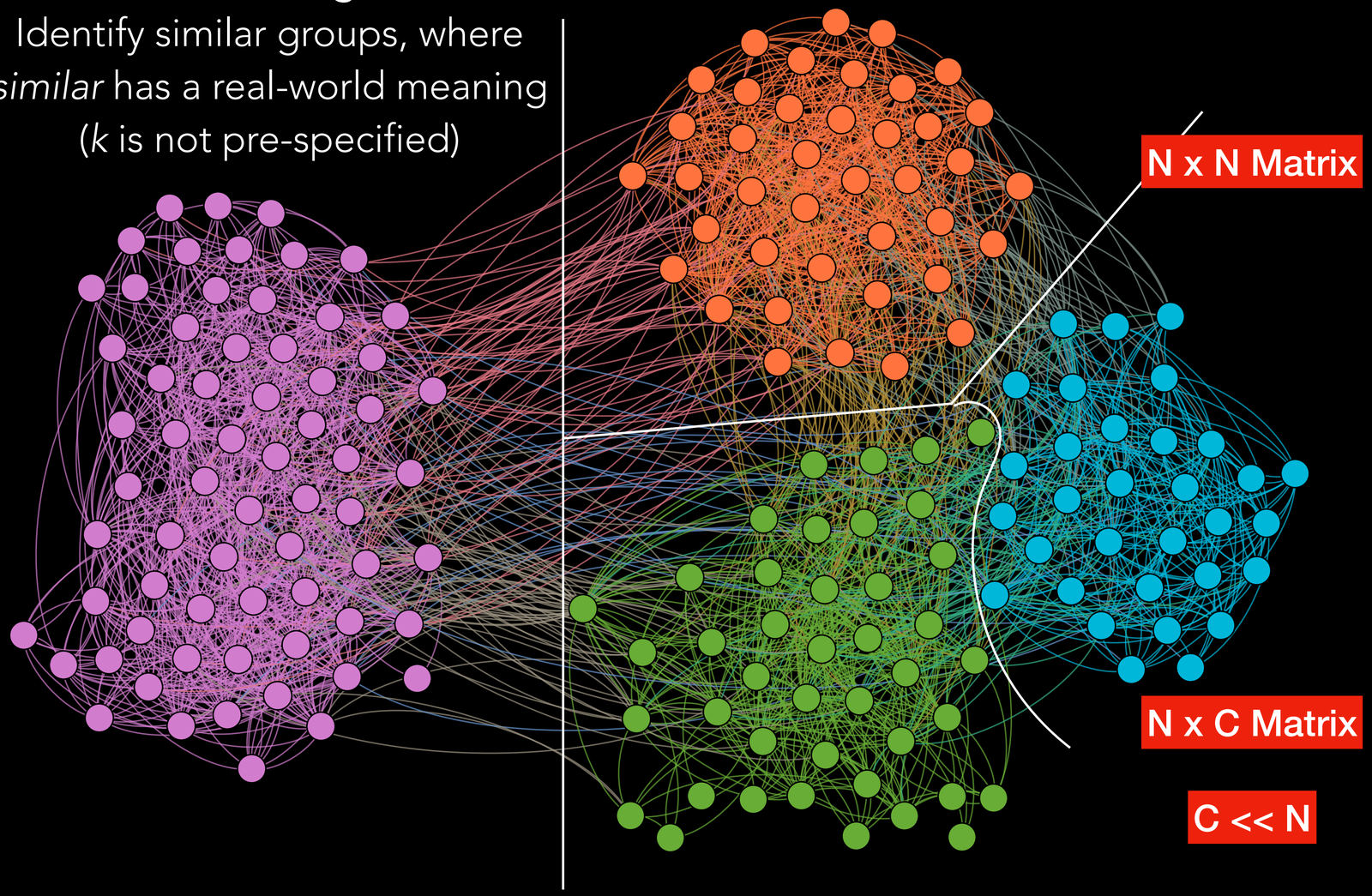
Describe at least two methods for dimensionality reduction

This Module's Learning Objectives

Define the connection between latent factors and dimensionality reduction

Describe at least two methods for dimensionality reduction

Clustering:
Identify similar groups, where *similar* has a real-world meaning (*k* is not pre-specified)



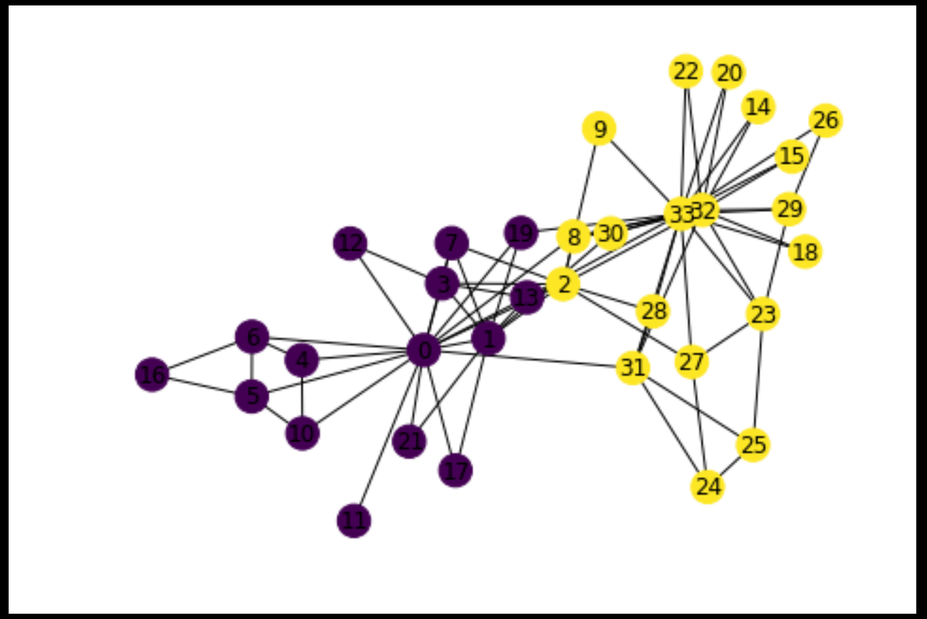
Adjacency Matrix *A*

	v_1	v_2	v_3	…	v_n
v_1	0	0	1		1
v_2	1	0	0		0
v_3	0	1	0		0
…					
v_n	0	0	0		0

	C_1	C_2	C_3	C_4
v_1	0	0	1	0
v_2	1	0	0	0
v_3	0	1	0	0
…				
v_n	0	0	0	1

Girvan-Newman Clustering

Zachary's Karate Club Graph

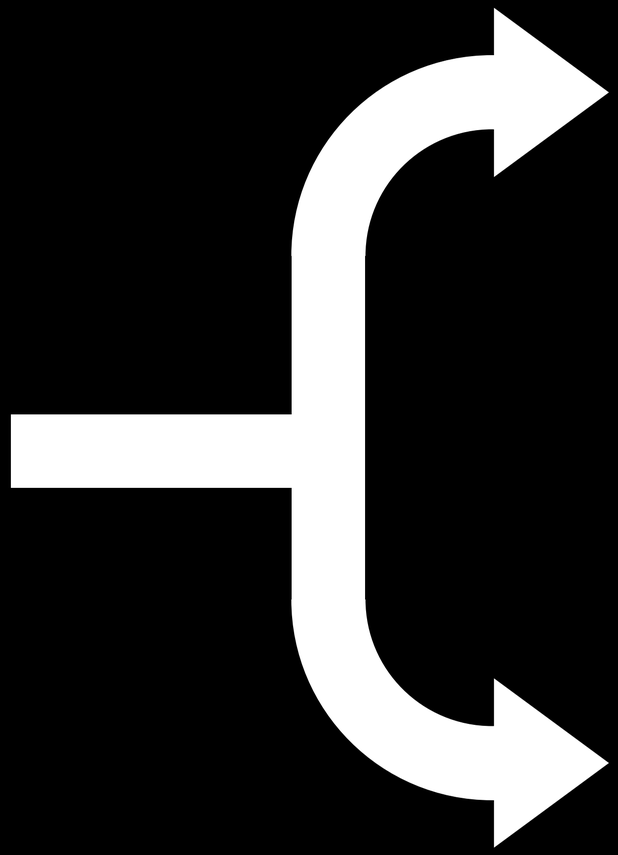


Have discussed methods for extracting communities from graphs or selecting important features

Many general methods also exist

Some feature matrix *A*

	d_1	d_2	d_3	…	d_n
x_1	0	0	1		1
x_2	1	0	0		0
x_3	0	1	0		0
…					
x_m	0	0	0		0



Decompose to latent factors

Reduce dimensions with minimum reconstruction error

Some feature matrix *A*

	d_1	d_2	d_3	…	d_n
x_1	0	0	1		1
x_2	1	0	0		0
x_3	0	1	0		0
…					
x_m	0	0	0		0

First Principal Component

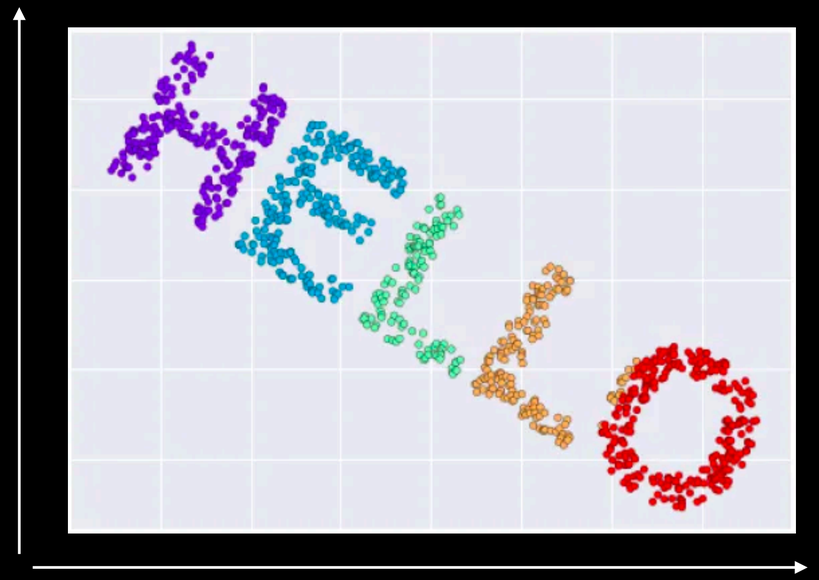
Second Principal Component

	d'_1	d'_2
x_1	0	0
x_2	1	0
x_3	0	1
…		
x_n	0	0

… **n components ordered by eigenvalues**

Keeping all components produces a noiseless rotation

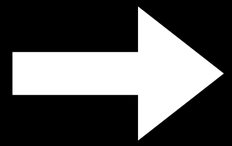
Principal Component Analysis (PCA)



Project onto dimension with highest variance

Can achieve the same reduction through a process of “decomposition”

Some feature matrix *A* (m x n)



Sub-matrix U (m x d) X **Sub-matrix V (d x n)**

Or matrix factorization

	Feature 1	Feature 2
	0.24	0.16
	0.28	0.18
	0.3	0.2
	0.24	0.16

User Matrix

	Feature 1	Feature 2
	0.24	0.16
	0.28	0.18
	0.3	0.2
	0.24	0.16

User Matrix

X

				
Feature 1	1.5	1.2	1.0	0.8
Feature 2	1.7	0.6	1.1	0.4

Home Matrix

	Feature 1	Feature 2
	0.24	0.16
	0.28	0.18
	0.3	0.2
	0.24	0.16

User Matrix

X

				
Feature 1	1.5	1.2	1.0	0.8
Feature 2	1.7	0.6	1.1	0.4

Home Matrix

=

				
	0.2	0.9	0.4	0.2
	0.8	0.2	0.7	0.2
	0.2	1.0	0.2	0.4
	0.2	0.7	0.8	0.2

User Home Matrix

But what if you could go backwards?

				
	0.2	0.9	0.4	0.2
	0.8	0.2	0.7	0.2
	0.2	1.0	0.2	0.4
	0.2	0.7	0.8	0.2

User-Home Matrix

				
	0.2	0.9	0.4	0.2
	0.8	0.2	0.7	0.2
	0.2	1.0	0.2	0.4
	0.2	0.7	0.8	0.2

User-Home Matrix

≅

What approximately equals this matrix?

				
	0.2	0.9	0.4	0.2
	0.8	0.2	0.7	0.2
	0.2	1.0	0.2	0.4
	0.2	0.7	0.8	0.2

User-Home Matrix

≅

	Feature 1	Feature 2
	0.24	0.16
	0.28	0.18
	0.3	0.2
	0.24	0.16

User Matrix

	0.2	0.9	0.4	0.2
	0.8	0.2	0.7	0.2
	0.2	1.0	0.2	0.4
	0.2	0.7	0.8	0.2

User-Home Matrix

≅

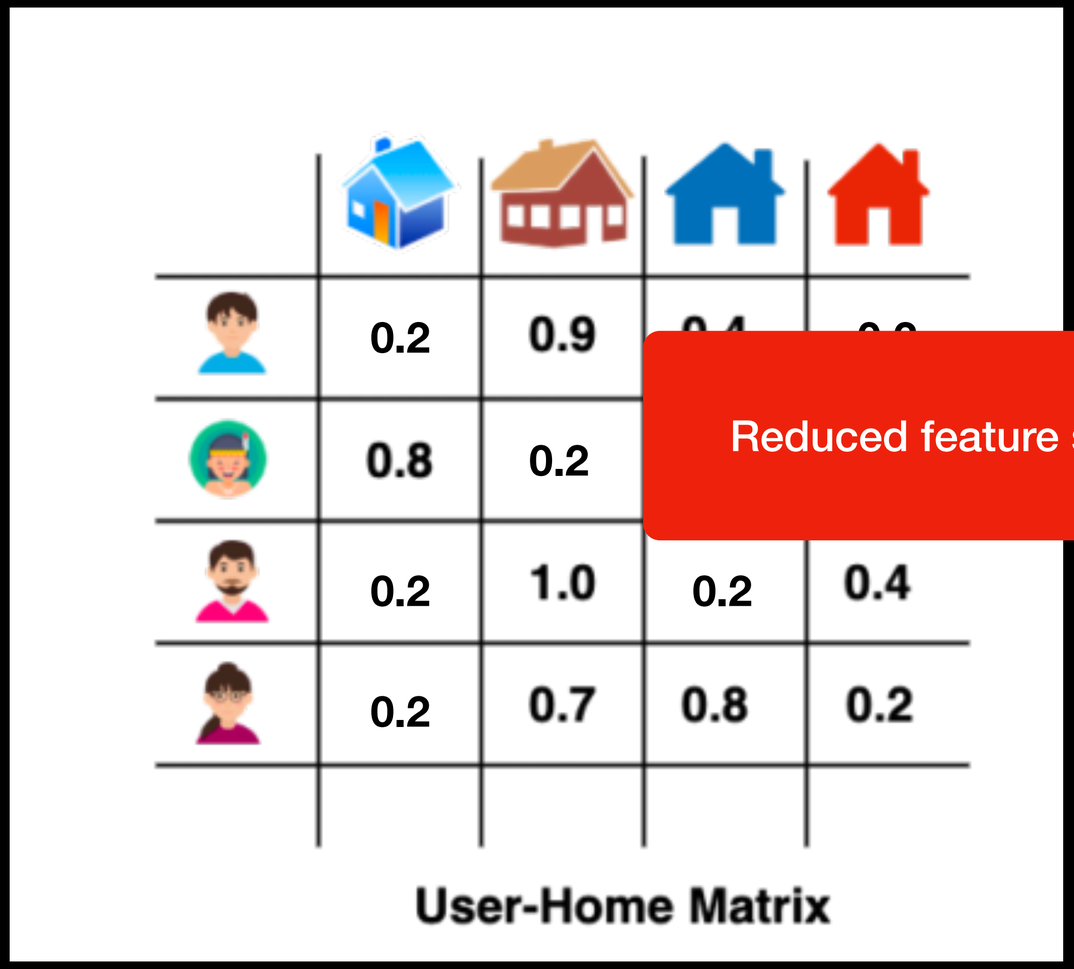
	Feature 1	Feature 2
	0.24	0.16
	0.28	0.18
	0.3	0.2
	0.24	0.16

User Matrix

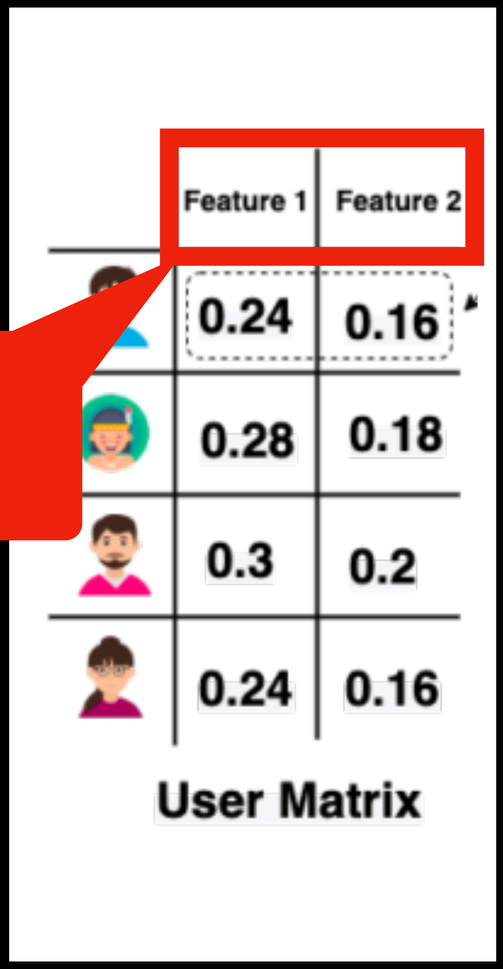
X

Feature 1	1.5	1.2	1.0	0.8
Feature 2	1.7	0.6	1.1	0.4

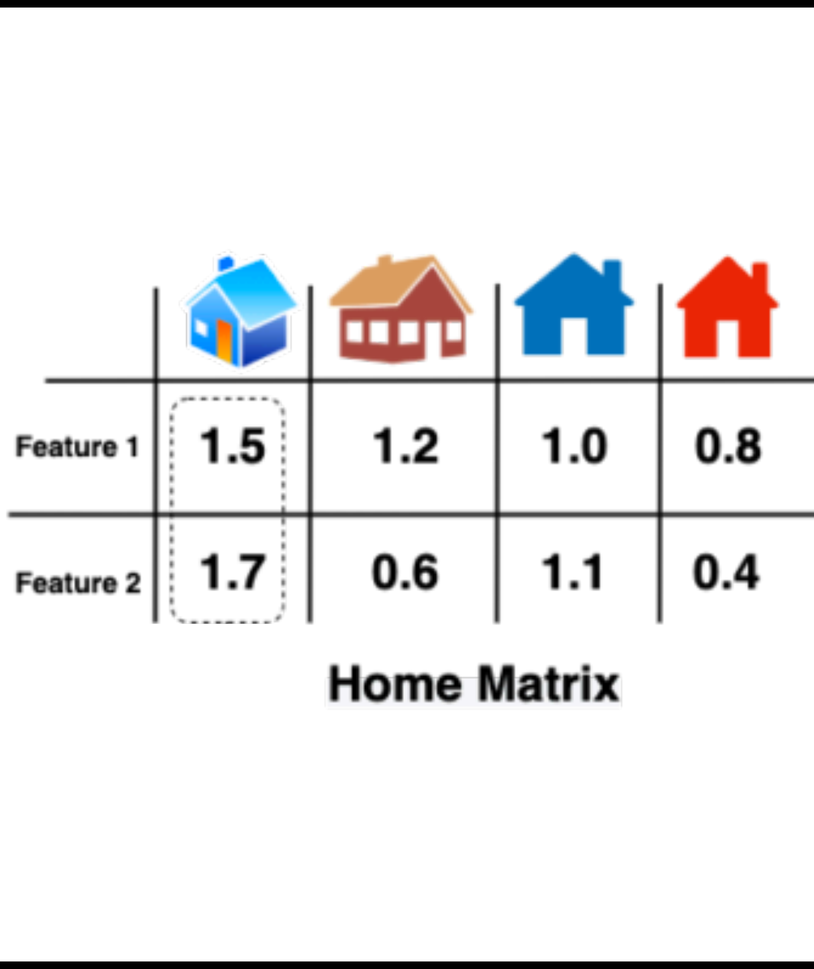
Home Matrix



Reduced feature set

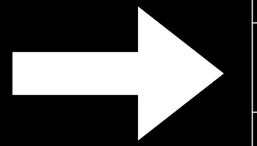


X



Some feature matrix *A* (m x n)

	d_1	d_2	d_3	…	d_n
x_1	0	0	1		1
x_2	1	0	0		0
x_3	0	1	0		0
…					
x_m	0	0	0		0



Matrix U (m x r)

	d'_1	…	d'_r
x_1	0		0
x_2	1		0
x_3	0		1
…			
x_m	0		0

X

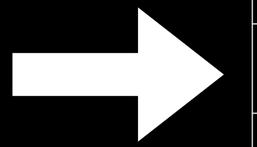
Matrix V (n x r)

^{T}

	d'_1	…	d'_r
d_1	0		0
d_2	1		0
d_3	0		1
…			
d_n	0		0

Some feature matrix *A* (m x n)

	d_1	d_2	d_3	…	d_n
x_1	0	0	1		1
x_2	1	0	0		0
x_3	0	1	0		0
…					
x_m	0	0	0		0



Matrix U (m x r)

	d'_1	…	d'_r
x_1	0		0
x_2	1		0
x_3	0		1
…			
x_m	0		0

Matrix ∑ (r x r)

s1	0	0		0
0	s2	0		0
0	0	s3		0
			…	
0	0	0	0	sr

Matrix V (n x r)

	d'_1	…	d'_r
d_1	0		0
d_2	1		0
d_3	0		1
…			
d_n	0		0

^{T}

Singular Value Decomposition (SVD)

Some feature matrix *A*

	d_1	d_2	d_3	…	d_n
x_1	0	0	1		1
x_2	1	0	0		0
x_3	0	1	0		0
…					
x_m	0	0	0		0

Matrix U (n x r)

	d'_1	…	d'_r
x_1	0		0
x_2	1		0
x_3	0		1
…			
x_n	0		0

Matrix ∑ (r x r)

s1	0	0		0
0	s2	0		0
0	0	s3		0
			…	
0	0	0	0	

Matrix V (n x r)

	d'_1	…	d'_r
v_1	0		0
v_2	1		0
v_3	0		1
…			
v_n	0		0

T

Keep only important components

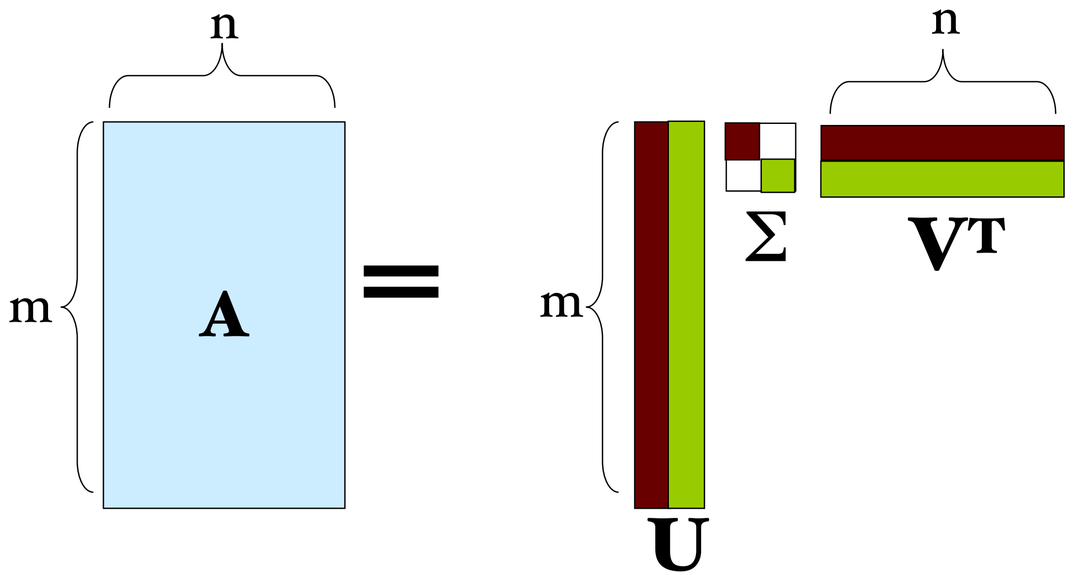
SVD - Definition

$$A_{[m \times n]} = U_{[m \times r]} \Sigma_{[r \times r]} (V_{[n \times r]})^T$$

- **A**: **Input data matrix**
  - *m* x *n* matrix (e.g., *m* documents, *n* terms)
- **U**: **Left singular vectors**
  - *m* x *r* matrix (*m* documents, *r* concepts)
- Σ: **Singular values**
  - *r* x *r* diagonal matrix (strength of each 'concept')
- **V**: **Right singular vectors**
  - *n* x *r* matrix (*n* terms, *r* concepts)

SVD

$$\mathbf{A} \approx \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \sum_i \sigma_i \mathbf{u}_i \circ \mathbf{v}_i^T$$



SVD - Properties

It is **always** possible to decompose a real matrix A into $A = U \Sigma V^T$, where

- U, Σ, V: unique
- Σ: diagonal
  - Entries (**singular values**) are positive, and sorted in decreasing order ($s_1 \geq s_2 \geq \ldots \geq 0$)

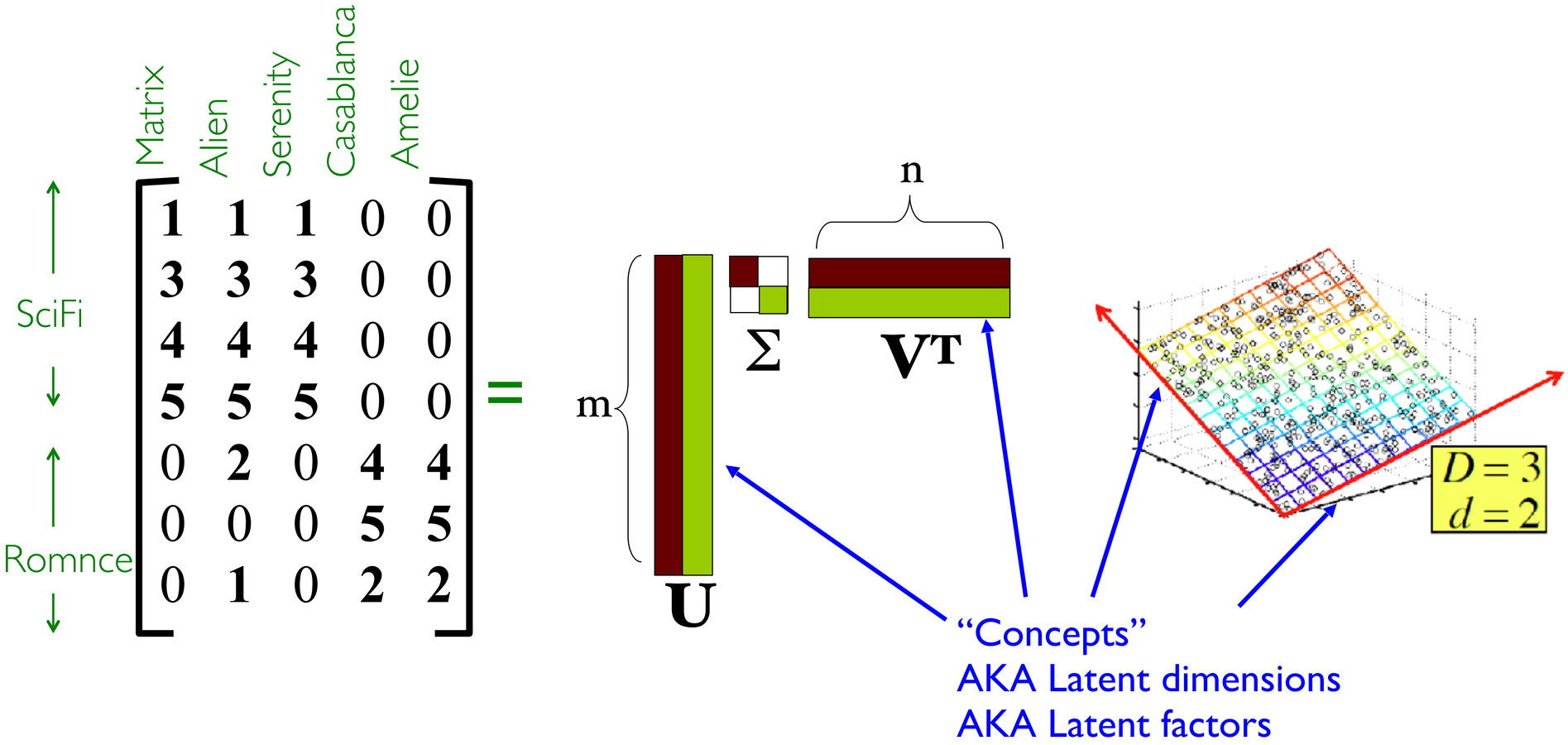
$\Sigma =$

s1	0	0		0
0	s2	0		0
0	0	s3		0
			…	
0	0	0	0	sr

Nice proof of uniqueness: http://www.mpi-inf.mpg.de/~bast/ir-seminar-ws04/lecture2.pdf

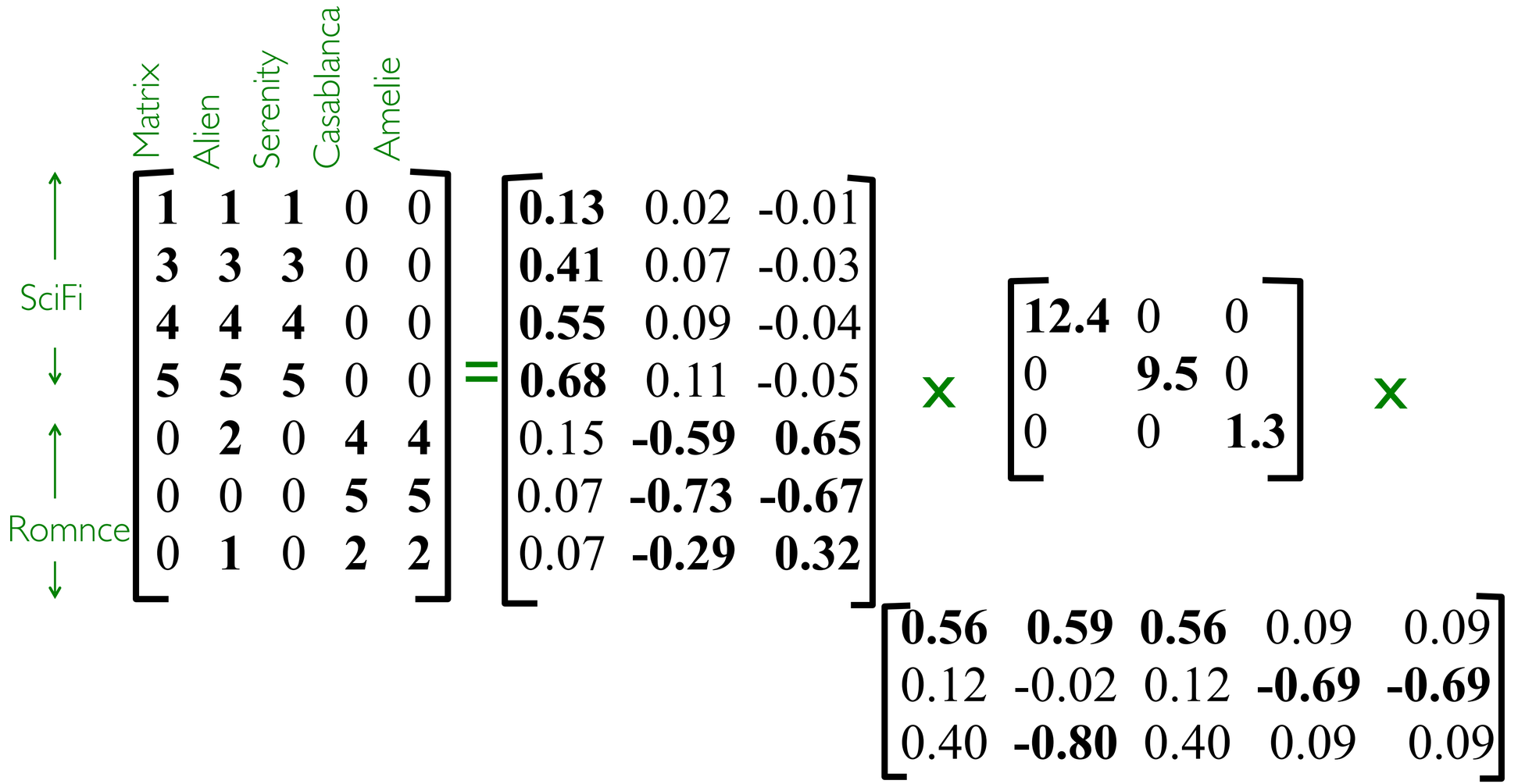
SVD – Example: Users-to-Movies

- **A = U Σ V^T - example: Users to Movies**



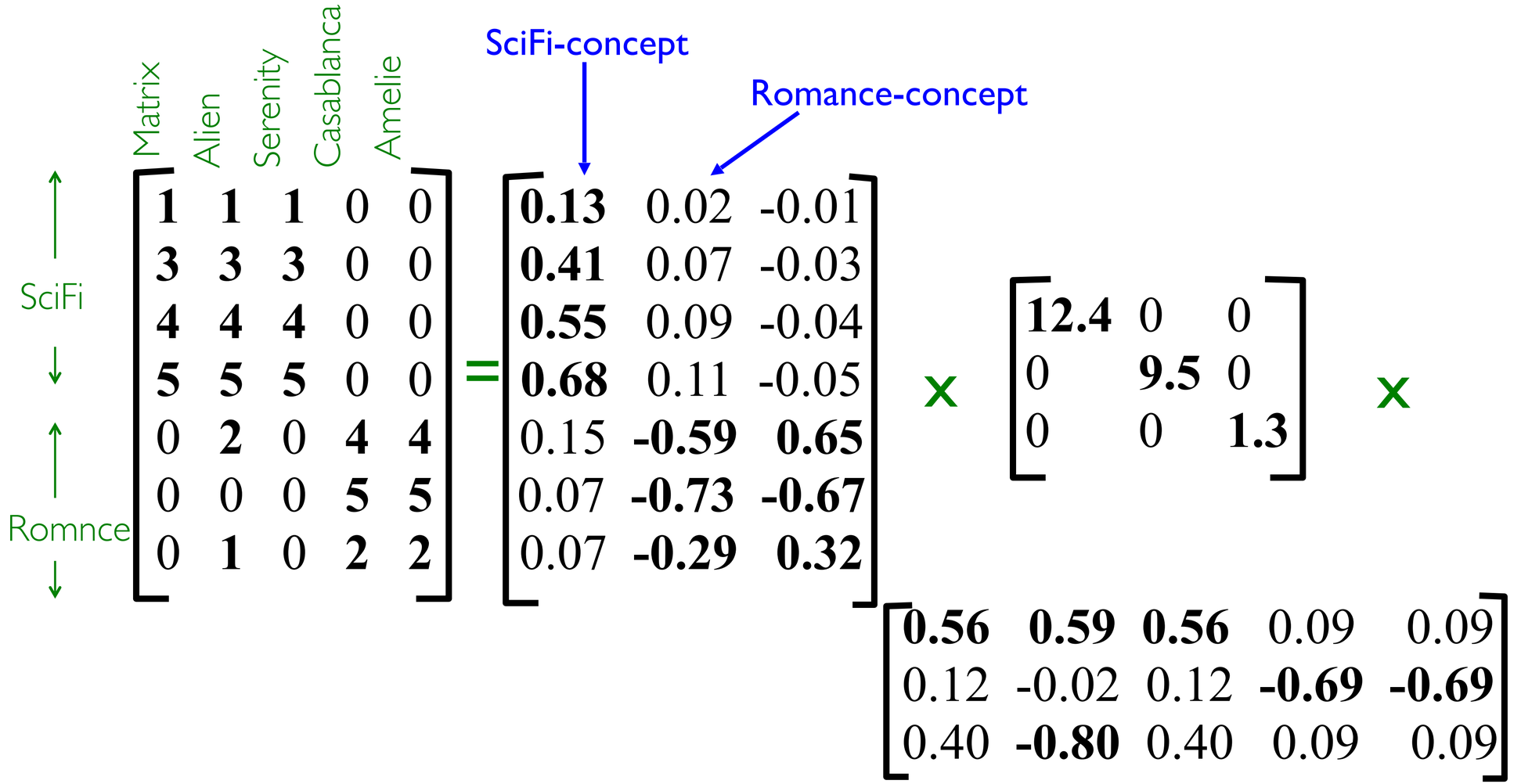
SVD – Example: Users-to-Movies

- **A = U Σ V^T - example: Users to Movies**



SVD – Example: Users-to-Movies

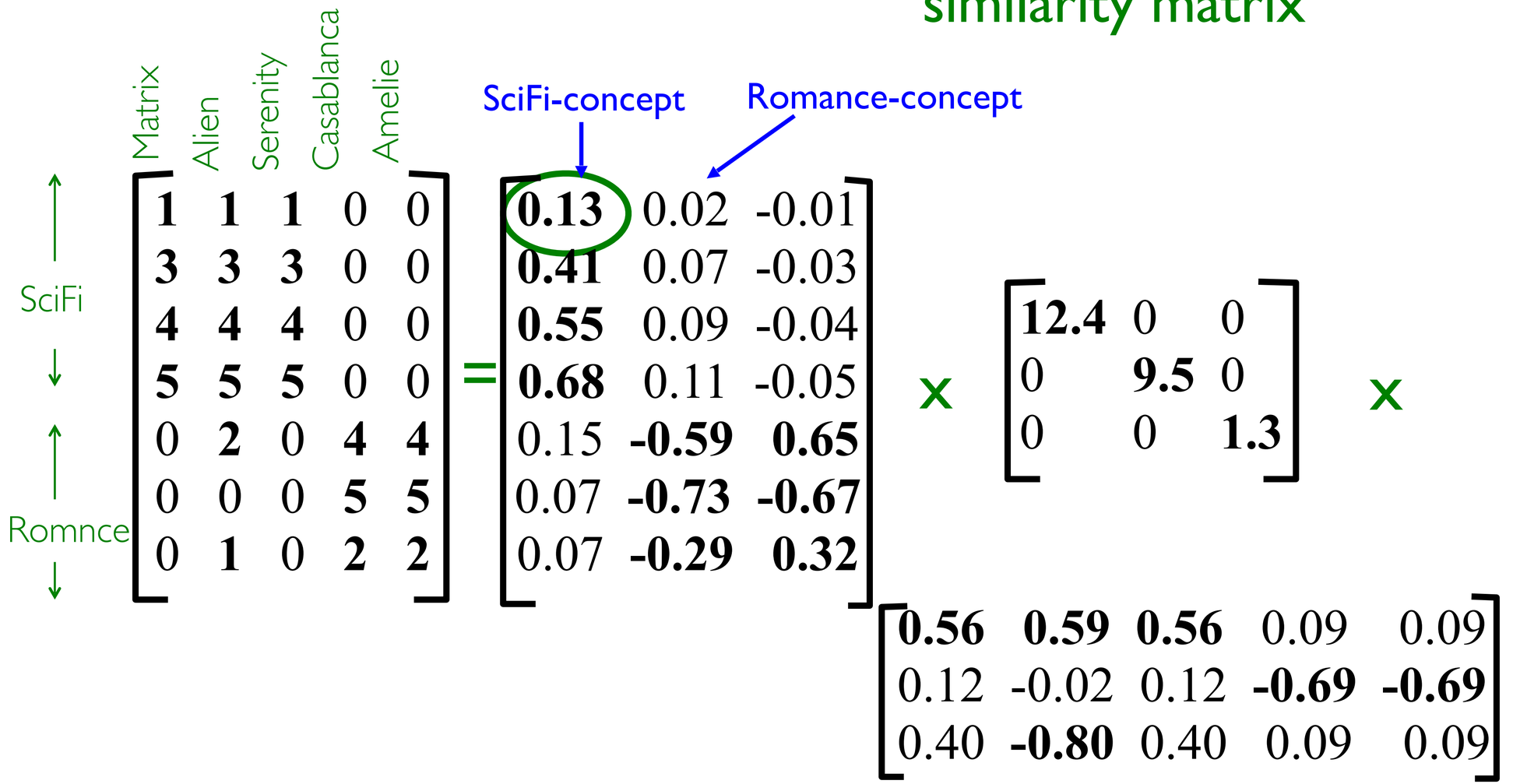
- **A = U Σ V^T - example: Users to Movies**



SVD – Example: Users-to-Movies

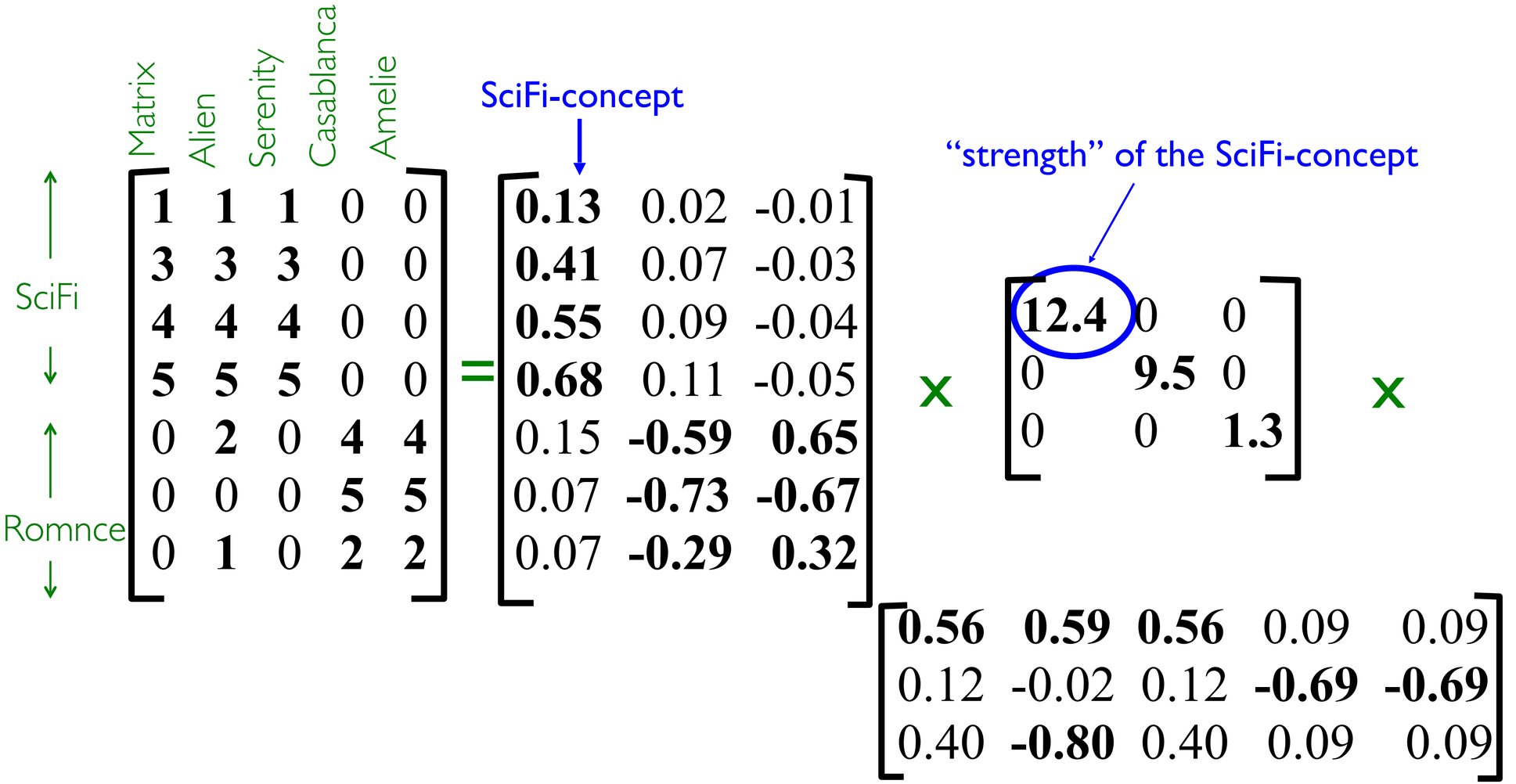
- **A = U Σ V^T - example:**

U is "user-to-concept" similarity matrix



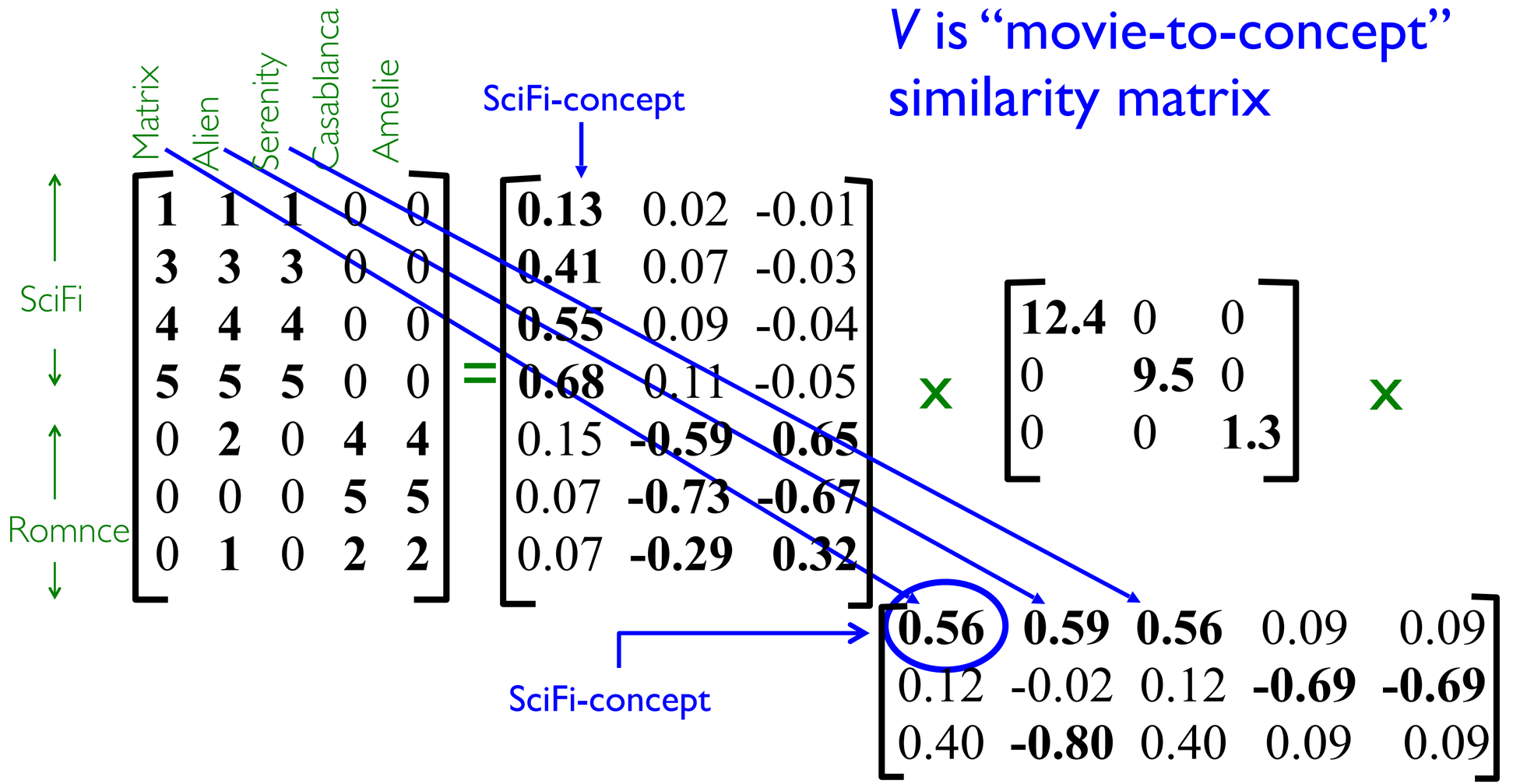
SVD – Example: Users-to-Movies

- **$A = U \Sigma V^T$ - example:**



SVD – Example: Users-to-Movies

- **A = U Σ V^T - example:**



SVD - One Interpretation

$$A_{[m \times n]} = U_{[m \times r]} \Sigma_{[r \times r]} (V_{[n \times r]})^T$$

User Interests

Concept Strengths (axes of most variation)

Movie Genre assignments

SVD - Conclusions

- **SVD: A= U Σ V^T: unique**
  - **U**: user-to-concept similarities
  - **V**: movie-to-concept similarities
  - Σ : strength of each concept
- **Dimensionality reduction:**
  - keep the few largest singular values (80-90% of 'energy')
  - SVD: picks up linear correlations

Prev Up Next

scikit-learn 0.24.1
Other versions

Please cite us if you use the software.

sklearn.decomposition.TruncatedSVD
Examples using sklearn.decomposition.TruncatedSVD

sklearn.decomposition.TruncatedSVD

class sklearn.decomposition.**TruncatedSVD**(*n_components=2*, *, *algorithm='randomized'*, *n_iter=5*, *random_state=None*, *tol=0.0*) [source]

n_components is the reduced dimensionality

Dimensi SA).

This tra ction by means of truncated singular value decomposition (SVD). Contrary data before computing the singular value decomposition. This means it can w

In particular, truncated SVD works on term count/tf-idf matrices as returned by the vectorizers in sklearn.feature_extraction.text. In that context, it is known as latent semantic analysis (LSA).

This estimator supports two algorithms: a fast randomized SVD solver, and a "naive" algorithm that uses ARPACK as an eigensolver on `X * X.T` or `X.T * X`, whichever is more efficient.

Read more in the User Guide.

Parameters:

n_components : *int, default=2*
Desired dimensionality of output data. Must be strictly less than the number of features. The default value is useful for visualisation. For LSA, a value of 100 is recommended.

algorithm : *{'arpack', 'randomized'}, default='randomized'*
SVD solver to use. Either "arpack" for the ARPACK wrapper in SciPy (scipy.sparse.linalg.svds), or "randomized" for the randomized algorithm due to Halko (2009).

n_iter : *int, default=5*
Number of iterations for randomized SVD solver. Not used by ARPACK. The default is larger than the default in randomized_svd to handle sparse matrices that may have large slowly decaying spectrum.

random_state : *int, RandomState instance or None, default=None*
Used during randomized svd. Pass an int for reproducible results across multiple function calls. See

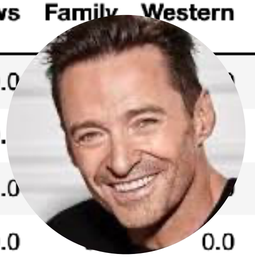
This Module's Learning Objectives

Define the connection between latent factors and dimensionality reduction

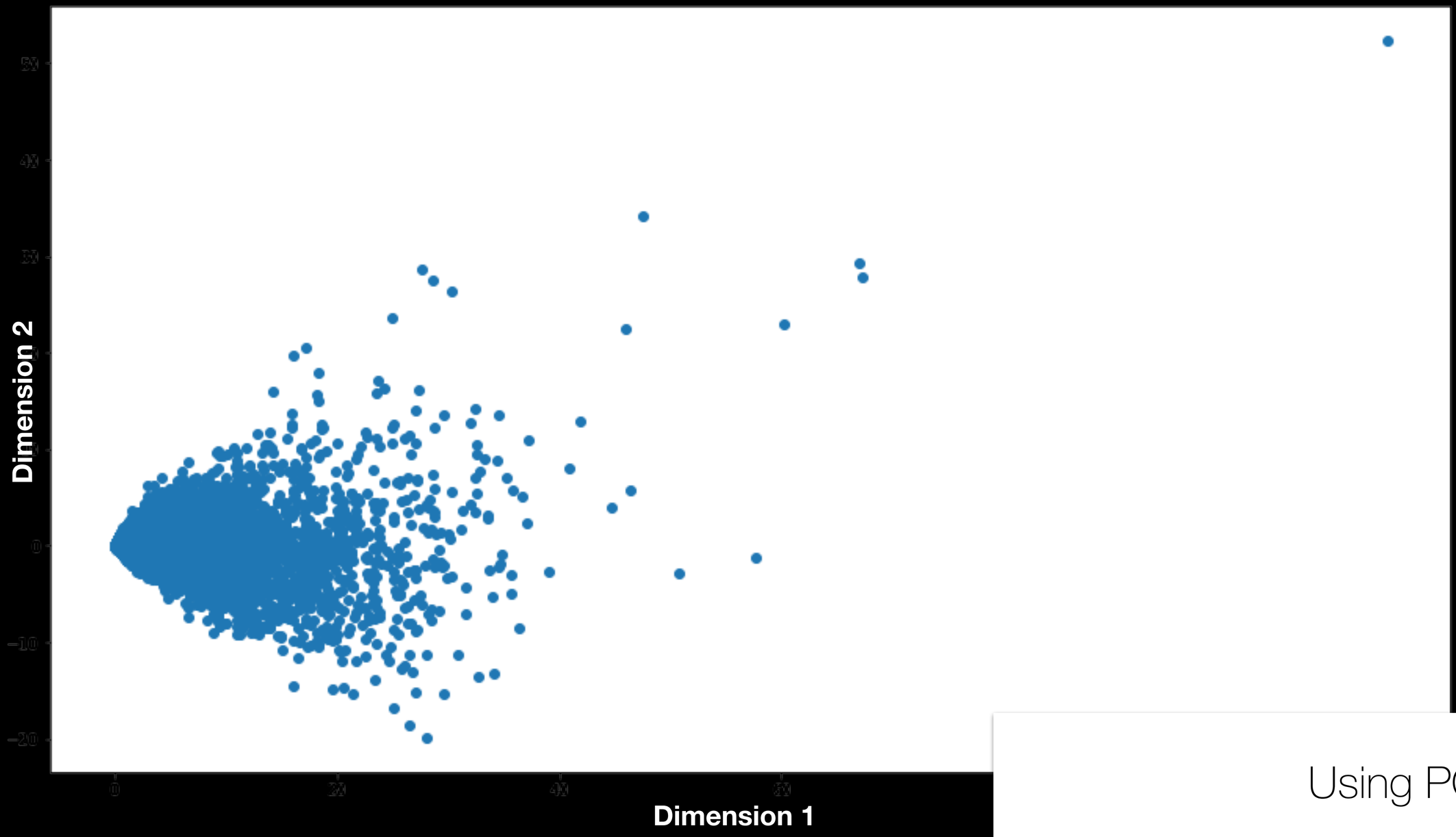
Describe at least two methods for dimensionality reduction

```
actor_genre_df.head(20)
```

	Comedy	Fantasy	Romance	Action	Crime	Adventure	Mystery	Thriller	Drama	Biography	...	Sport		News	Family	Western	Short
nm0000212	16.0	3.0	16.0	5.0	4.0	2.0	5.0	3.0	16.0	2.0	...	0.0	0.0	0.0			0.0
nm0413168	8.0	3.0	6.0	14.0	6.0	11.0	5.0	2.0	13.0	5.0	...	0.0	0.0	0.			0.0
nm0000630	10.0	2.0	6.0	4.0	1.0	2.0	2.0	4.0	17.0	6.0	...	4.0	1.0	1.0			0.0
nm0005227	12.0	1.0	3.0	2.0	0.0	3.0	0.0	1.0	5.0	1.0	...	1.0	0.0	0.0		0.0	0.0
nm0697338	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
nm1300519	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
nm0940707	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
nm0625977	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
nm0792032	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
nm0496571	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
nm2868805	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
nm2866192	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
nm0001379	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	1.0	...	1.0	0.0	0.0	1.0	0.0	0.0
nm0462648	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	2.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
nm0000953	6.0	0.0	0.0	1.0	3.0	0.0	0.0	2.0	9.0	7.0	...	0.0	0.0	0.0	0.0	0.0	0.0
nm0001782	0.0	0.0	0.0	1.0	0.0	1.0	0.0	1.0	1.0	1.0	...	0.0	0.0	0.0	0.0	0.0	0.0
nm0005077	1.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	1.0	0.0	...	0.0	0.0	0.0	1.0	0.0	0.0
nm0550626	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0								
nm0177016	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0								
nm0907480	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0								



How might you visualize these rows?



Using PCA/SVD

Prev Up Next

scikit-learn 1.2.1
Other versions

Please cite us if you use the software.

API Reference
sklearn.base: Base classes and utility functions
sklearn.calibration: Probability Calibration
sklearn.cluster: Clustering
sklearn.compose: Composite Estimators
sklearn.covariance: Covariance Estimators
sklearn.cross_decomposition: Cross decomposition
sklearn.datasets: Datasets
sklearn.decomposition: Matrix Decomposition
sklearn.discriminant_analysis: Discriminant Analysis
sklearn.dummy: Dummy estimators
sklearn.ensemble: Ensemble Methods
sklearn.exceptions: Exceptions and warnings
sklearn.experimental: Experimental
sklearn.feature_extraction: Feature Extraction
sklearn.feature_selection: Feature Selection
sklearn.gaussian_process: Gaussian Processes
: Impute

Toggle Menu

sklearn.decomposition: Matrix Decomposition

The sklearn.decomposition module includes matrix decomposition algorithms, including among others PCA, NMF or ICA. Most of the algorithms of this module can be regarded as dimensionality reduction techniques.

User guide: See the Decomposing signals in components (matrix factorization problems) section for further details.

decomposition.DictionaryLearning([...])	Dictionary learning.
decomposition.FactorAnalysis([n_components, ...])	Factor Analysis (FA).
decomposition.FastICA([n_components, ...])	FastICA: a fast algorithm for Independent Component Analysis.
decomposition.IncrementalPCA([n_components, ...])	Incremental principal components analysis (IPCA).
decomposition.KernelPCA([n_components, ...])	Kernel Principal component analysis (KPCA) [R396fc7d924b8-1].
decomposition.LatentDirichletAllocation([...])	Latent Dirichlet Allocation with online variational Bayes algorithm.
decomposition.MiniBatchDictionaryLearning([...])	Mini-batch dictionary learning.
decomposition.MiniBatchSparsePCA([...])	Mini-batch Sparse Principal Components Analysis.
decomposition.NMF([n_components, init, ...])	Non-Negative Matrix Factorization (NMF).
decomposition.MiniBatchNMF([n_components, ...])	Mini-Batch Non-Negative Matrix Factorization (NMF).
decomposition.PCA([n_components, copy, ...])	Principal component analysis (PCA).
decomposition.SparsePCA([n_components, ...])	Sparse Principal Components Analysis (SparsePCA).
decomposition.SparseCoder(dictionary, *[, ...])	Sparse coding.
decomposition.TruncatedSVD([n_components, ...])	Dimensionality reduction using truncated SVD (aka LSA).

decomposition.dict_learning(X, n_components, ...)	Solve a dictionary learning matrix factorization problem.
decomposition.dict_learning_online(X[, ...])	Solve a dictionary learning matrix factorization problem online.
decomposition.fastica(X[, n_components, ...])	Perform Fast Independent Component Analysis.
decomposition.non_negative_factorization(X)	Compute Non-negative Matrix Factorization (NMF).
decomposition.sparse_encode(X, dictionary, *)	Sparse coding.

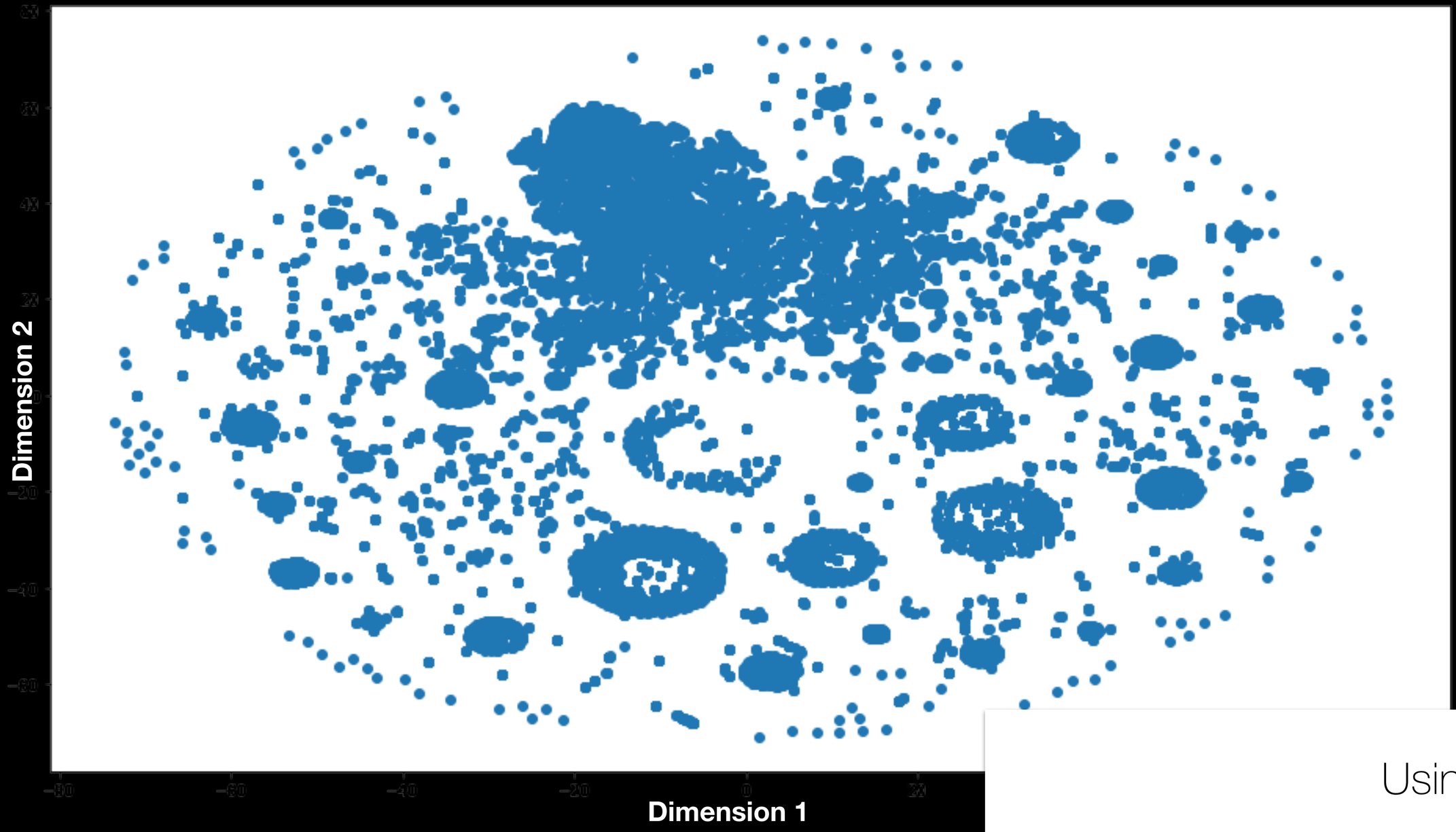
sklearn.discriminant_analysis: Discriminant Analysis

Linear Discriminant Analysis and Quadratic Discriminant Analysis

User guide: See the Linear and Quadratic Discriminant Analysis section for further details.

discriminant_analysis.LinearDiscriminantAnalysis([...])	Linear Discriminant Analysis.
discriminant_analysis.QuadraticDiscriminantAnalysis(*)	Quadratic Discriminant Analysis.

sklearn.dummy: Dummy estimators



Using tSNE

Questions?

Prof. Cody Buntain | @codybuntain | cbuntain@umd.edu
Director, Information Ecosystems Lab