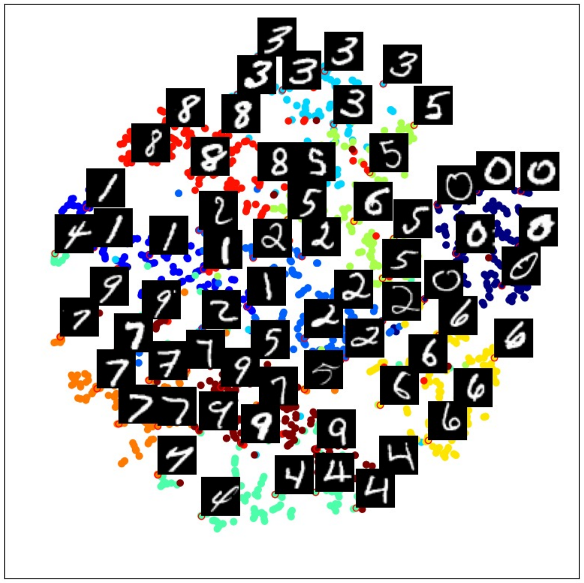
CS273A: Dimensionality Reduction & PCA



Prof. Alexander Ihler

Fall 2023

Dimensionality Reduction

Motivaton

Principal Component Analysis (PCA)

Examples of PCA

Images

Text

Ratings

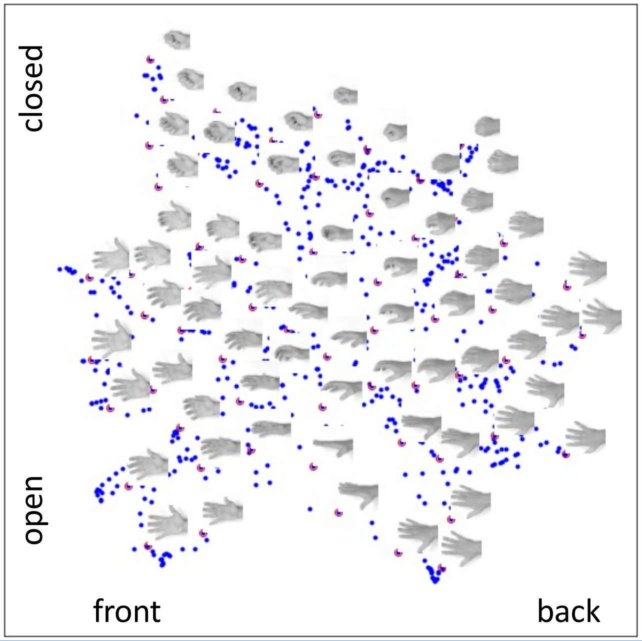
AutoEncoders

Implicit Embeddings

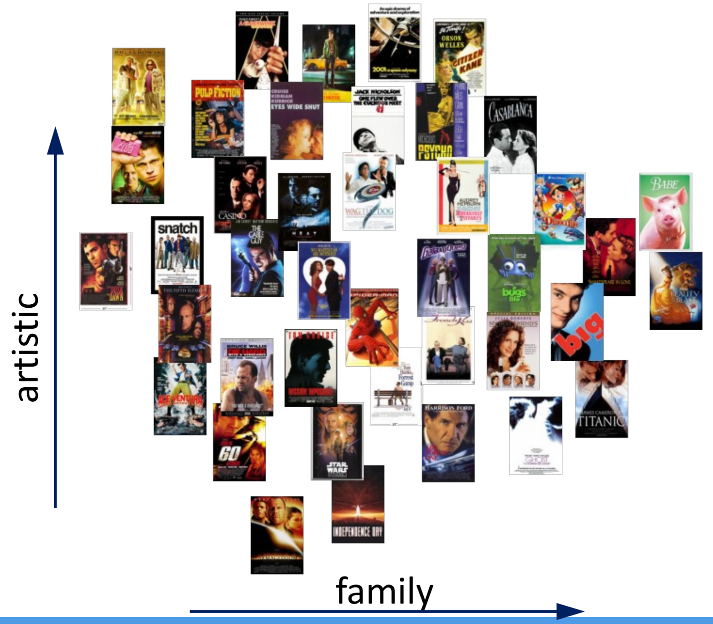
Motivation

- High-dimensional data
  - Images of faces
  - Text from articles
  - All S&P 500 stocks
- Can we describe them in a "simpler" way?
  - Embedding: place data in R^d, such that "similar" data are close

Ex: embedding images in 2D



Ex: embedding movies in 2D

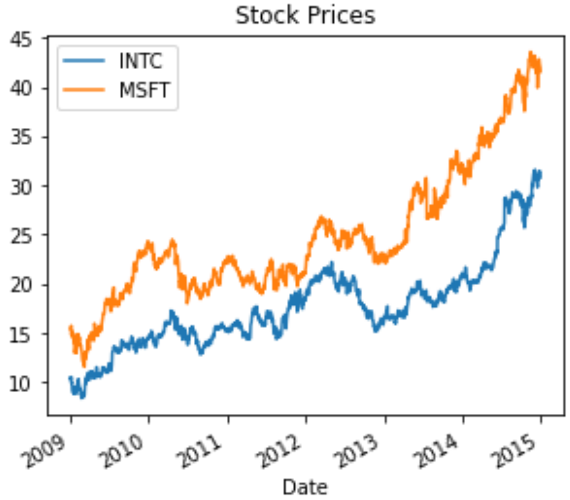


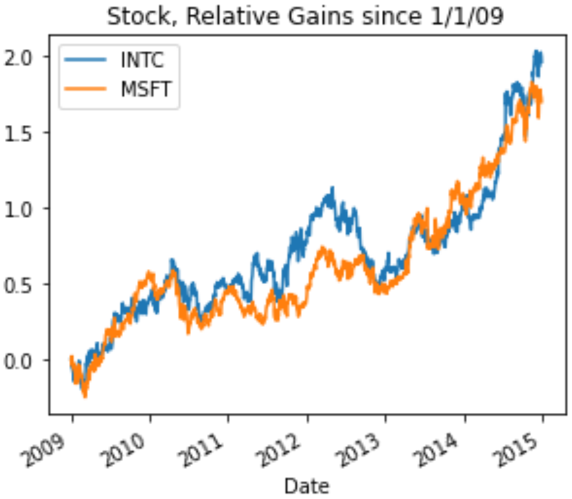
Motivation

- High-dimensional data
  - Images of faces
  - Text from articles
  - All S&P 500 stocks
- Can we describe them in a “simpler” way?
  - Embedding: place data in R^d, such that “similar” data are close
- Ex: S&P 500 – vector of 500 (change in) values per day
  - But, lots of structure
  - Some elements tend to “change together”
  - Maybe we only need a few values to approximate it?
  - “Tech stocks up 2x, manufacturing up 1.5x, …” ?
- How can we access that structure?

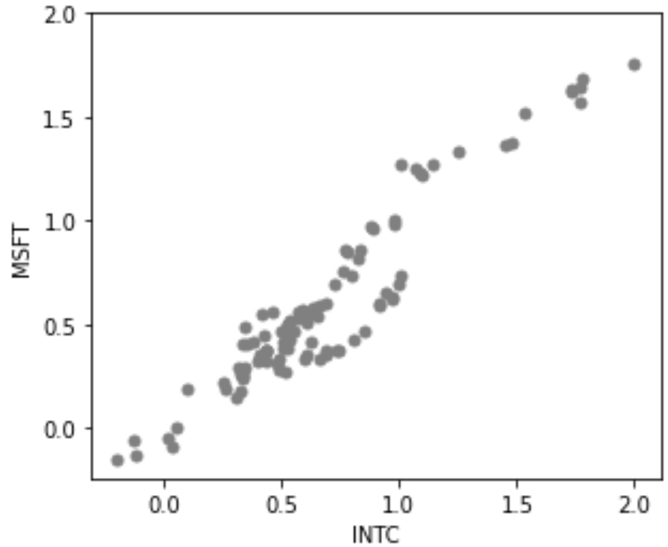
Ex: S&P 2

- Let's look at just two stocks: MSFT, INTC
  - Transform to "relative gains" for convenience





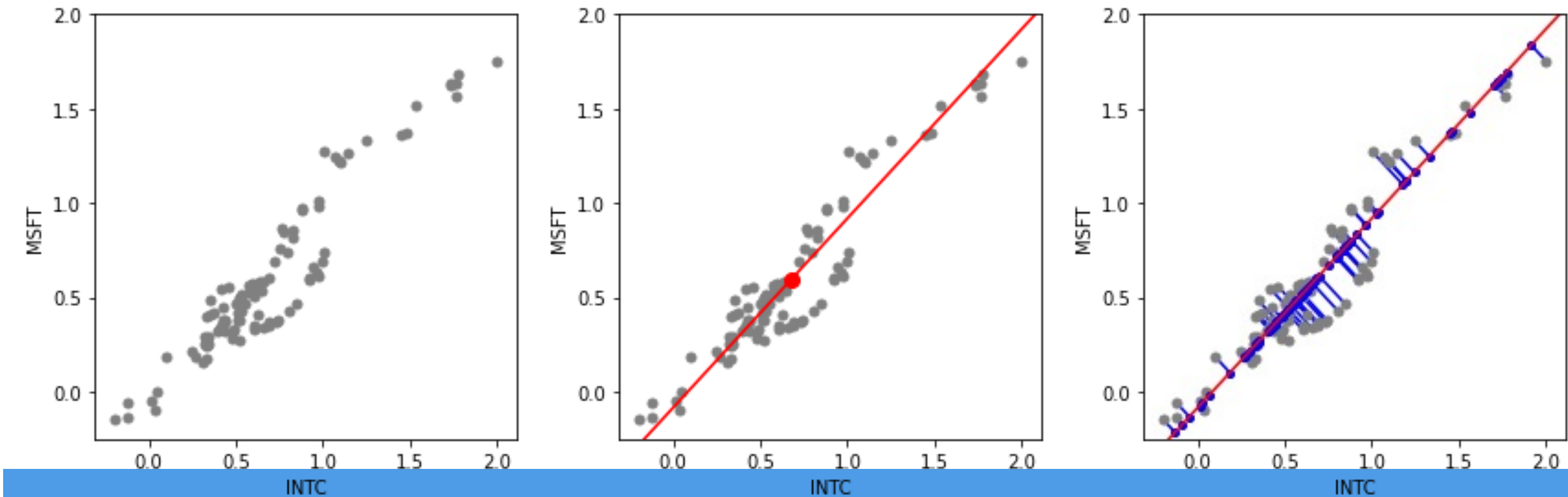
- 100 randomly selected days:
  - Pretty clear correlation structure
  - Can we represent this with just one number, instead of two?



Dimensionality reduction

- Ex: data with two real values $[x_1,x_2]$
- We'd like to describe each point using only one value $[z_1]$
- We'll communicate a "model" to convert: $[x_1,x_2] \sim f(z_1)$

- Ex: linear function f(z): $[x_1,x_2] = \mu + z * \underline{v} = \mu + z * [v_1,v_2]$
- μ, $\underline{v}$ are the same for all data points (communicate once)
- z tells us the closest point on v to the original point $[x_1,x_2]$



Some uses of latent spaces

- Data compression
  - Cheaper, low-dimensional representation

- Noise removal
  - Simple "true" data + noise

- Supervised learning, e.g. regression:
  - Remove colinear / nearly colinear features
  - Reduce feature dimension => combat overfitting

Dimensionality Reduction

Motivaton

Principal Component Analysis (PCA)

Examples of PCA

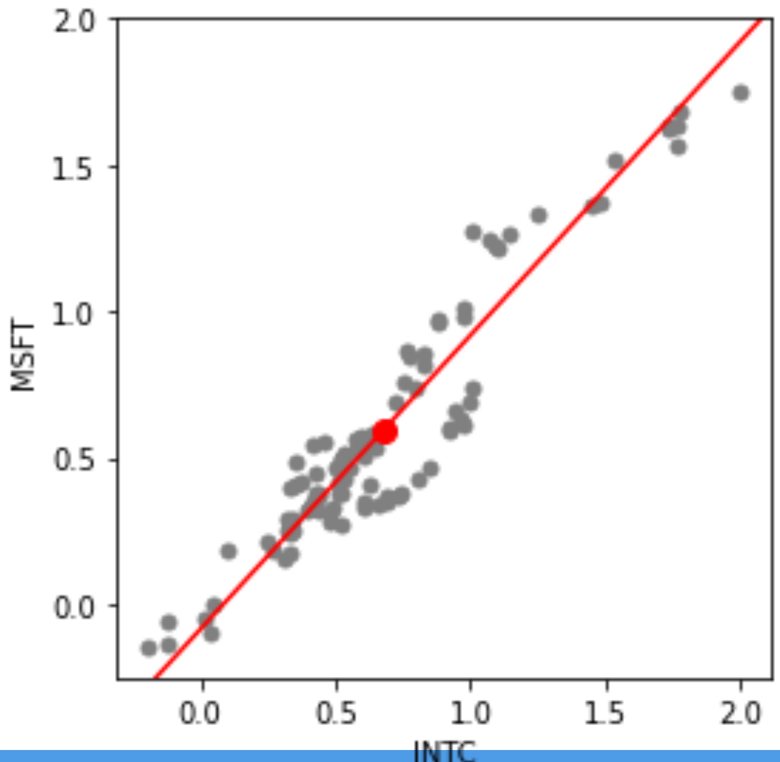
Images	Text	Ratings

AutoEncoders

Implicit Embeddings

Principal Components Analysis

- How should we find v?
  - Assume X is zero mean, or $\tilde{X} = X - \mu$
  - Find "v" as the direction of maximum "spread" (variance)
  - Solution is the eigenvector with largest eigenvalue



Project X to v: $z = \tilde{X} \cdot v$

Variance of projected points:

$$\sum_i (z^{(i)})^2 = z^T z = v^T \, \tilde{X}^T \tilde{X} \, v$$

Best "direction" v:

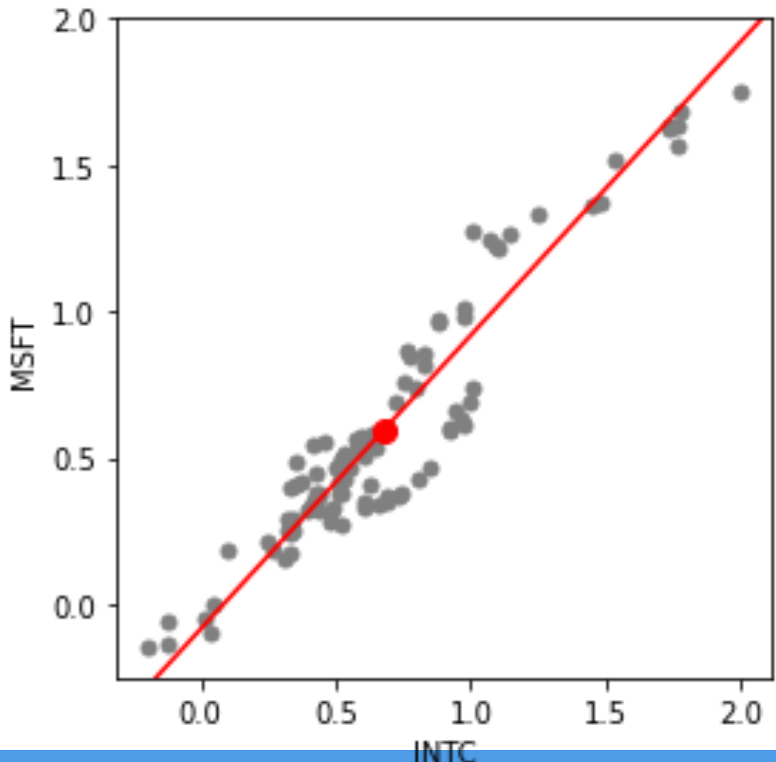
$$\max_v \quad v^T \, \tilde{X}^T \tilde{X} \, v \quad s.t. \; ||v|| = 1$$

$\Rightarrow$ largest eigenvector of $X^T X$

Principal Components Analysis

- How should we find v?
  - Assume X is zero mean, or $\tilde{X} = X - \mu$
  - Find "v" as the direction of maximum "spread" (variance)
  - Solution is the eigenvector with largest eigenvalue

  - Equivalent: v also leaves the smallest residual variance! ("error")



Project X to v: $z = \tilde{X} \cdot v$

Variance of projected points:

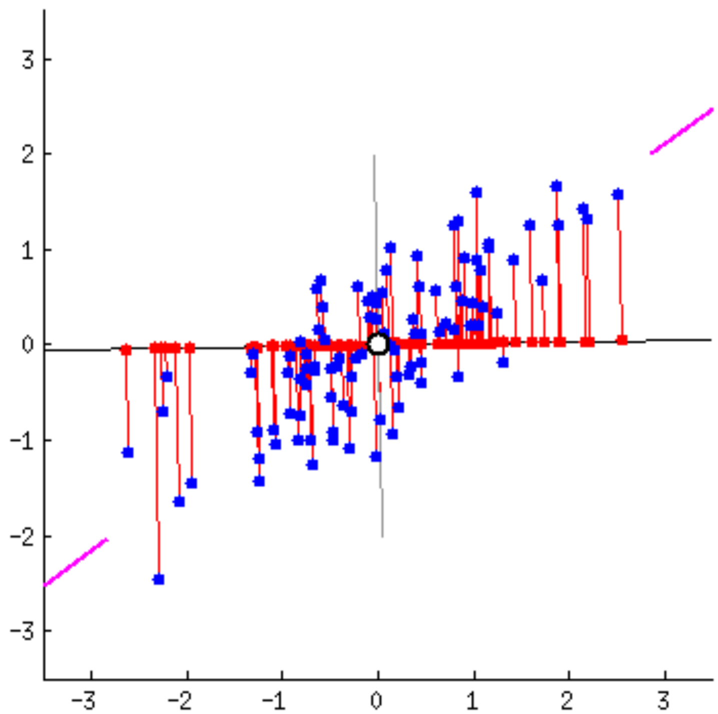
$$\sum_i (z^{(i)})^2 = z^T z = v^T \tilde{X}^T \tilde{X} v$$

Best "direction" v:

$$\max_v \; v^T \tilde{X}^T \tilde{X} v \quad s.t. \; ||v|| = 1$$

$\Rightarrow$ largest eigenvector of $X^T X$

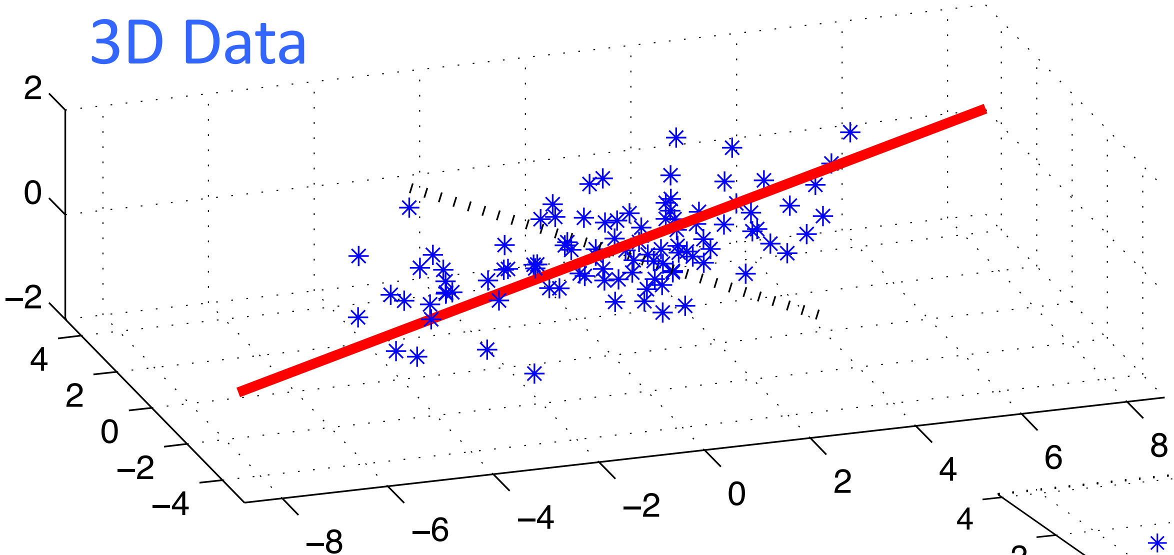
Principal Components Analysis



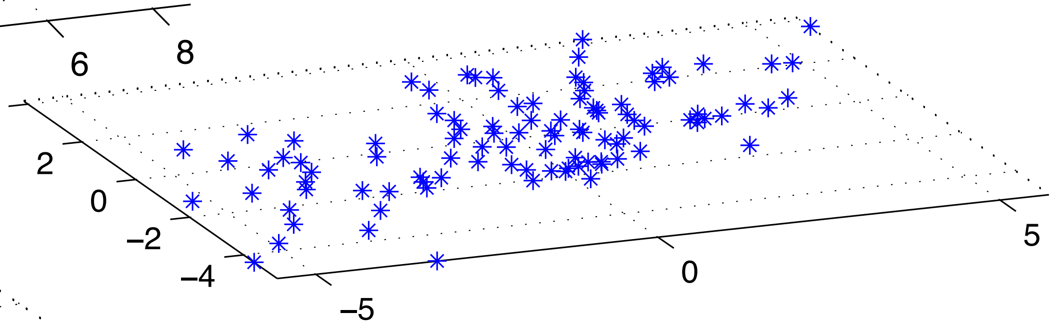
https://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues/140579#140579

Principal Components Analysis (PCA)

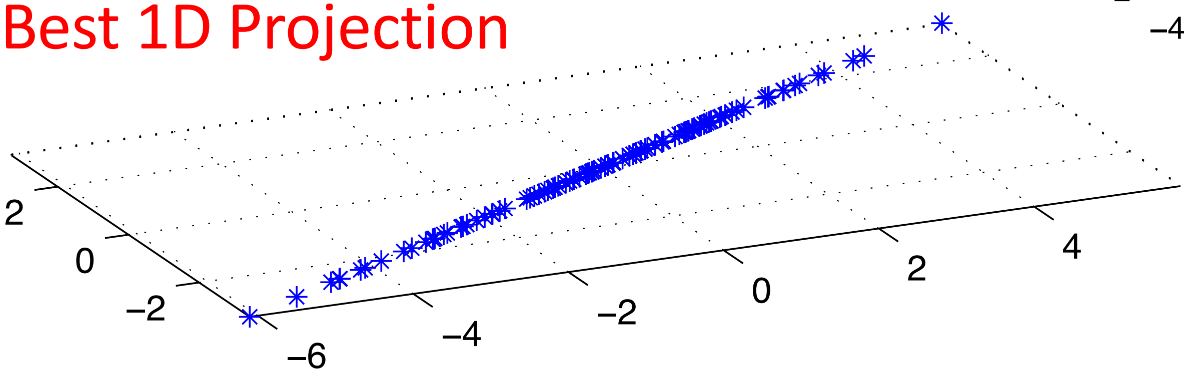
3D Data



Best 2D Projection



Best 1D Projection

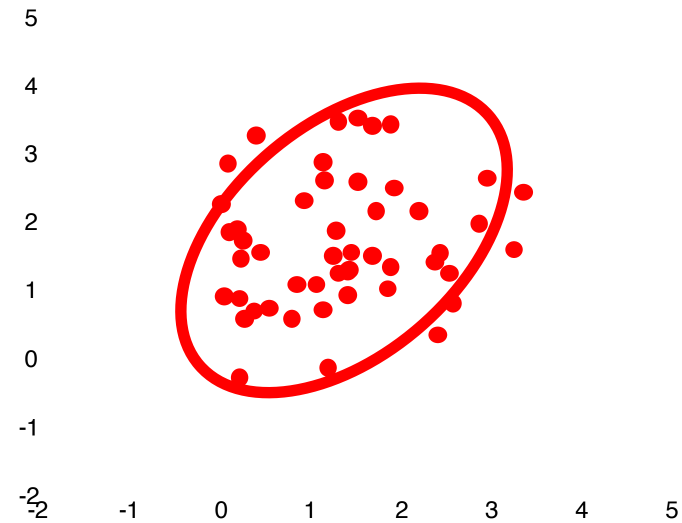


Another interpretation

- Data covariance: $\Sigma = \frac{1}{m}\tilde{X}^T\tilde{X}$ $\quad\quad \tilde{X} = X - \mu$
  - Describes "spread" of the data
  - Draw this with an ellipse
  - Gaussian is

$$p(x) \propto \exp\big(-\frac{1}{2}\Delta^2\big)$$

$$\Delta^2 = (x-\mu)\Sigma^{-1}(x-\mu)^T$$



  - Ellipse shows the contour, Δ^2 = constant

Geometry of the Gaussian

$$\Delta^2 = (\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})$$

Oval shows constant Δ^2 value…

$$\Sigma = U\Lambda U^T$$

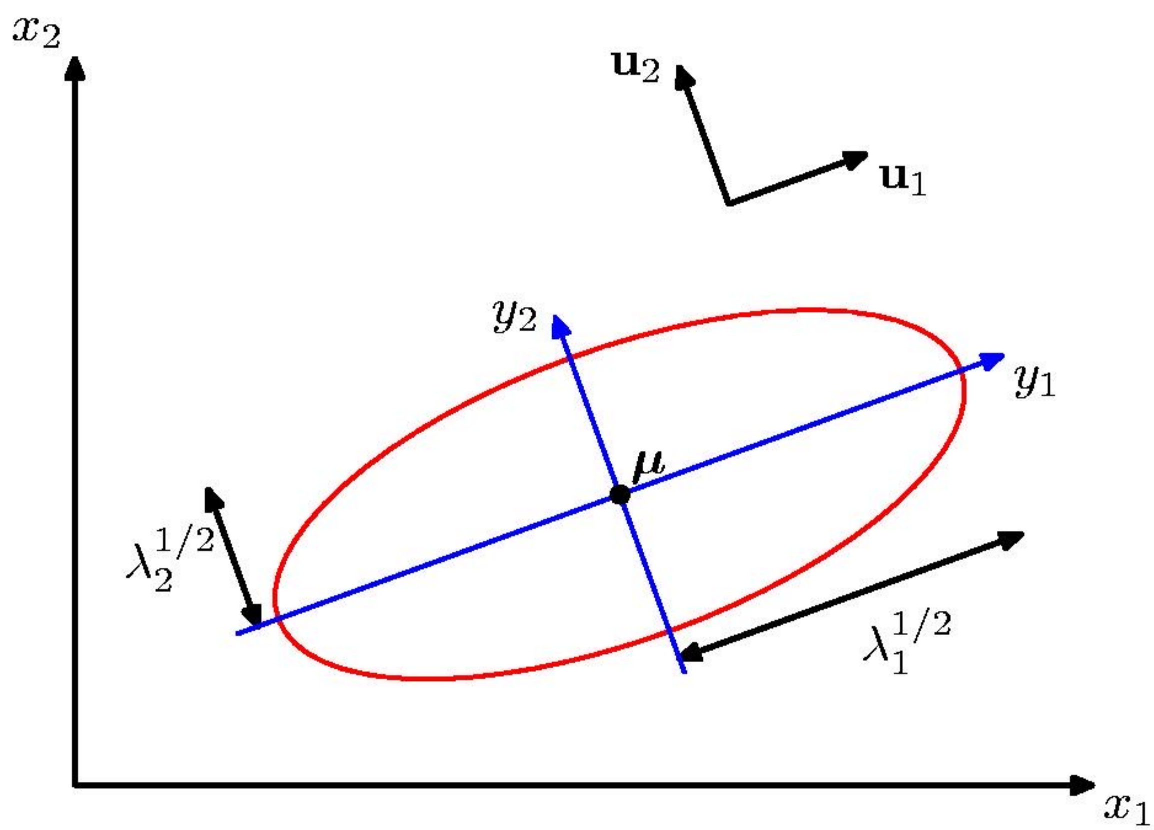
Write Σ in terms of eigenvectors…

$$\boldsymbol{\Sigma}^{-1} = \sum_{i=1}^{D} \frac{1}{\lambda_i}\mathbf{u}_i\mathbf{u}_i^{\mathrm{T}}$$

Then…

$$\Delta^2 = \sum_{i=1}^{D} \frac{y_i^2}{\lambda_i}$$

$$y_i = \mathbf{u}_i^{\mathrm{T}}(\mathbf{x}-\boldsymbol{\mu})$$



PCA representation

- Subtract data mean from each point
- (Typically) scale each dimension by its variance
  - Helps pay less attention to magnitude of the variable
- Compute covariance matrix, $S = 1/m \sum (x^i-\mu)' \; (x^i-\mu)$
- Compute the k largest eigenvectors of S

$$S = V\ D\ V^\wedge T$$

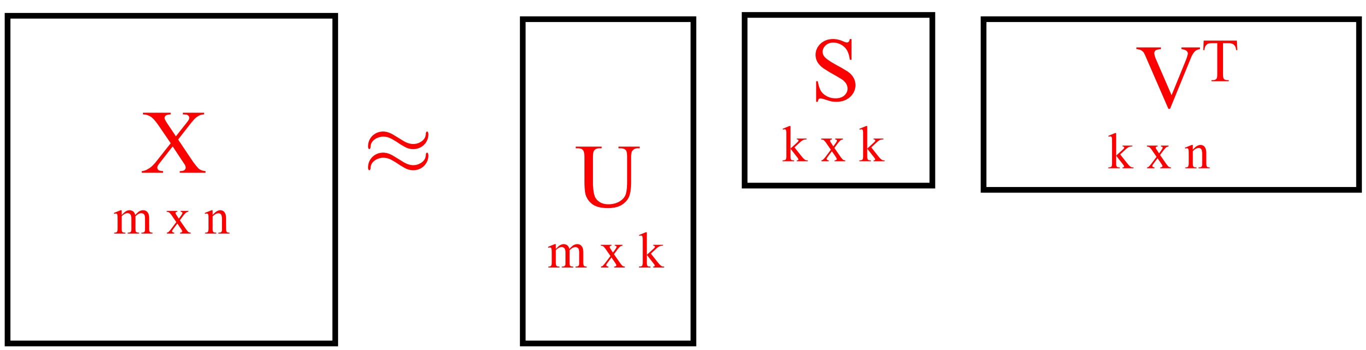
```
mu = np.mean( X, axis=0, keepdims=True )   # find mean over data points
X0 = X - mu                                # zero-center the data
S = X0.T @ X0  / m                         # S = np.cov( X.T ), data covariance
D,V = np.linalg.eig( S )                   # find eigenvalues/vectors: can be slow!
pi = np.argsort(D)[::-1]                   # sort eigenvalues largest to smallest
D,V = D[pi], V[:,pi]                       #
D,V = D[0:k], V[:,0:k]                     # and keep the k largest
```

Singular Value Decomposition

- Alternative method to calculate (still subtract mean 1st)
- Decompose $X = U\ S\ V^T$
  - Orthogonal: $X^T X = V\ S\ S\ V^T = V\ D\ V^T$
  - $X X^T = U\ S\ S\ U^T = U\ D\ U^T$

- U*S matrix provides coefficients
  - Example $x_i = U_{i,1}\ S_{11}\ v_1 + U_{i,2}\ S_{22}\ v_2 + \ldots$

- Gives the least-squares approximation to X of this form



SVD for PCA

- Subtract data mean from each point
- (Typically) scale each dimension by its variance
  - Helps pay less attention to magnitude of the variable
- Compute the SVD of the data matrix

```
mu = np.mean( X, axis=0, keepdims=True )   # find mean over data points
X0 = X - mu                                 # zero-center the data

U,S,Vh = scipy.linalg.svd(X0, False)        # X0 = U * diag(S) * Vh

Xhat = U[:,0:k].dot( np.diag(S[0:k]) ).dot( Vh[0:k,:] )   # approx using k largest eigendir
```

Dimensionality Reduction

- Motivaton
- Principal Component Analysis (PCA)
- **Examples of PCA**
  - **Images**
  - Text
  - Ratings
- AutoEncoders
- Implicit Embeddings

Applications of SVD

- "Eigen-faces"
  - Represent image data (faces) using PCA

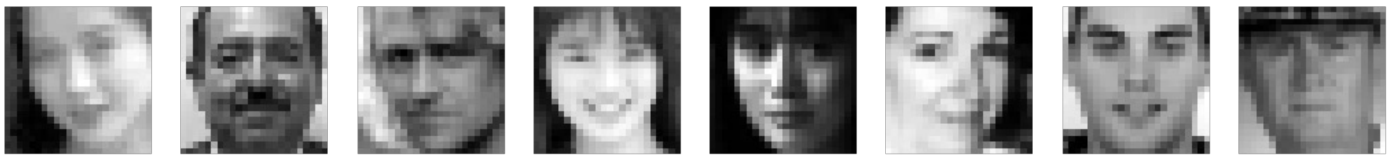
- LSI / "topic models"
  - Represent text data (bag of words) using PCA

- Collaborative filtering
  - Represent rating data matrix using PCA

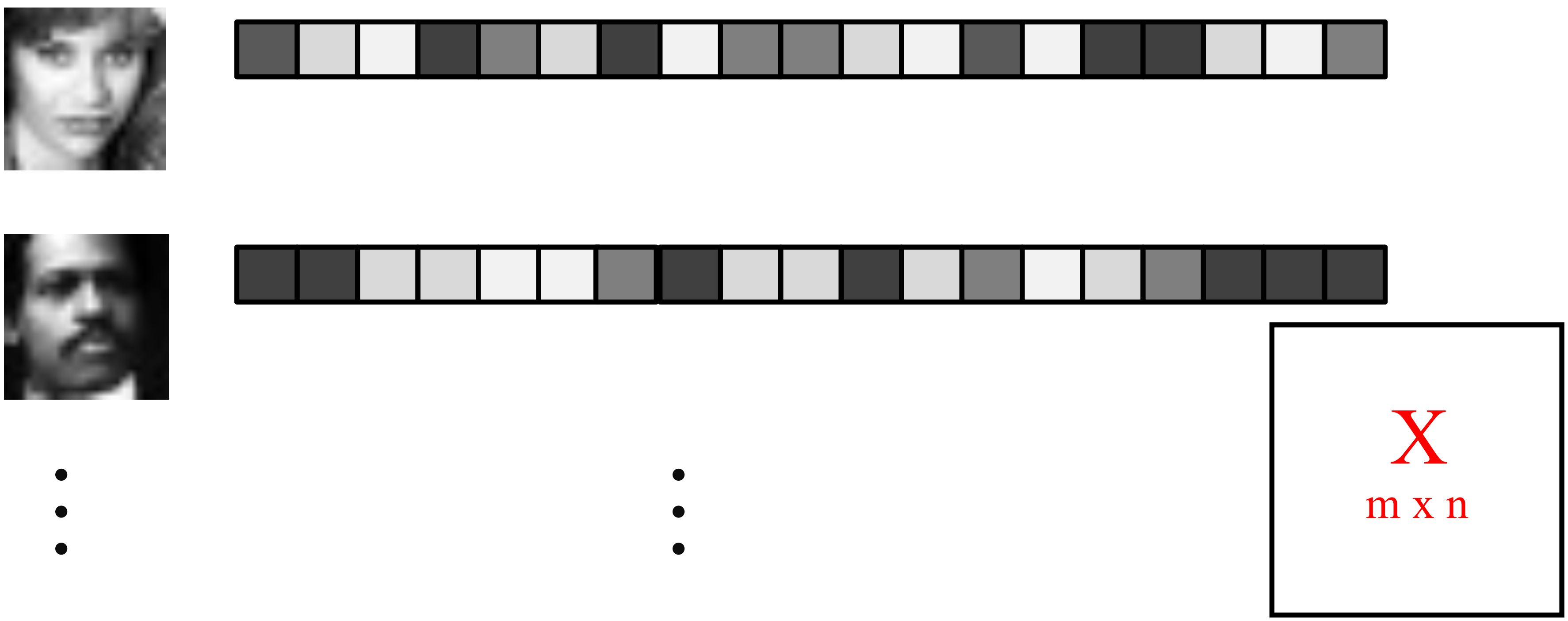
and more…

"Eigen-faces"

- "Eigen-X" = represent X using PCA
- Ex: Viola Jones data set
  - 24x24 images of faces = 576 dimensional measurements

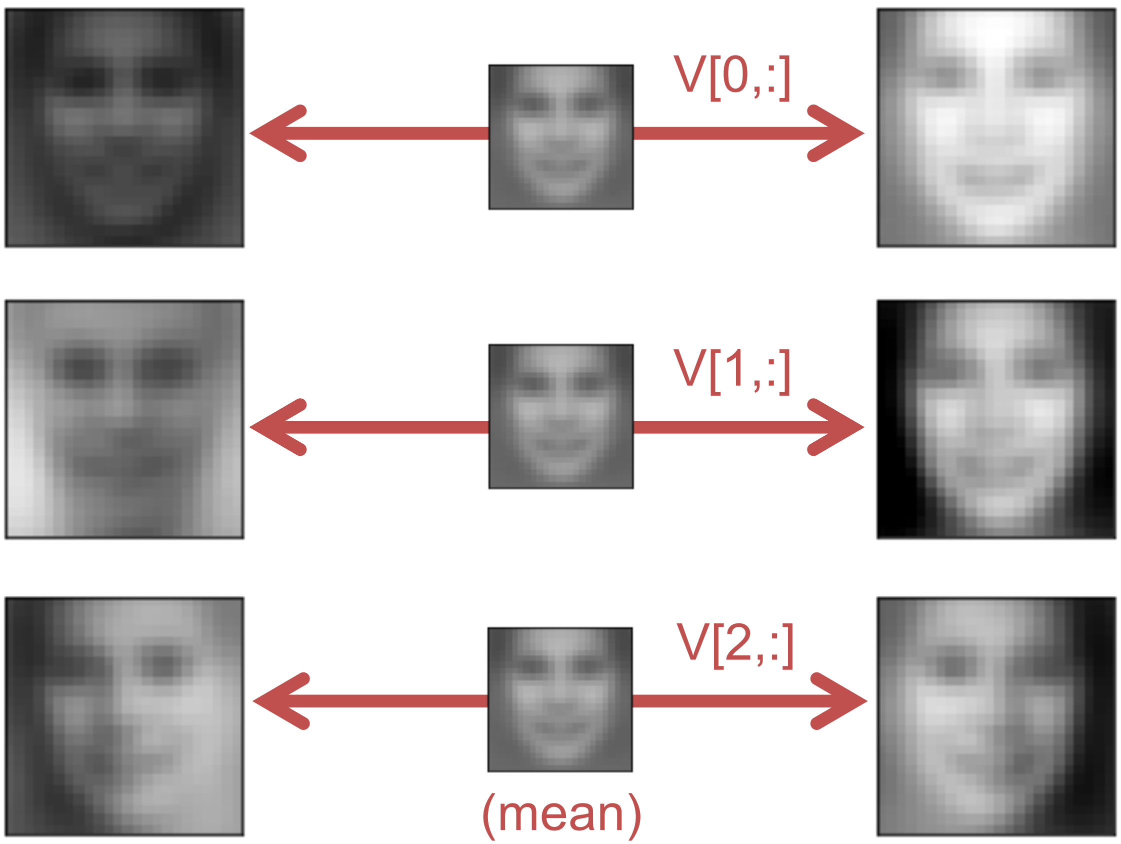


. . .



"Eigen-faces"

- "Eigen-X" = represent X using PCA
- Ex: Viola Jones data set
  - 24x24 images of faces = 576 dimensional measurements
  - Keep first K PCA components



First three components are all "lighting" – b/c lighting affects the value of every pixel!

$$\underset{m \times n}{X} \approx \underset{m \times k}{U} \; \underset{k \times k}{S} \; \underset{k \times n}{V^T}$$

"Eigen-faces"

- "Eigen-X" = represent X using PCA
- Ex: Viola Jones data set
  - 24x24 images of faces = 576 dimensional measurements
  - Keep first K PCA components

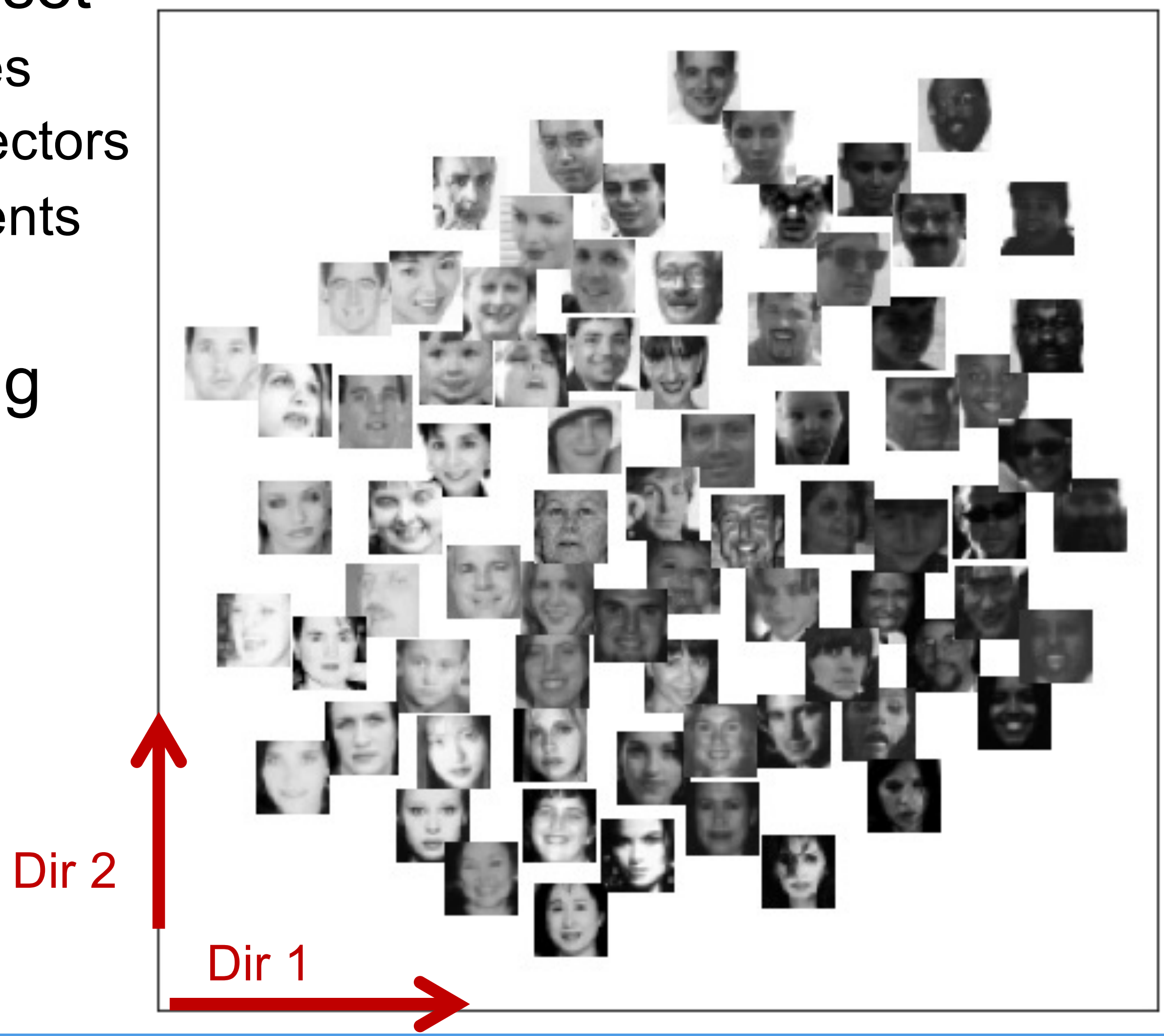
Mean Dir 1 Dir 2 Dir 3 Dir 4 …

Projecting data onto first k dimensions

"Eigen-faces"

- "Eigen-X" = represent X using PCA
- Ex: Viola Jones data set
  - 24x24 images of faces
    = 576 dimension vectors
  - Keep first K components
- Can lay out data using
  latent coordinates:

Projecting data
onto first 2
dimensions



Dimensionality Reduction

- Motivaton
- Principal Component Analysis (PCA)
- **Examples of PCA**
  - Images
  - **Text**
  - Ratings
- AutoEncoders
- Implicit Embeddings

Text representations

- "Bag of words"
  - Remember word counts but not order
- Example:

Rain and chilly weather didn't keep thousands of paradegoers from camping out Friday night for the 111th Tournament of Roses.

Spirits were high among the street party crowd as they set up for curbside seats for today's parade.

``I want to party all night," said Tyne Gaudielle, 15, of Glendale, who spent the last night of the year along Colo Boulevard with a group of friends.

Whether they came for the partying or the parade, campe in for a long night. Rain continued into the evening and temperatures were expected to dip down into the low 40s

VOCABULARY:
0001 ability
0002 able
0003 accept
0004 accord
…
1393 parade
…
1402 party
…

WORD COUNTS:	
0137 asked	1
0185 begin	1
….	
1393 parade	9
1402 party	2
1408 past	2
….	

PCA on BoW text data =
Latent Semantic Indexing ("LSA")

Ex. from [Landauer et al., 1998]

Ex: LSA on Paper Titles

Documents

c1: Human machine interface for ABC comput
c2: A survey of user opinion of computer syste
c3: The EPS user interface management syster
c4: System and human system engineering tes
c5: Relation of user perceived response time t

m1: The generation of random, binary, ordere
m2: The intersection graph of paths in trees
m3: Graph minors IV: Widths of trees and wel
m4: Graph minors: A survey

Vocabulary

human	system	survey
interface	response	trees
computer	time	graph
user	eps	minors

Data Matrix X

$x_j^{(i)}$: word j appears in doc i

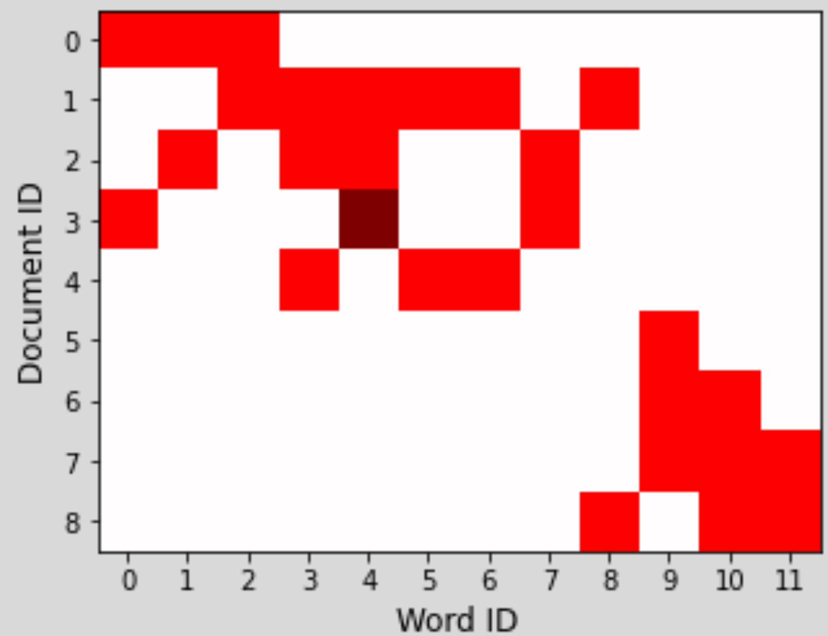
Huge matrix (sparse: mostly zeros)

Typically:

(1) Don't subtract the mean, etc.

(2) Normalize rows to sum to one

(controls for short vs long documents)



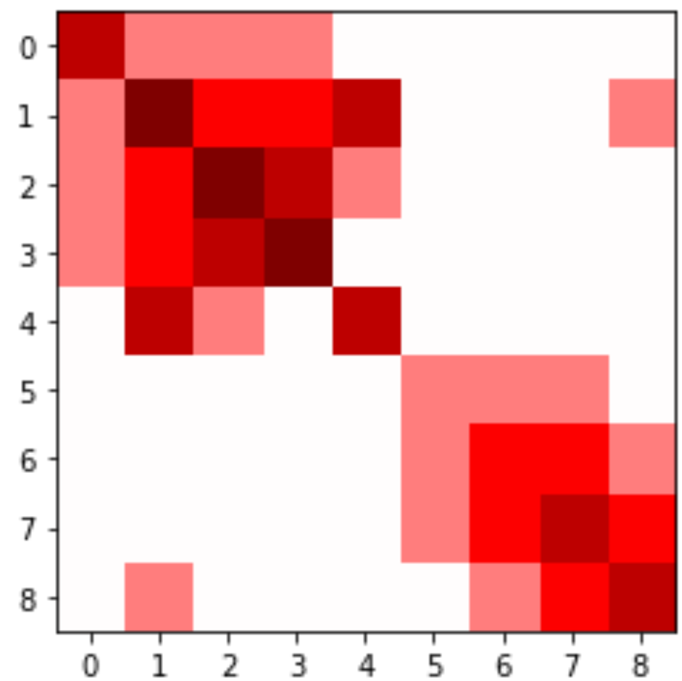
Matrices are big, but data is sparse

- Typical example:
  - Number of docs, $D \sim 10^6$
  - Number of unique words in vocab, $W \sim 10^5$
  - FULL Storage required $\sim 10^{11}$
  - Sparse Storage required $\sim 10^8$

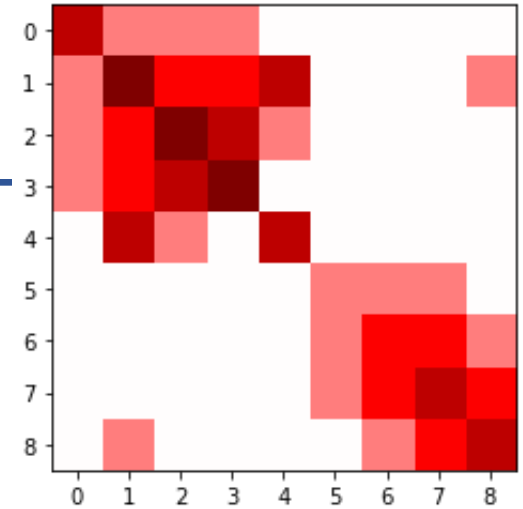
- D x W matrix (# docs x # words)
  - Each entry is non-negative
  - Typically integer / count data

Ex: LSA on Paper Titles

- Measure document similarity using word overlap
  - Corresponds to dot product between documents
- We can see the structure of the document corpus in the similarity matrix $X \cdot X^T$
  - Block structure
  - HCI papers broadly similar
  - Math papers broadly similar



Ex: LSA on Paper Titles



- Why use PCA?
  - Text overlap: noisy measure of similarity
  - Features are very sparse…
- Ex:

c2: A survey of user opinion of computer system response time

m4: Graph minors: A survey

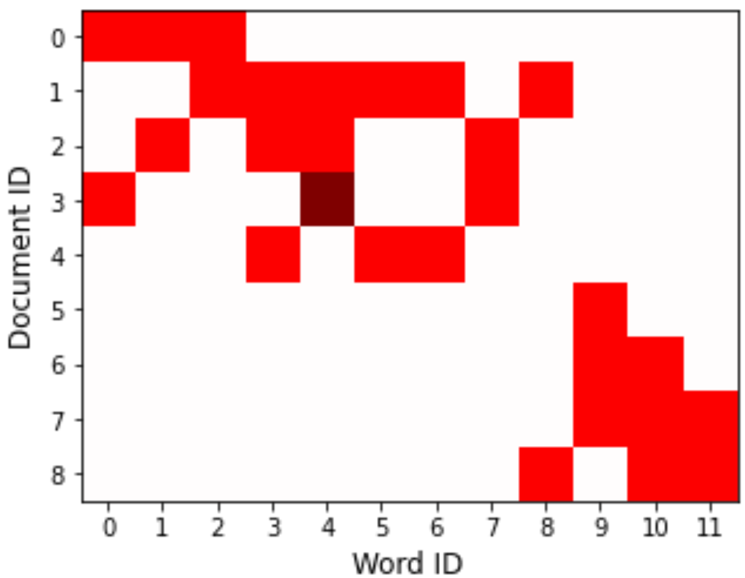
c1: Human machine interface for ABC computer applications

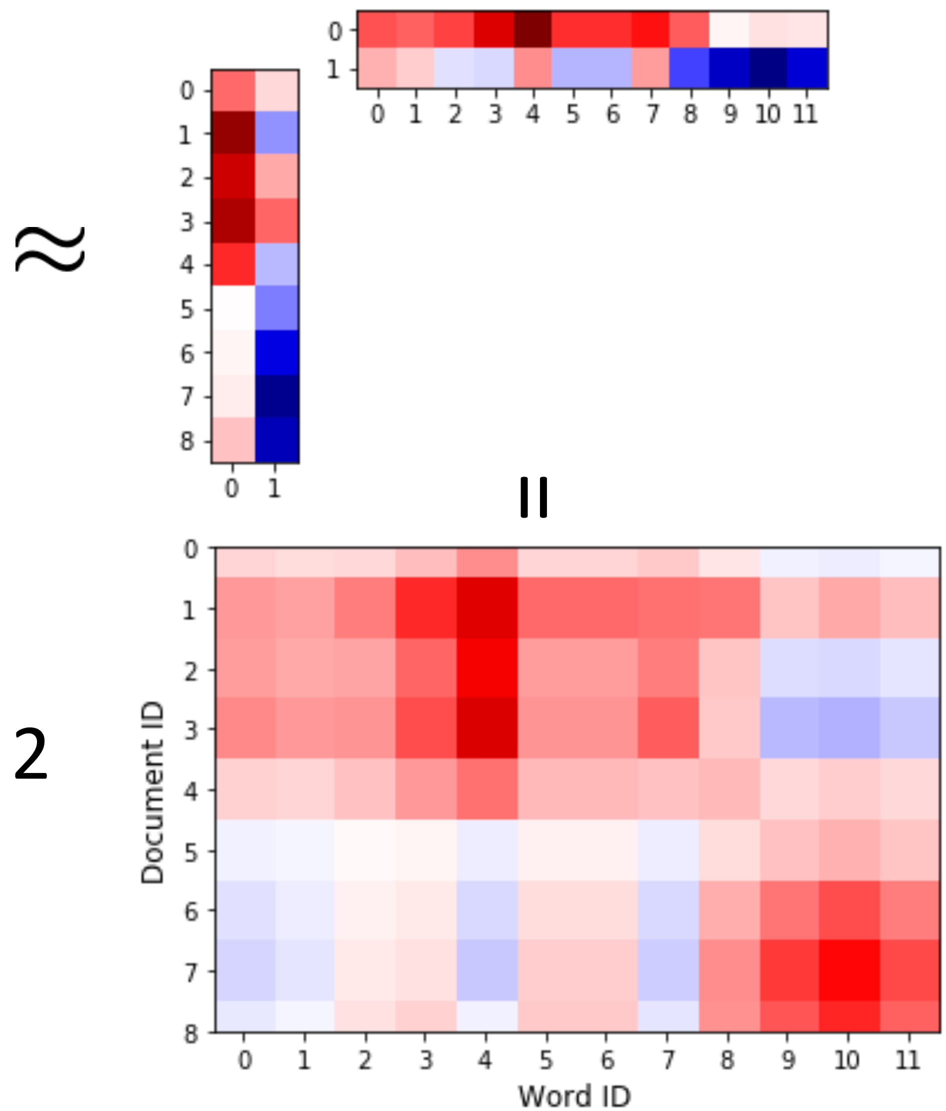
Same "distance"!

- Using PCA first removes some of this "noise"
  - Useful for document comparison, "fuzzy" search
  - Hopefully closer to "concept comparision" than word overlap

Ex: LSA on Paper Titles

- Take PCA of the word-doc count matrix

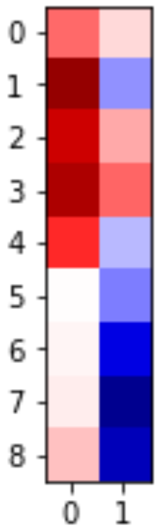


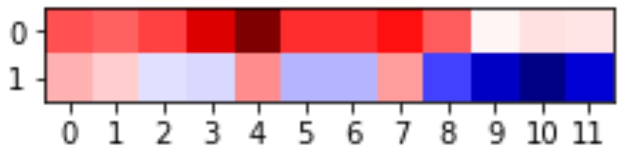


- Approximate X with rank 2

Ex: LSA on Paper Titles

- Interpreting the latent dimensions



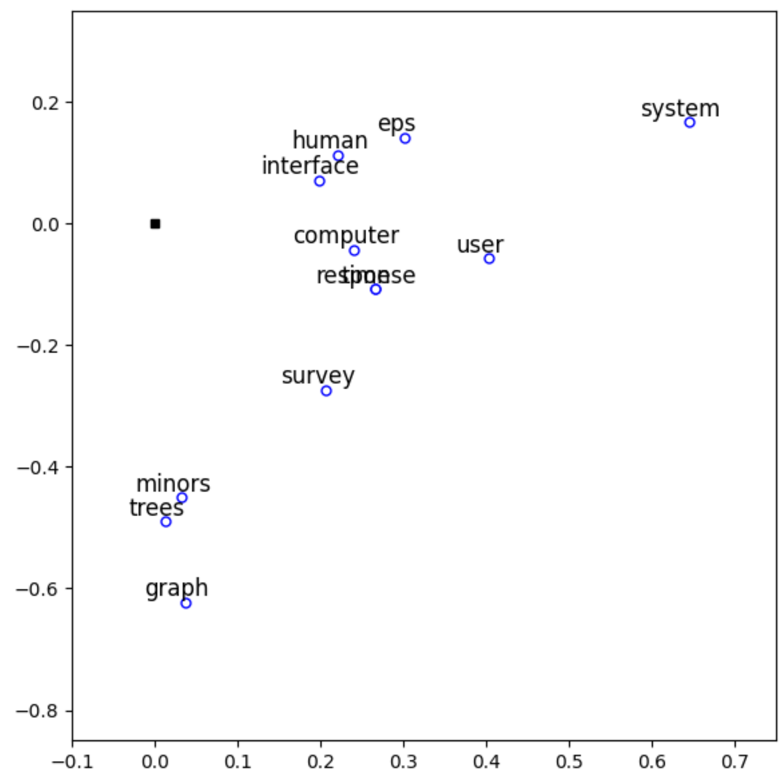


Dimension 1:
  words 0..8 positive (“human”, ”interface”…)

Dimension 2:
  words 9..11 negative (“tree”, “graph”…)

Docs 0..4:
  positive first dimension
  (word set 1 appears often)

Docs 5..8:
  negative 2nd dimension
  (word set 2 appears often)



Latent Semantic Analysis (LSA)

- What do the principal components look like?

PRINCIPAL COMPONENT 1

0.135 genetic
0.134 gene
0.131 snp
0.129 disease
0.126 genome_wide
0.117 cell
0.110 variant
0.109 risk
0.098 population
0.097 analysis
0.094 expression
0.093 gene_expression
0.092 gwas

PRINCIPAL COMPONENT 2

0.247 snp
-0.196 cell
0.187 variant
0.181 risk
0.180 gwas
0.162 population
0.162 genome_wide
0.155 genetic
0.130 loci
-0.116 mir
-0.116 expression
0.113 allele
0.108 schizophrenia

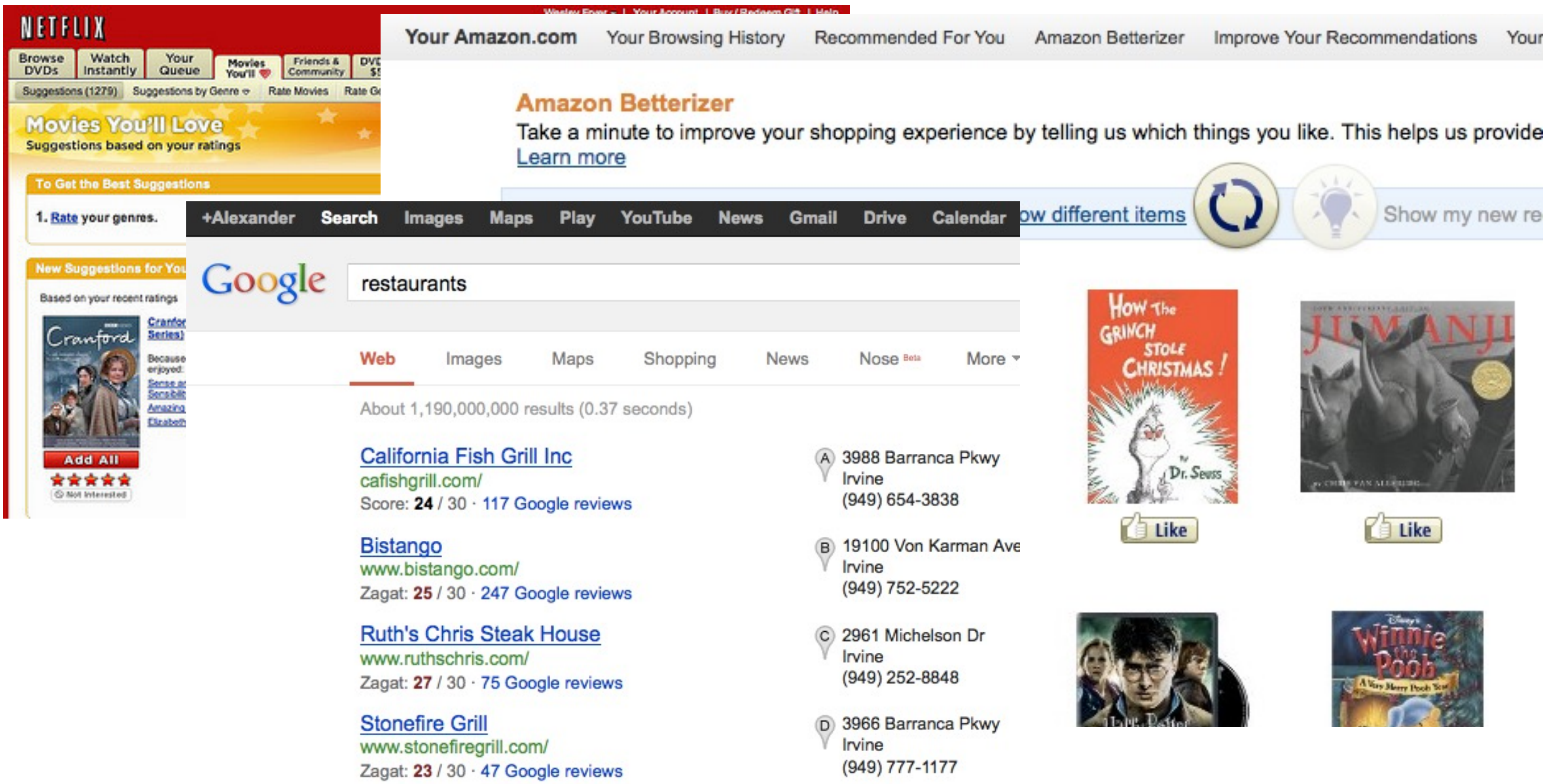
Q: But what does "-0.196 cell" mean?

May want to use **nonnegative** matrix factorization – like PCA, but all entries positive

Dimensionality Reduction

- Motivaton
- Principal Component Analysis (PCA)
- **Examples of PCA**
  - Images
  - Text
  - **Ratings**
- AutoEncoders
- Implicit Embeddings

Examples


NETFLIX
Browse DVDs
Watch Instantly
Your Queue
Movies You'll ♥
Friends & Community
Suggestions (1279)
Suggestions by Genre
Rate Movies
Movies You'll Love
Suggestions based on your ratings
To Get the Best Suggestions
1. Rate your genres.
New Suggestions for You
Based on your recent ratings
Add All
Your Amazon.com
Your Browsing History
Recommended For You
Amazon Betterizer
Improve Your Recommendations
Amazon Betterizer
Take a minute to improve your shopping experience by telling us which things you like. This helps us provide
Learn more
ow different items
Show my new re
Like
Like
+Alexander
Search
Images
Maps
Play
YouTube
News
Gmail
Drive
Calendar
Google
restaurants
Web
Images
Maps
Shopping
News
Nose Beta
More
About 1,190,000,000 results (0.37 seconds)
California Fish Grill Inc
cafishgrill.com/
Score: 24 / 30 · 117 Google reviews
A 3988 Barranca Pkwy
Irvine
(949) 654-3838
Bistango
www.bistango.com/
Zagat: 25 / 30 · 247 Google reviews
B 19100 Von Karman Ave
Irvine
(949) 752-5222
Ruth's Chris Steak House
www.ruthschris.com/
Zagat: 27 / 30 · 75 Google reviews
C 2961 Michelson Dr
Irvine
(949) 252-8848
Stonefire Grill
www.stonefiregrill.com/
Zagat: 23 / 30 · 47 Google reviews
D 3966 Barranca Pkwy
Irvine
(949) 777-1177

Collaborative filtering (Netflix)

From Y. Koren of BellKor team

users

movies	1	2	3	4	5	6	7	8	9	10	11	12
1	1		3		?	5			5		4	
2			5	4			4			2	1	3
3	2	4		1	2		3		4	3	5	
4		2	4		5			4			2	
5			4	3	4	2					2	5
6	1		3		3			2			4	

Latent space models

From Y. Koren of BellKor team

- Model ratings matrix as combination of user and movie factors
- Infer values from known ratings
- Extrapolate to unranked

users

items												
	1		3			5			5		4	
			5	4			4			2	1	3
	2	4		1	2		3		4	3	5	
		2	4		5			4			2	
			4	3	4	2					2	5
	1		3		3			2			4	

~

~ items

.1	-.4	.2
-.5	.6	.5
-.2	.3	.5
1.1	2.1	.3
-.7	2.1	-2
-1	.7	.3

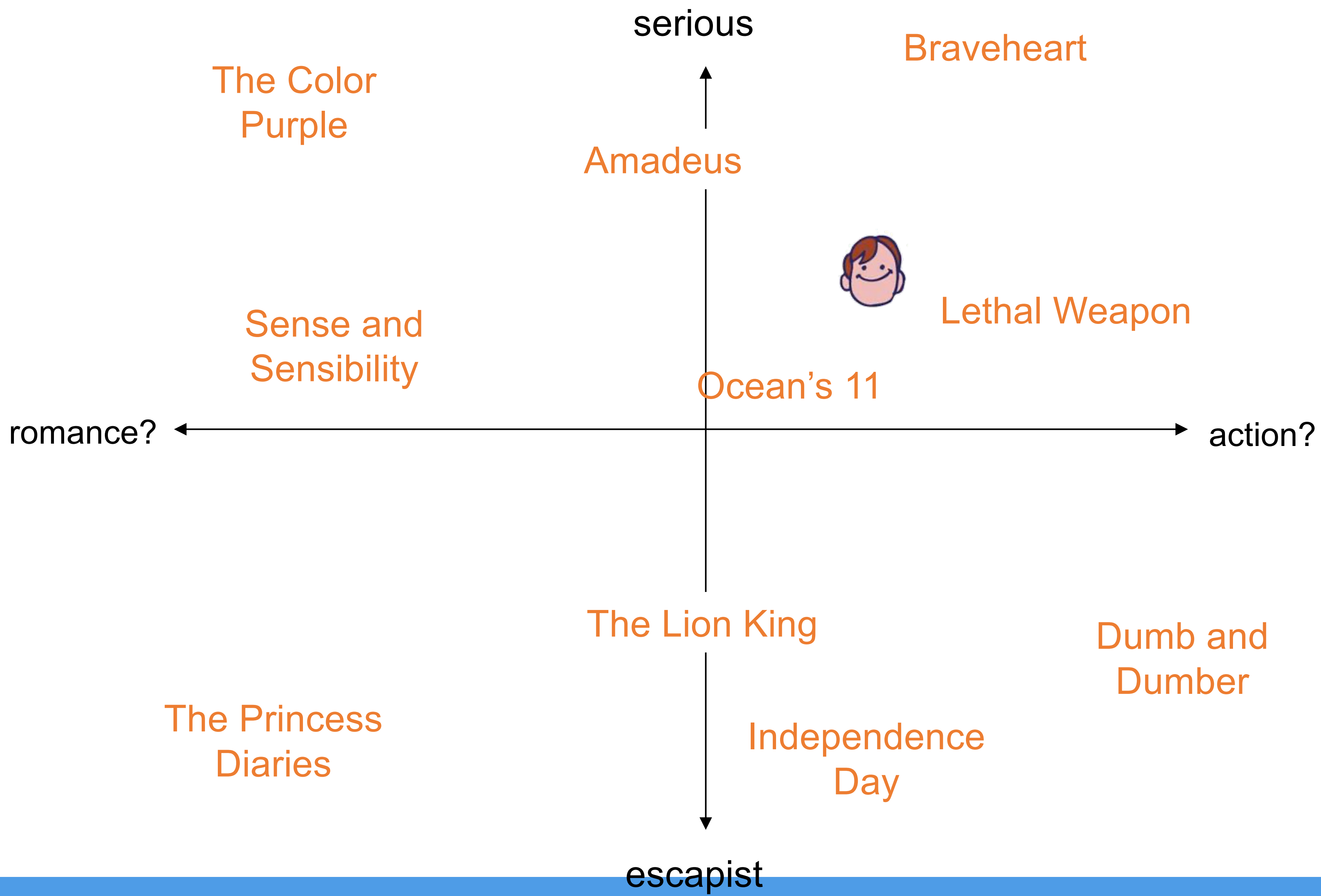
●

users

1.1	-.2	.3	.5	-2	-.5	.8	-.4	.3	1.4	2.4	-.9
-.8	.7	.5	1.4	.3	-1	1.4	2.9	-.7	1.2	-.1	1.3
2.1	-.4	.6	1.7	2.4	.9	-.3	.4	.8	.7	-.6	.1

Latent space models

From Y. Koren
of BellKor team



Some SVD dimensions

See timelydevelopment.com

Dimension 1

Offbeat / Dark-Comedy	Mass-Market / 'Beniffer' Movies
Lost in Translation	Pearl Harbor
The Royal Tenenbaums	Armageddon
Dogville	The Wedding Planner
Eternal Sunshine of the Spotless Mind	Coyote Ugly
Punch-Drunk Love	Miss Congeniality

Dimension 2

Good	Twisted
VeggieTales: Bible Heroes: Lions	The Saddest Music in the World
The Best of Friends: Season 3	Wake Up
Felicity: Season 2	I Heart Huckabees
Friends: Season 4	Freddy Got Fingered
Friends: Season 5	House of 1

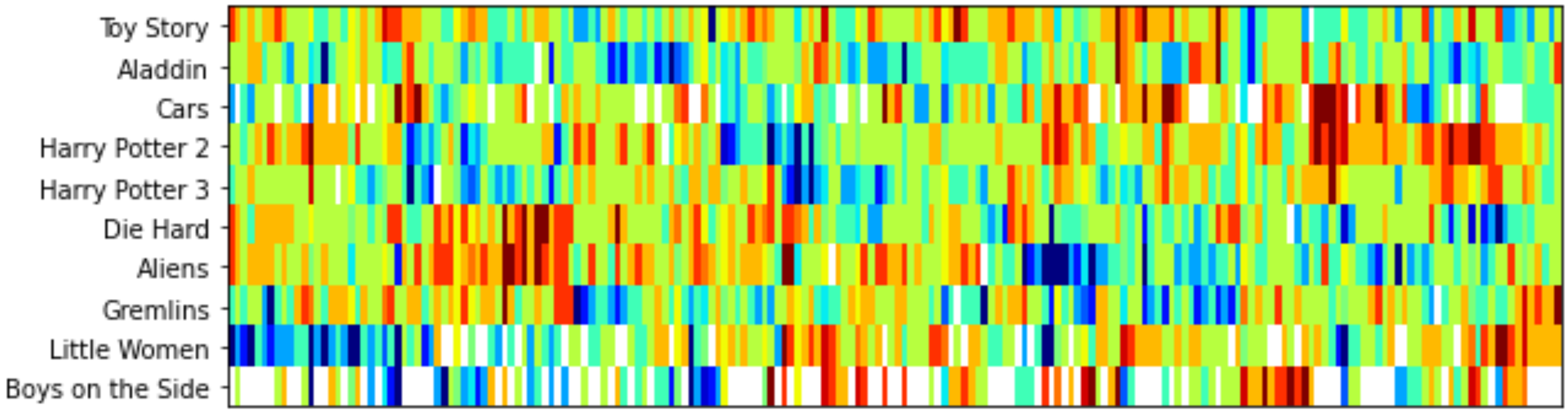
Dimension 3

What a 10 year old boy would watch	What a liberal woman would watch
Dragon Ball Z: Vol. 17: Super Saiyan	Fahrenheit 9/11
Battle Athletes Victory: Vol. 4: Spaceward Ho!	The Hours
Battle Athletes Victory: Vol. 5: No Looking Back	Going Upriver: The Long War of John Kerry
Battle Athletes Victory: Vol. 7: The Last Dance	Sex and the City: Season 2
Battle Athletes Victory: Vol. 2: Doubt and Conflic	Bowling for Columbine

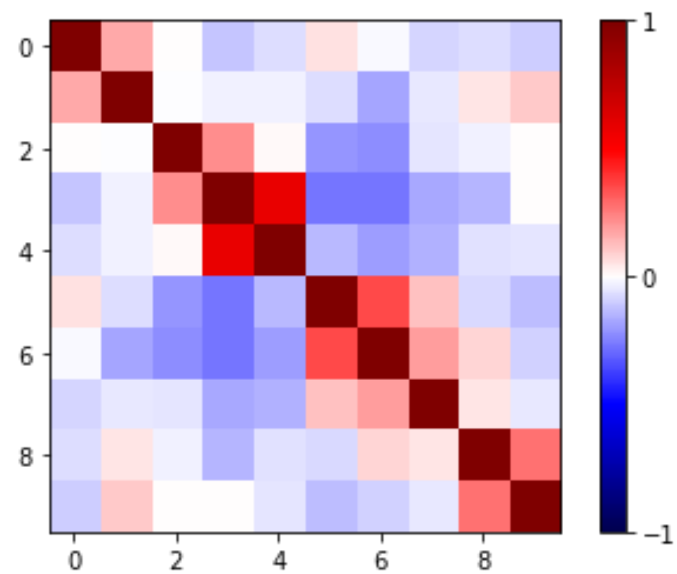
Ex: MovieLens Toy

- 10 movies, 200 users

(Sorted using clustering…)

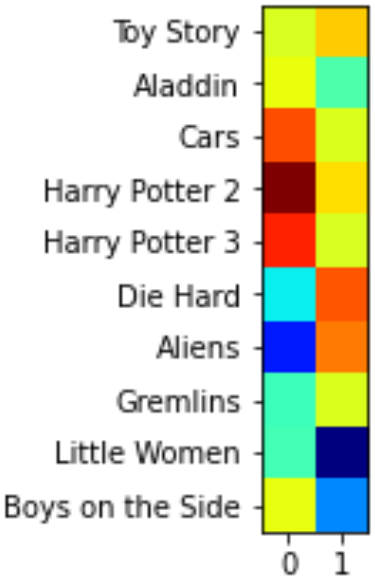


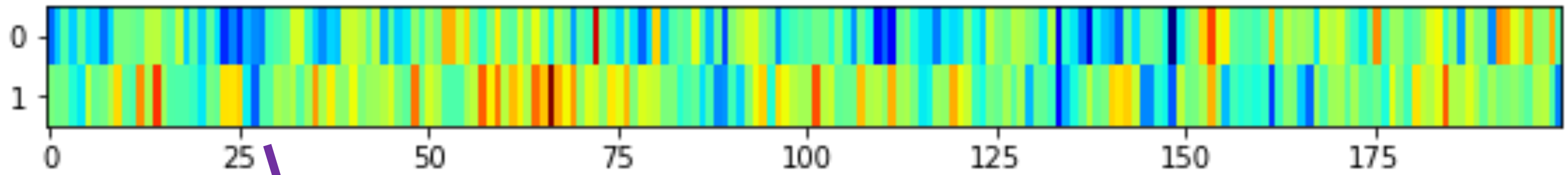
Rating similarity



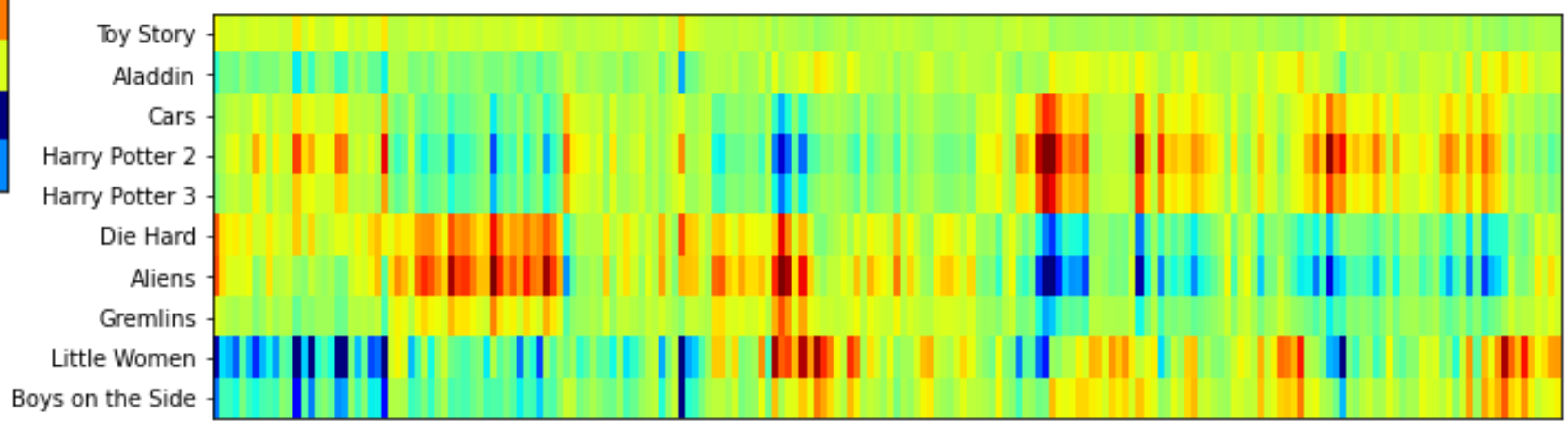
Ex: MovieLens Toy

- Use PCA on the ratings matrix
  - Keep, e.g., 2 principal components



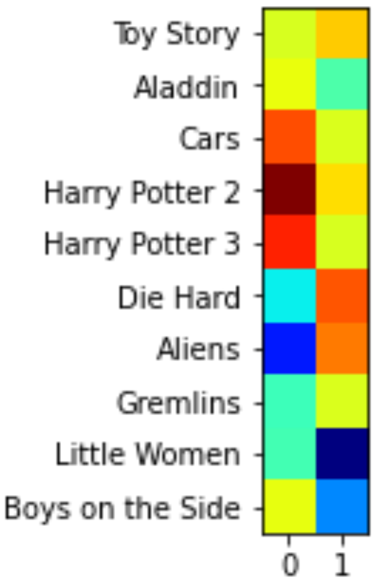


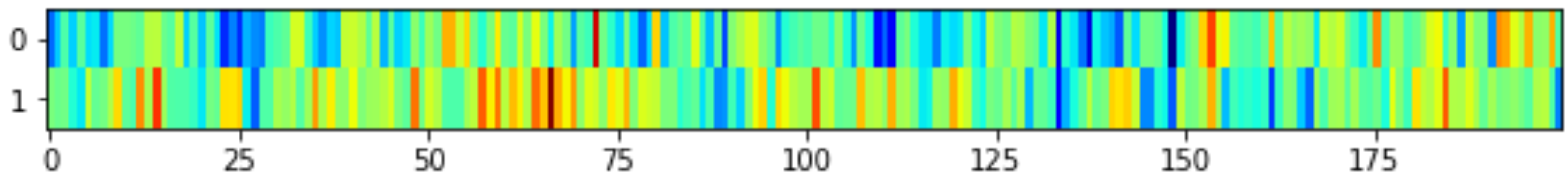
Approximates $x^{(i)}_j$ by U_j @S@ V_i



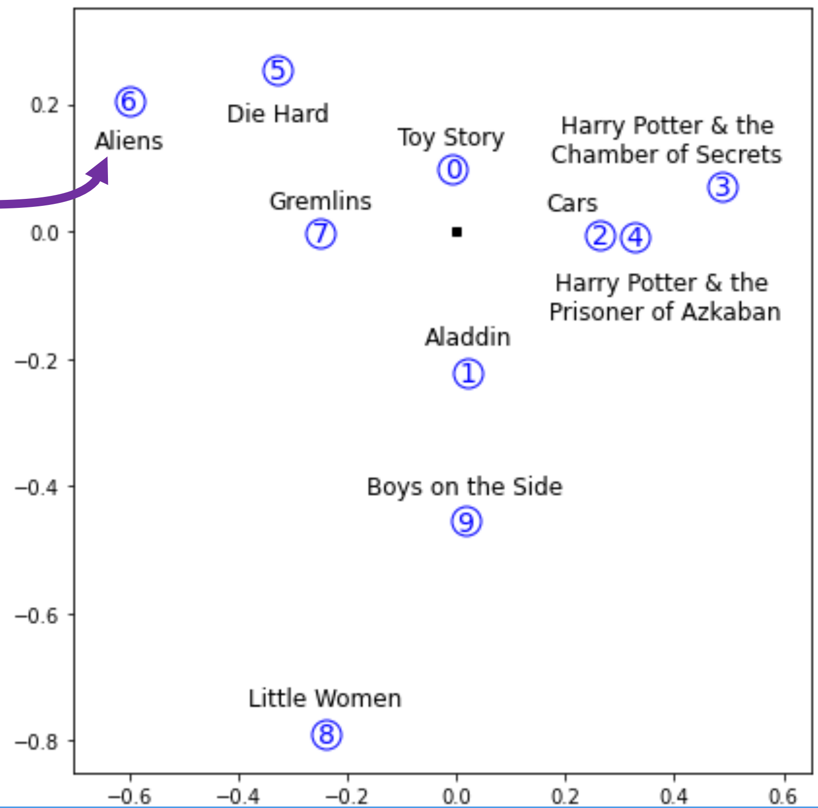
Ex: MovieLens Toy

- Use PCA on the ratings matrix
  - Keep, e.g., 2 principal components





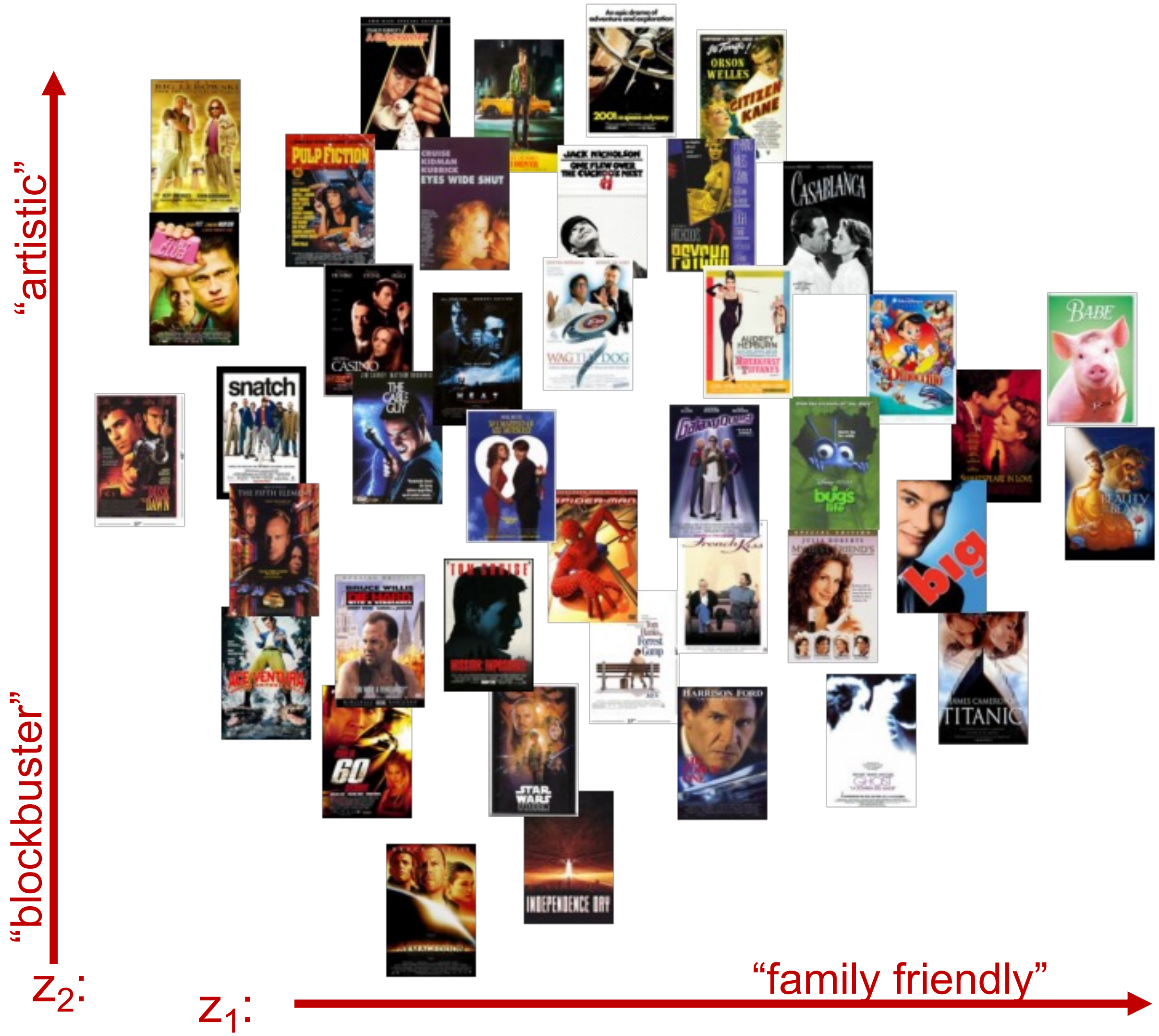
Each movie described by a 2D vector



Each user **also** described by a 2D vector; "likes" movies with a high dot-product (same direction from the origin)

Ex: MovieLens

- 200 movies



Direction Z1 (least to most)

The Crow (1994)
Desperado (1995)
Ronin (1998)
From Dusk Till Dawn (1996)
Kill Bill: Vol. 1 (2003)

⋮ ⋮

About a Boy (2002)
Beauty and the Beast (1991)
The Sound of Music (1965)
Witness (1985)
When Harry Met Sally... (1989)

Direction Z2 (least to most)

Star Trek II: The Wrath of Khan (1982)
Ghostbusters (1984)
The Princess Bride (1987)
Planet of the Apes (1968)
Airplane! (1980)

⋮ ⋮

Dead Poets Society (1989)
Life Is Beautiful (La Vita è bella) (1997)
Juno (2007)
Good Morning, Vietnam (1987)
Chocolat (2000)

Latent space models

- Latent representation encodes some meaning
- What kind of movie is this? What movies is it similar to?
- Matrix is full of missing data
  - Hard to take SVD directly
  - Typically solve using gradient descent
  - Easy algorithm (see Netflix challenge forum)

$$J(U,V) = \sum_{u,m} (X_{mu} - \sum_k U_{mk} V_{ku})^2$$

```
# for user u, movie m, find the kth eigenvector & coefficient by iterating:
predict_um = U[m,:].dot( V[:,u] )    # predict: vector-vector product
err = ( rating[u,m] – predict_um )   # find error residual
V_ku, U_mk = V[k,u], U[m,k]          # make copies for update
U[m,k] += alpha * err * V_ku         # Update our matrices
V[k,u] += alpha * err * U_mk         #   (compare to least-squares gradient)
```

Dimensionality Reduction

- Motivaton
- Principal Component Analysis (PCA)
- Examples of PCA
  - Images
  - Text
  - Ratings
- **AutoEncoders**
- Implicit Embeddings

Collaborative Filtering Models

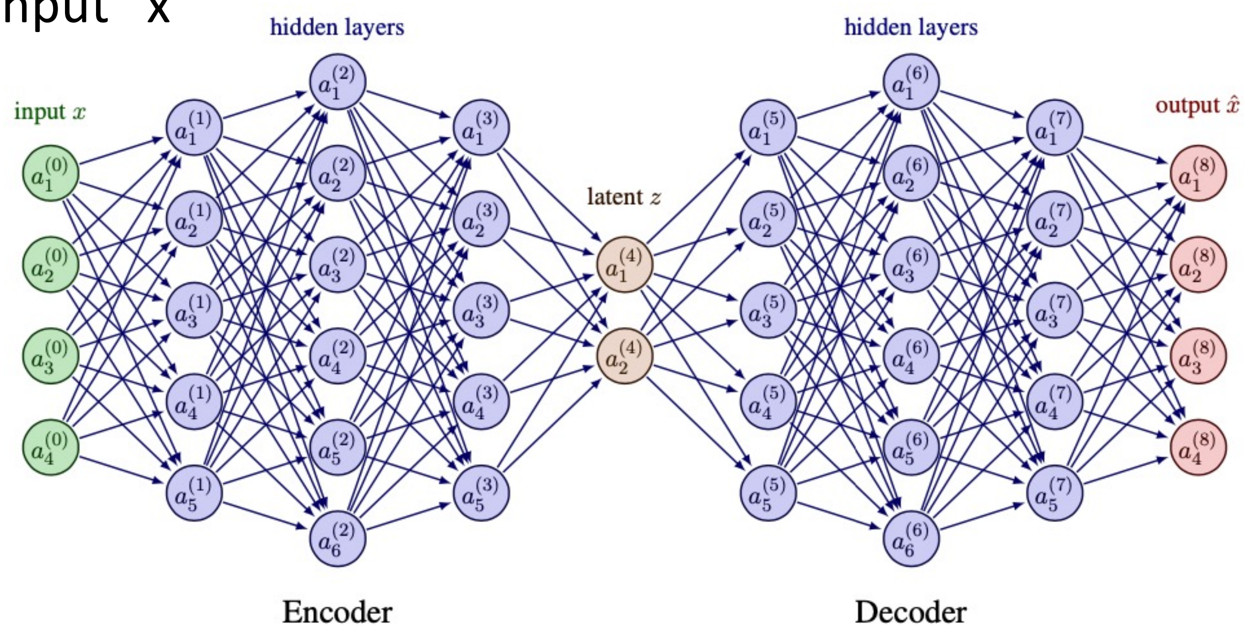
- Can make them a bit more sophisticated:
  - $r_{iu} \approx \mu + b_u + b_i + \sum_k W_{ik} V_{ku}$
  - "Overall average rating"
  - "User effect" + "Item effect"
  - Latent space effects (k indexes latent representation)
  - Use saturating non-linearity? (ratings only 0…9)

- Then, just train some loss, e.g. MSE, with SGD
  - Each (user, item, rating) is one data point

Nonlinear latent spaces

- Latent space
  - Any alternative representation (usually smaller) from which we can (approximately) recover the data
  - Linear: "Encode" $Z = X V^T$; "Decode" $X \approx Z V$

- Ex: Auto-encoders
  - Use neural network with ``bottleneck'' of latent dimension
  - Train to recover the input "x"



word2vec

- Related: word2vec
  - Trains an NN to recover the context of words
  - Use internal hidden

Dimensionality Reduction

Motivaton

Principal Component Analysis (PCA)

Examples of PCA

Images

Text

Ratings

AutoEncoders

Implicit Embeddings

Implicit Embeddings

- Auto-encoder viewpoint: learn a transformation

$$x \quad \longrightarrow \quad z = f(x) \quad \longrightarrow \quad \hat{x} = g(z)$$

  - f,g are "approximate inverses", to reconstruct $\{x^{(i)}\}$ as closely as possible

- A different approach: just learn the values $\{z^{(i)}\}$ directly
  - Transform defined by the locations $z^{(i)}$ for each data point $x^{(i)}$.
  - Optimize over locations z directly
  - Objective: preserve "similarity" between (some/all) data points.

- Examples:
  - Multi-dimensional scaling; IsoMap
  - Locally linear embedding (LLE)
  - T-distributed Stochastic Neighbor Embedding (TSNE)

- One issue: out-of-sample data are difficult to embed later…

Embedding: TSNE

- Ex: MNIST data