Principal Component Analysis

PROF XIAOHUI XIE
SPRING 2019

CS 273P Machine Learning and Data Mining

Slides courtesy of Alex Ihler

Machine Learning

Dimensionality Reduction

Principal Components Analysis (PCA)

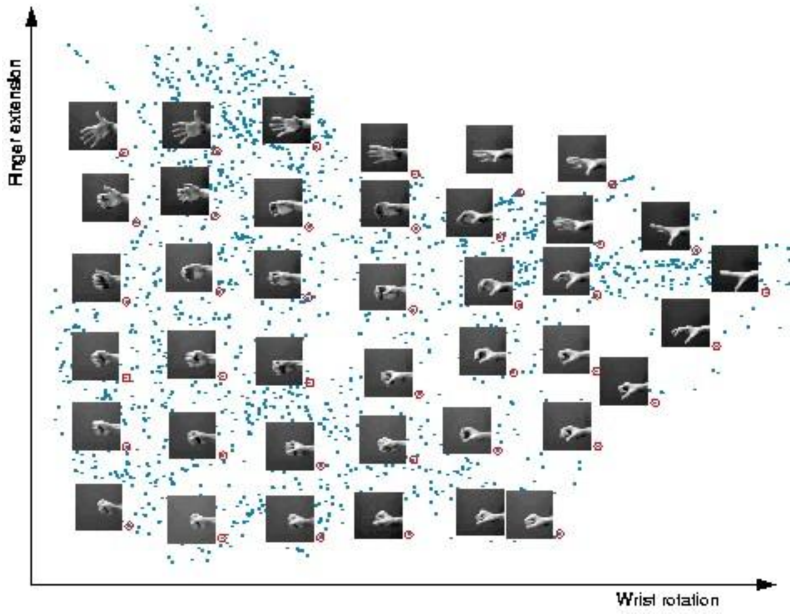
Applications of PCA: Eigenfaces & LSI

Collaborative Filtering

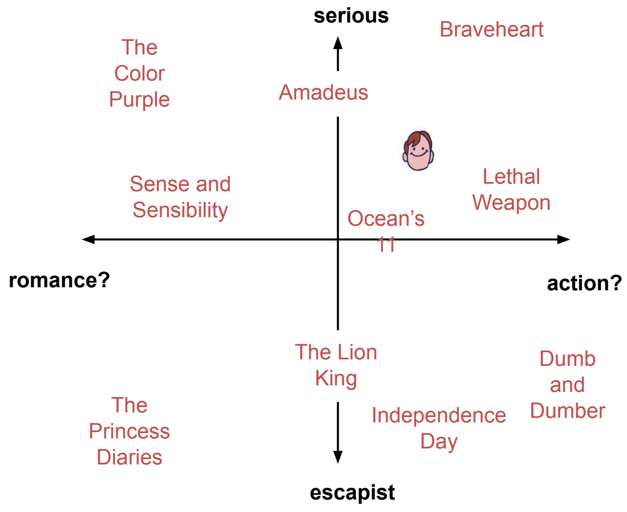
Motivation

- 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

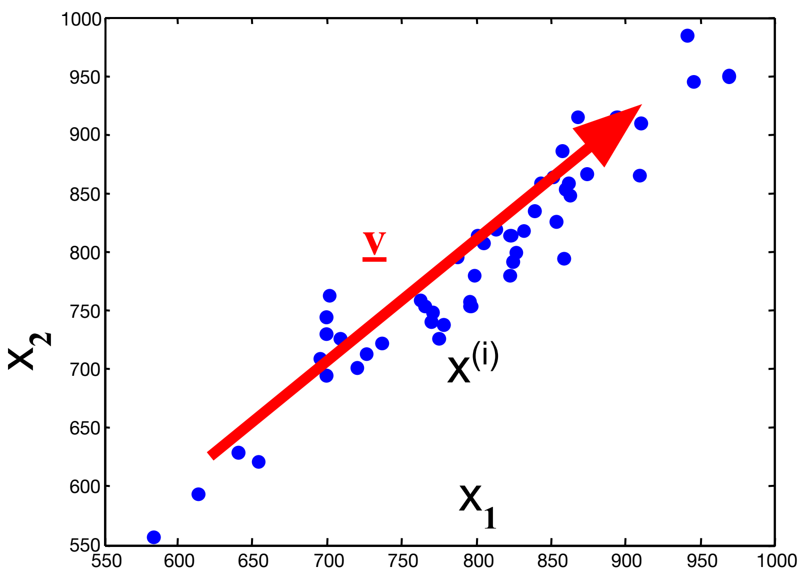
- Ex: S&P 500 – vector of 500 (change in) values per day
  - But, lots of structure
  - Some elements tend to “change together”
  - Maybe we only need a few values to approximate it?
  - “Tech stocks up 2x, manufacturing up 1.5x, …” ?

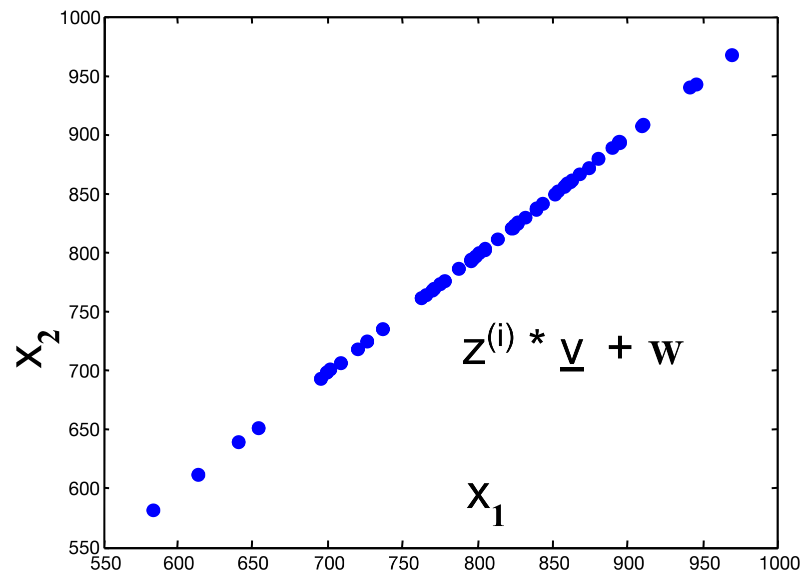
- How can we access that structure?

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



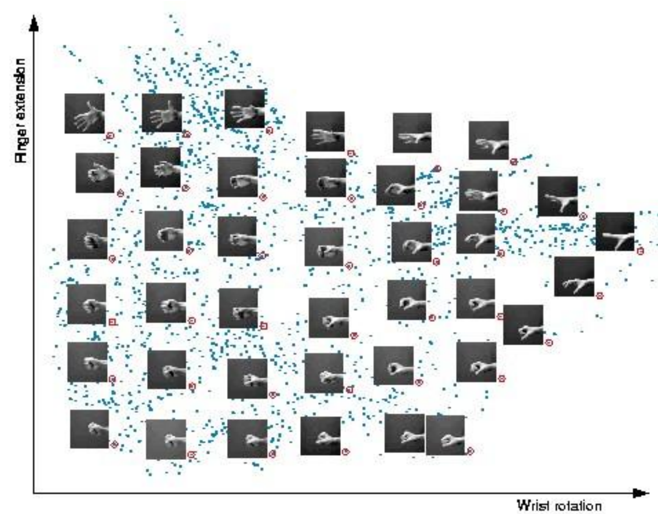


Some uses of latent spaces

- 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



Machine Learning

Dimensionality Reduction

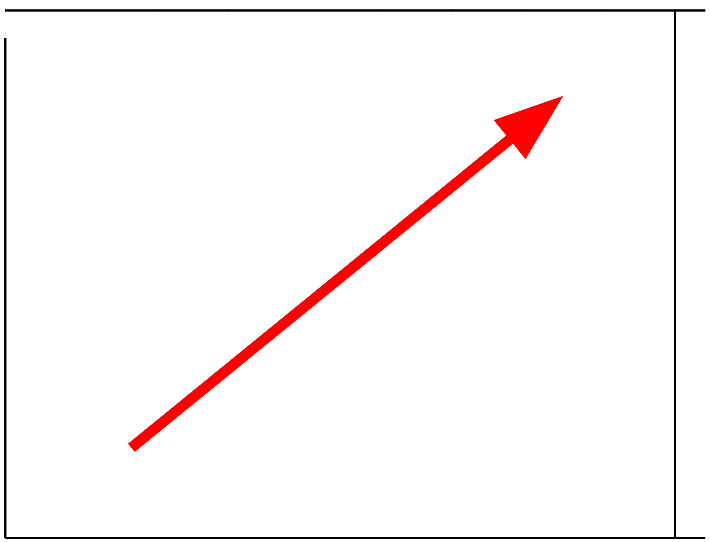
Principal Components Analysis (PCA)

Applications of PCA: Eigenfaces & LSI

Collaborative Filtering

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:

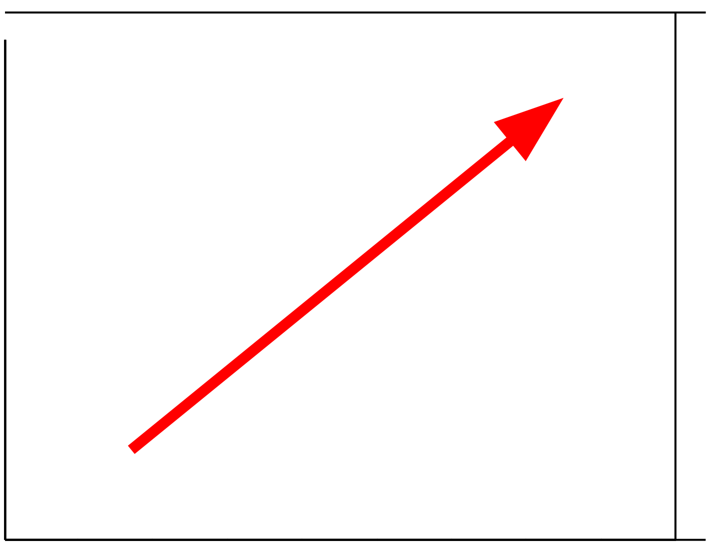
$$\max_v \; v^T \tilde{X}^T \tilde{X} v \quad s.t. \; ||v|| = 1$$

→ 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:

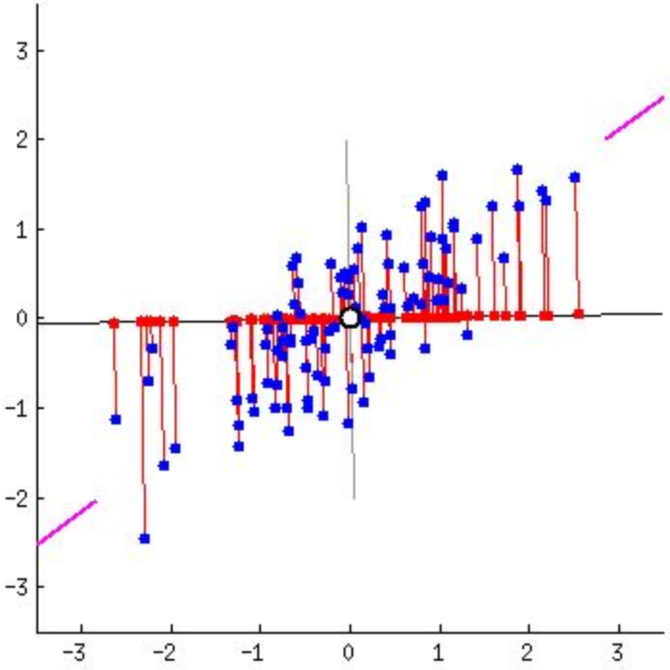
$$\sum_i (z^{(i)})^2 = z^T z = v^T \tilde{X}^T \tilde{X} v$$

Best "direction" v:

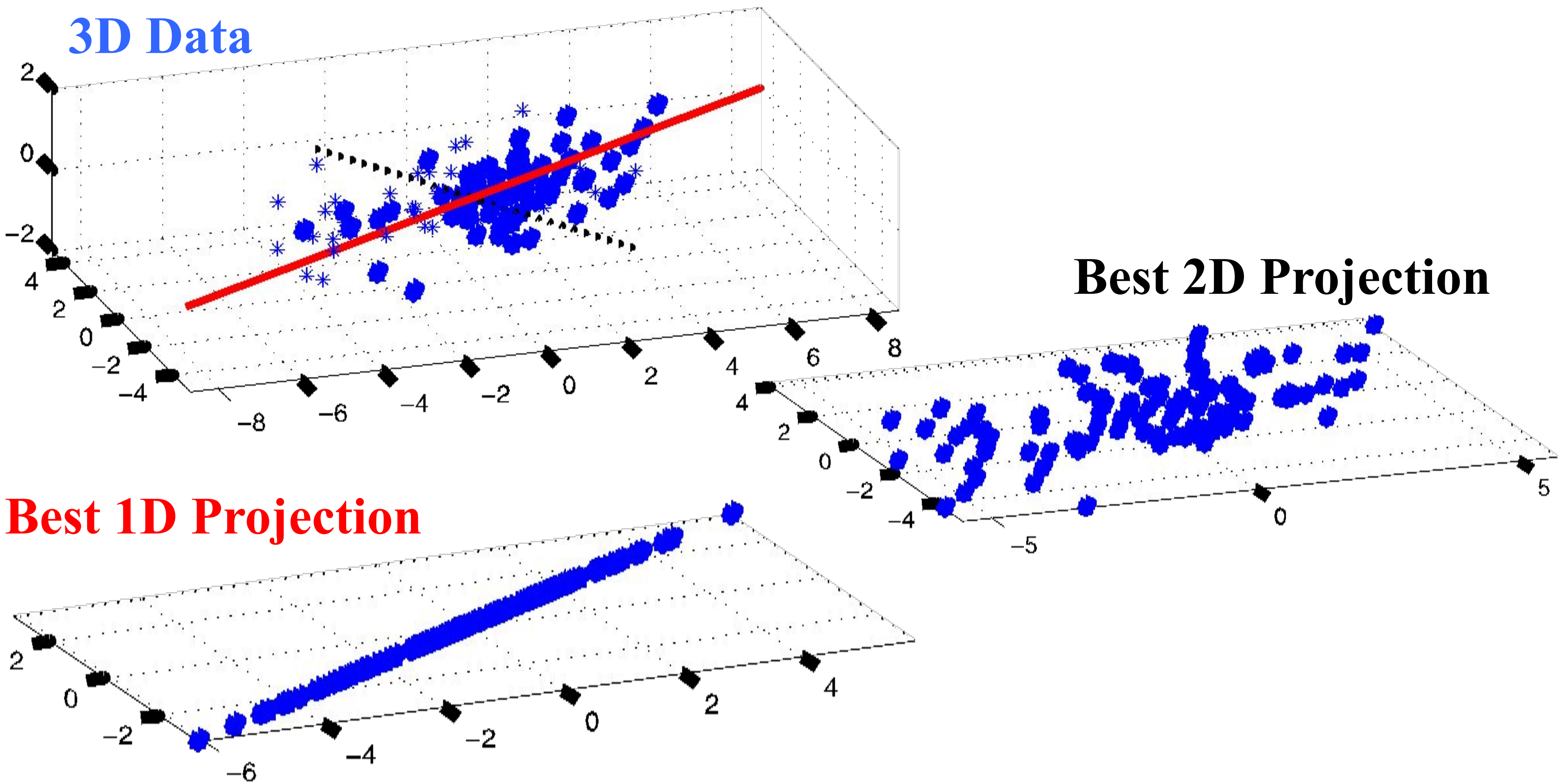
$$\max_v \; v^T \tilde{X}^T \tilde{X} v \quad s.t. \; ||v|| = 1$$

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

Principal Components Analysis



Principal Components Analysis (PCA)

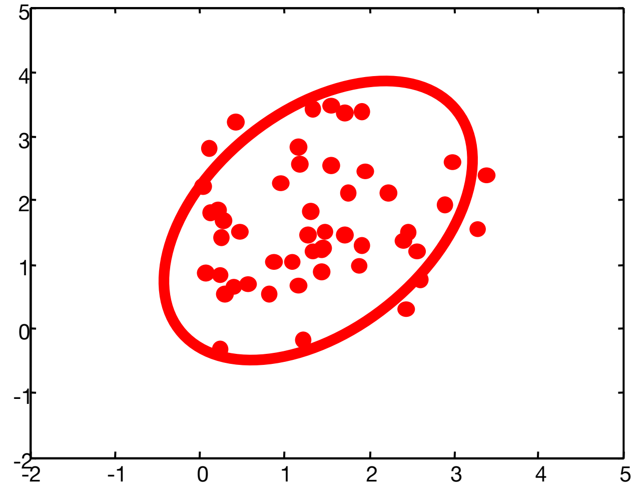


TODO

- Notion of “trace”: total variance of data?
  - Invariant to rotation?
  - Decompose into “variance captured” vs “residual”?
  - Clear derivation…
  - Notation for “zero-meaned” X (tilde X?)
  - Make text / BOW example clearer, details
  - Add word2vec example also (from end)

Another interpretation

- Data covariance: $\Sigma = \frac{1}{m}\tilde{X}^T\tilde{X}$ $\tilde{X} = X - \mu$



  – Describes "spread" of the data
  – Draw this with an ellipse
  – Gaussian is

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

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

  – 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…

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

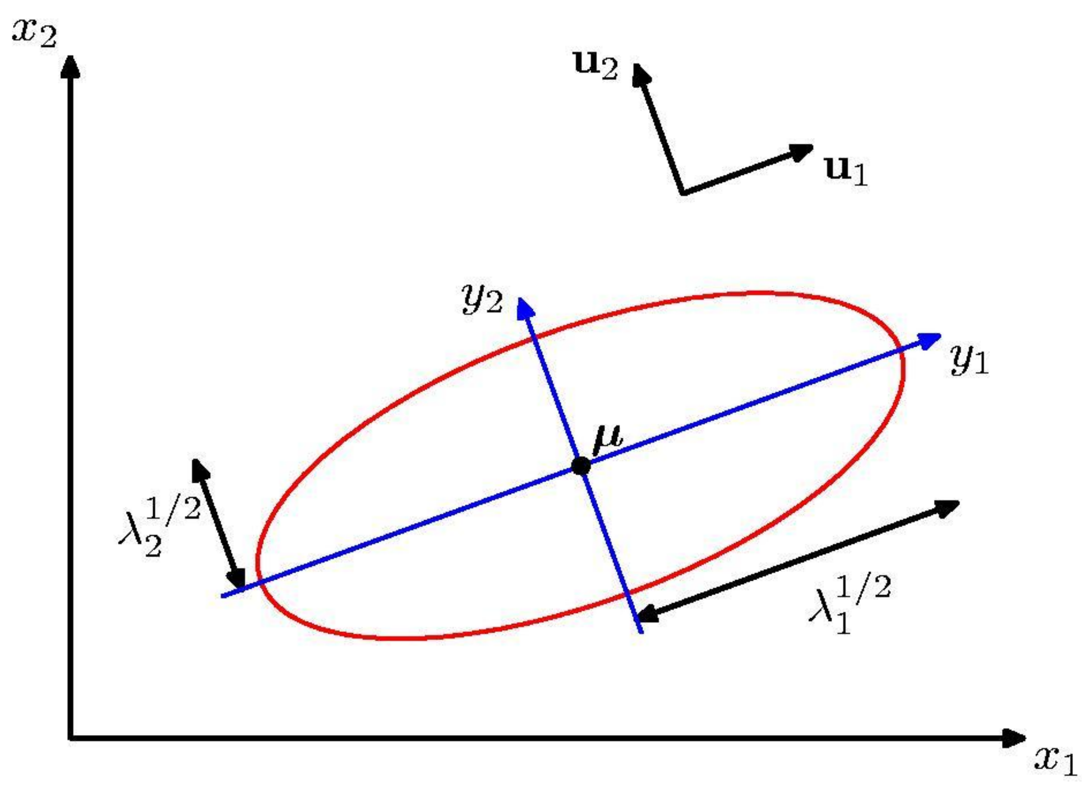
Write Σ in terms of eigenvectors…

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

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

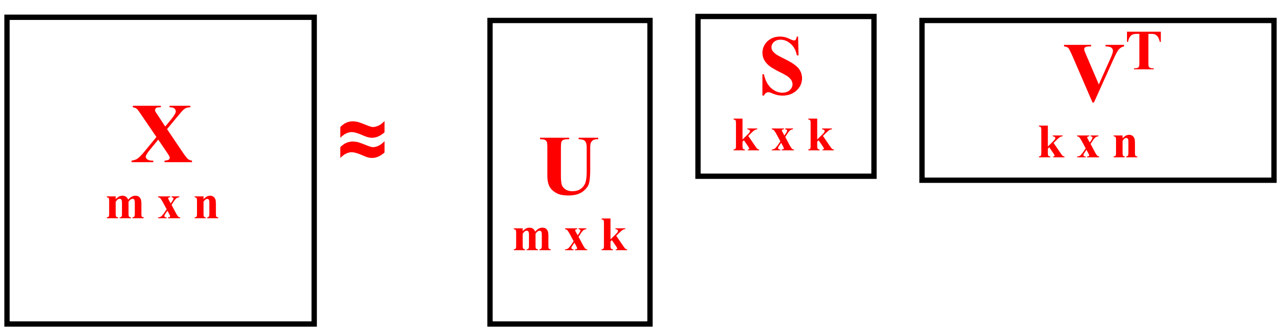
- Subtract data mean from each point
- (Typically) scale each dimension by its variance
  - Helps pay less attention to magnitude of the variable
- Compute covariance matrix, $S = 1/m \sum (x^i-\mu)' (x^i-\mu)$
- Compute the k largest eigenvectors of S

$$S = V D V^T$$

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

Machine Learning

Dimensionality Reduction

Principal Components Analysis (PCA)

Applications of PCA: Eigenfaces & LSA

Collaborative Filtering

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

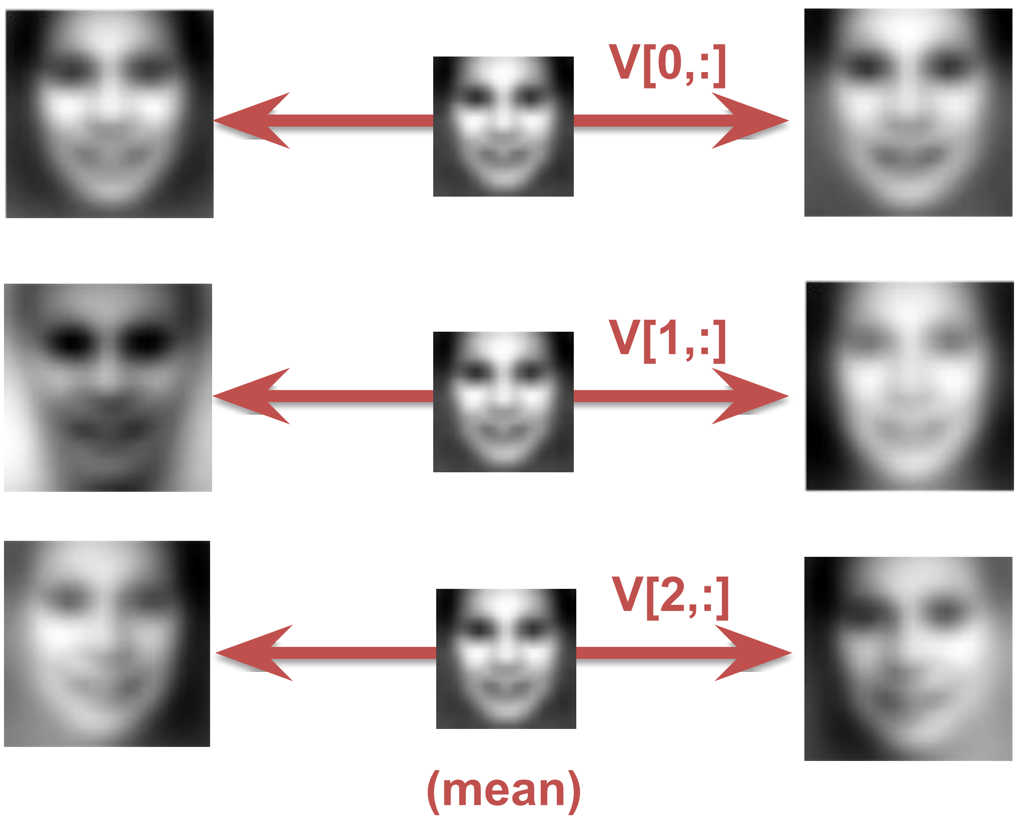
and more…

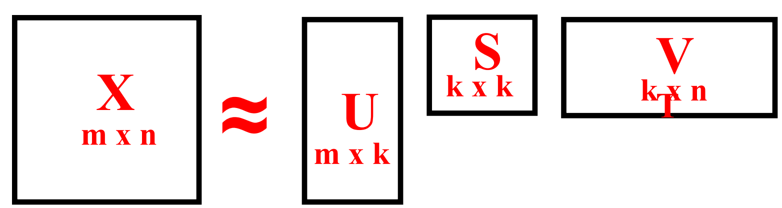
“Eigen-faces”

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

“Eigen-faces”

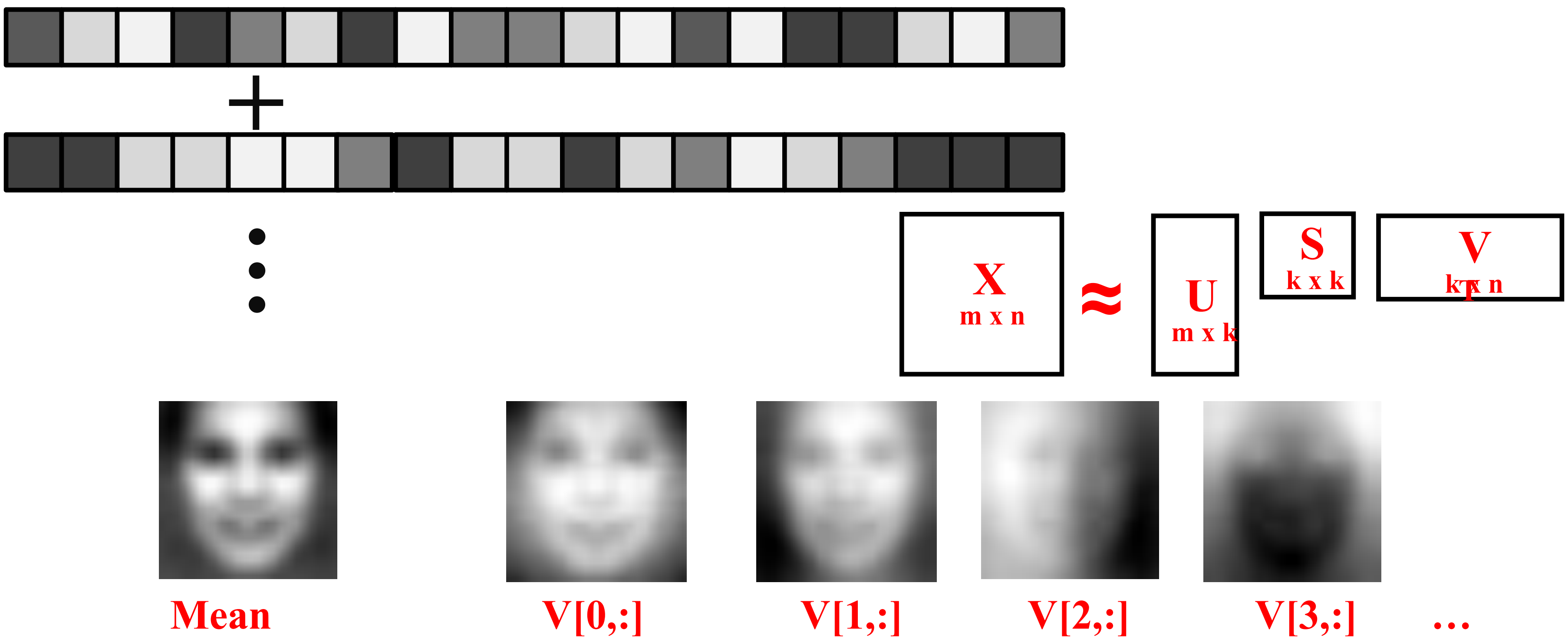
- “Eigen-X” = represent X using PCA
- Ex: Viola Jones data set
  - 24x24 images of faces = 576 dimensional measurements
  - Take first K PCA components





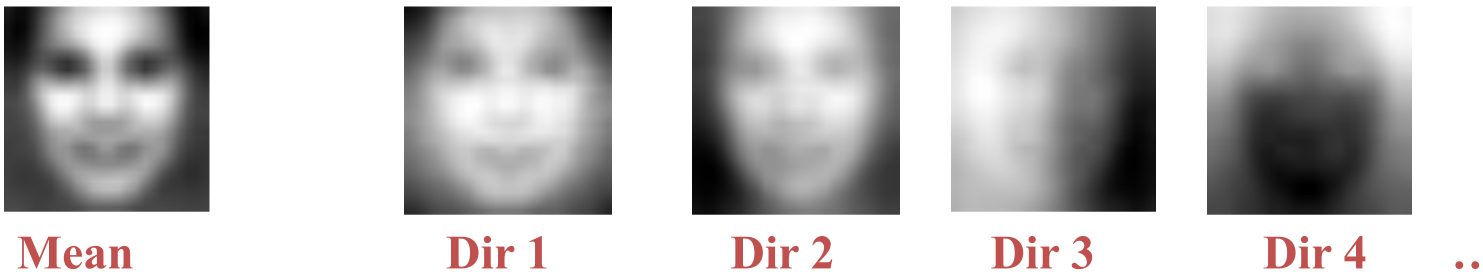
“Eigen-faces”

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

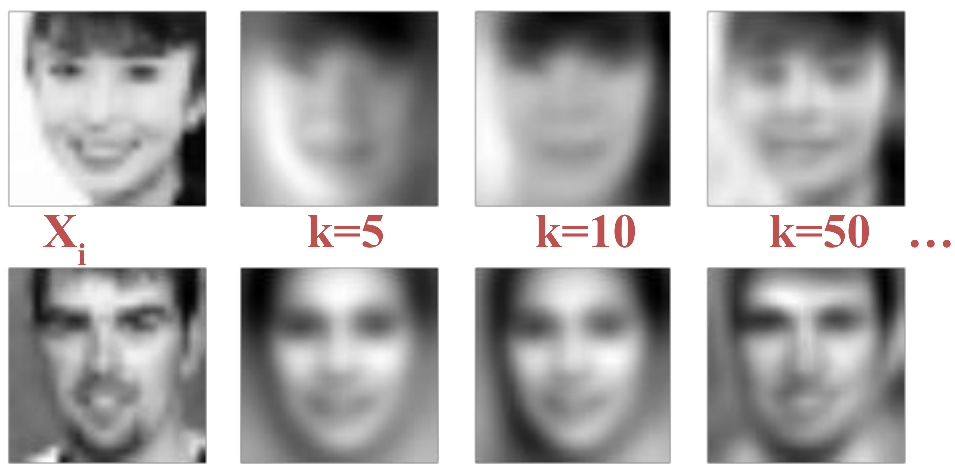


“Eigen-faces”

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



Projecting data onto first k dimensions



"Eigen-faces"

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

Projecting data onto first k dimensions

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 Colorado Boulevard with a group of friends.

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

Text representations

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

**Rain and chilly weather didn't keep thousands (
paradegoers from camping out Friday night for
of Roses.**

**Spirits were high among the street party crowd
for curbside seats for today's parade.**

**``I want to party all night,'' said Tyne Gaudielle
Glendale, who spent the last night of the year al
Boulevard with a group of friends.**

**Whether they came for the partying or the para
in for a long night. Rain continued into the even
temperatures were expected to dip down into th**

nyt/2000-01-01.0015.txt
rain
chilly
weather
didn
keep
thousands
paradegoers
camping
out
friday
night
111th
tournament
roses
spirits
high
among
street
party
crowd
they
set

Text representations

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

VOCABULARY:
0001 ability
0002 able
0003 accept
0004 accepted
0005 according
0006 account
0007 accounts
0008 accused
0009 act
0010 acting
0011 action
0012 active
….

Observed Data (text docs):

DOC #	WORD #	COUNT
1	29	1
1	56	1
1	127	1
1	166	1
1	176	1
1	187	1
1	192	1
1	198	2
1	356	1
1	374	1
1	381	2
…		

Example: Documents

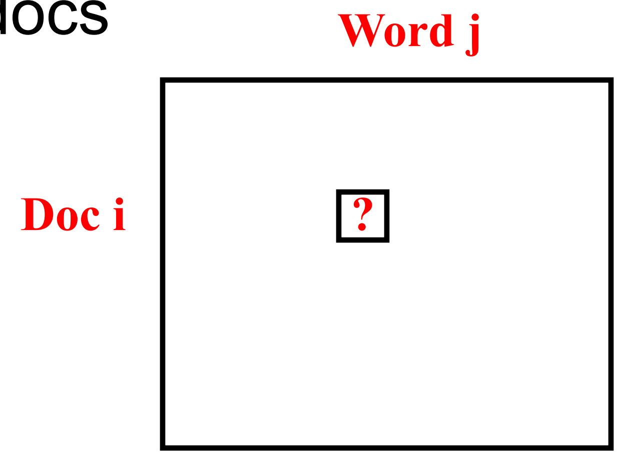
c1: Human machine interface for ABC computer applications
c2: A survey of user opinion of computer system response time
c3: The EPS user interface management system
c4: System and human system engineering testing of EPS
c5: Relation of user perceived response time to error measurement

m1: The generation of random, binary, ordered trees
m2: The intersection graph of paths in trees
m3: Graph minors IV: Widths of trees and well-quasi-ordering
m4: Graph minors: A survey

From http://lsa.colorado.edu/papers/dp1.LSAintro.pdf

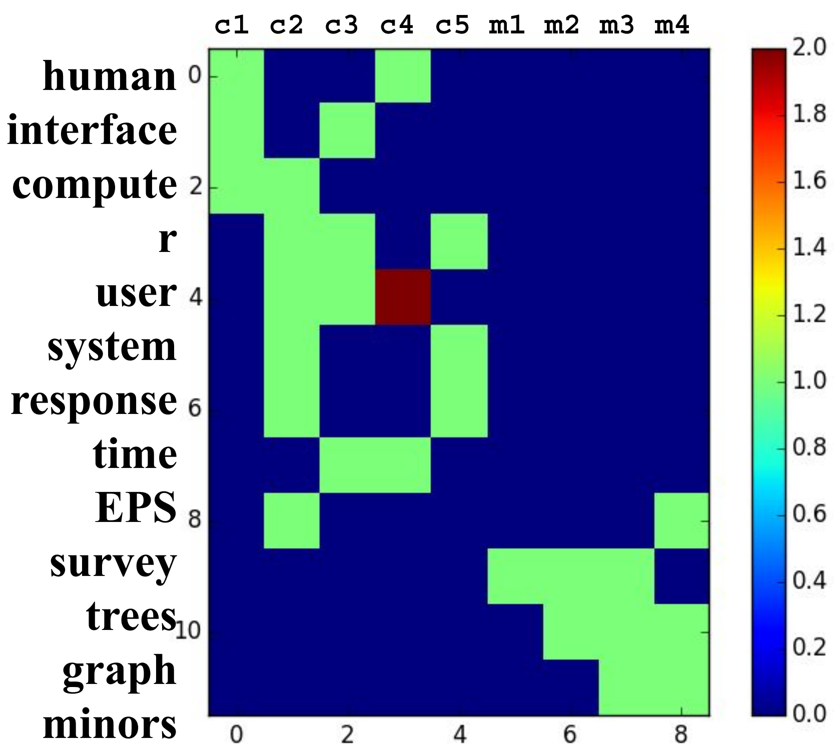
Latent Semantic Analysis (LSA)

- PCA for text data
- Create a giant matrix of words in docs
  - "Word j appears" = feature x_j
  - "in document i" = data example I



- Huge matrix (mostly zeros)
  - Typically normalize rows to sum to one, to control for short docs
  - Typically don't subtract mean or normalize columns by variance
  - Might transform counts in some way (log, etc)
- PCA on this matrix provides a new representation
  - Document comparison
  - Fuzzy search ("concept" instead of "word" matching)

Example: Word-Doc Matrix



Matrices are big, but data is sparse

- Typical example:
  - Number of docs, D ~ 10^6
  - Number of unique words in vocab, W ~ 10^5
  - FULL Storage required ~ 10^{11}
  - Sparse Storage required ~ 10^8
- DxW matrix (# docs x # words)
  - Each entry is non-negative
  - Typically integer / count data

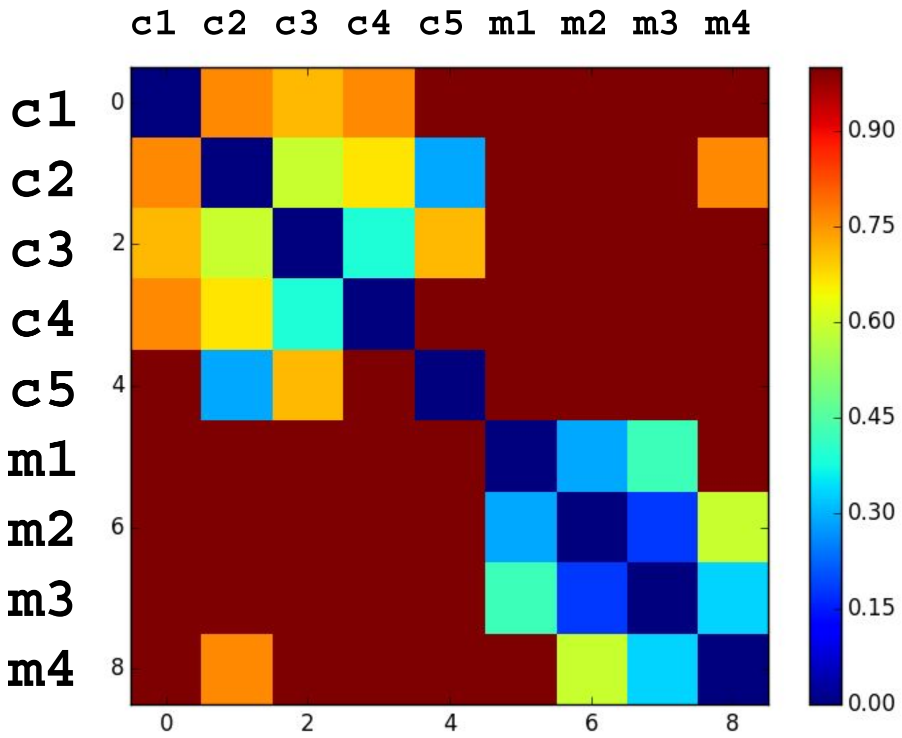
Problem with Sparse Matrices

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

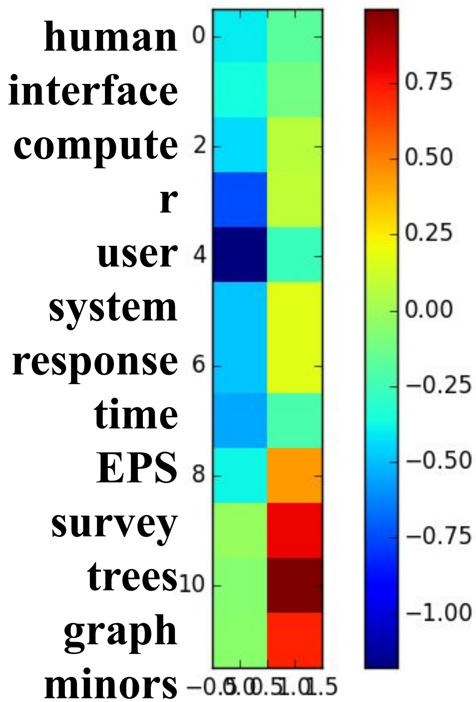
m4: Graph minors: A survey

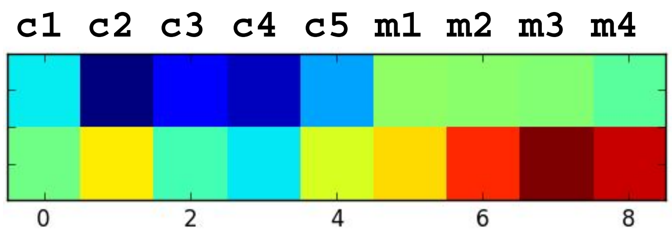
c1: Human machine interface for ABC computer applications

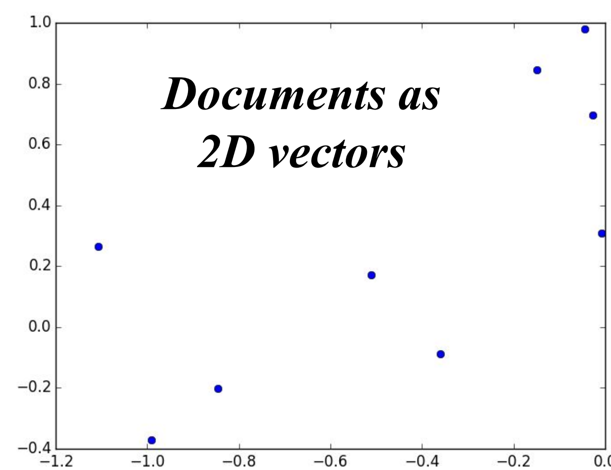
Example: Document Distance Matrix



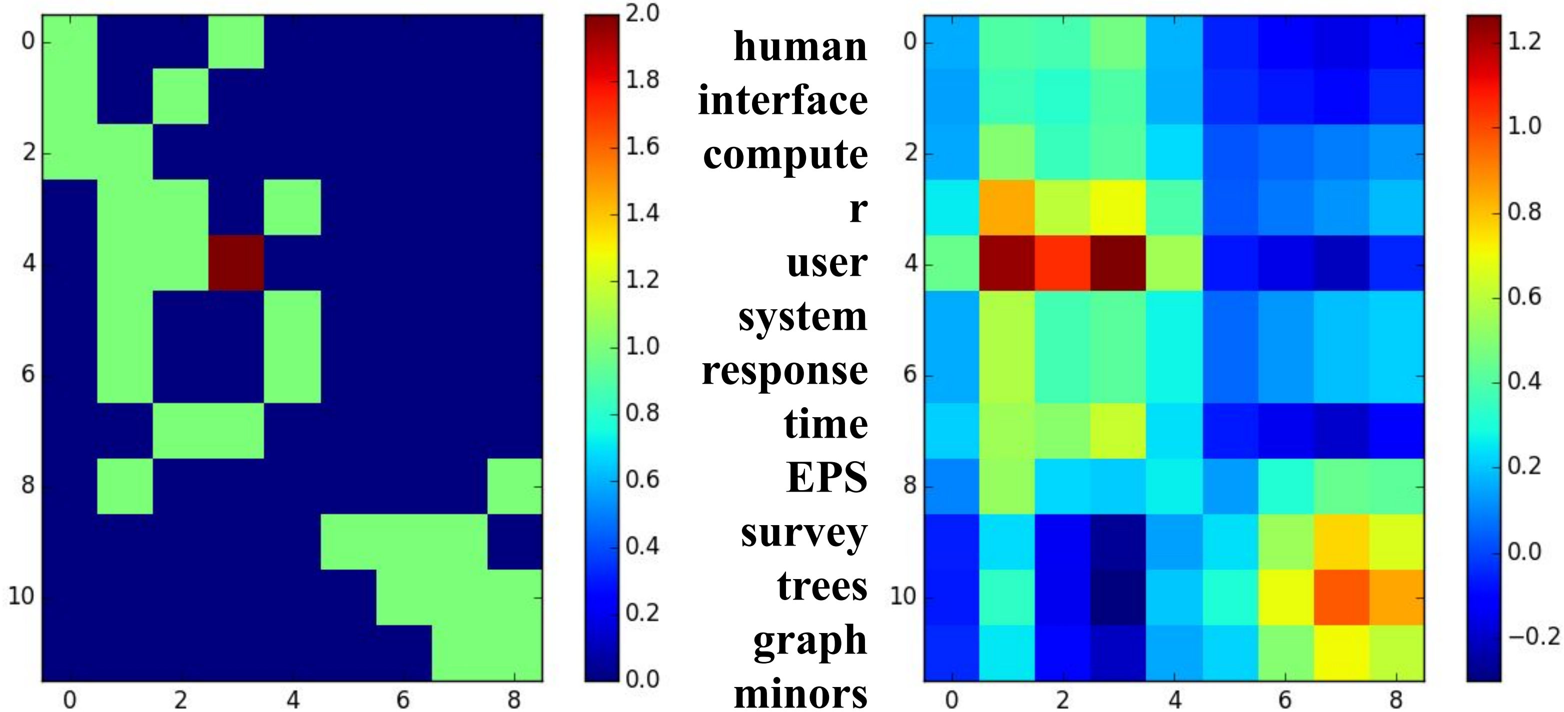
Example: Decomposition



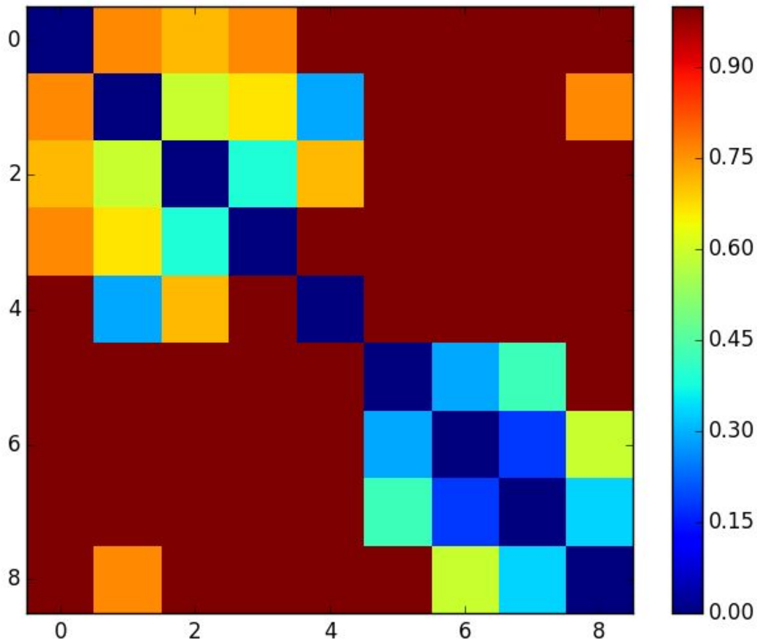


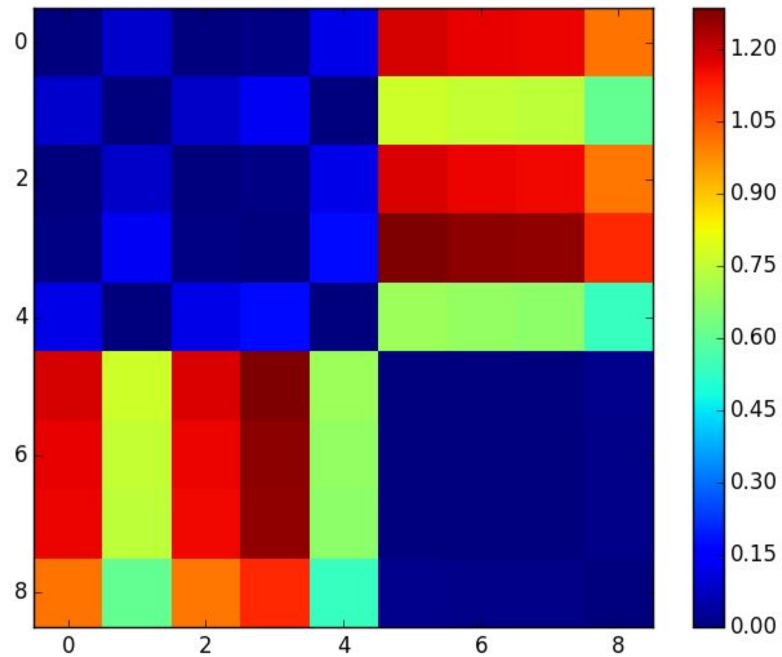


Example: Reconstruction



Example: Distance Matrix





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
0.089 control
0.088 human
0.086 cancer
0.084 protein
0.084 sample
0.083 loci
0.082 microarray

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
0.089 control
0.088 human
0.086 cancer
0.084 protein
0.084 sample
0.083 loci
0.082 microarray

PRINCIPAL COMPONENT 2

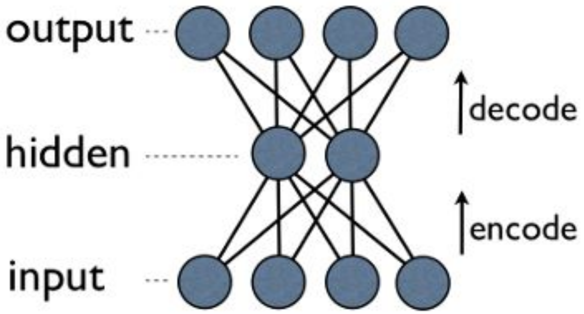
0.247 snp
-0.196 cell
0.187 variant
0.181 risk
0.180 gwas
0.162 population
0.162 genome_wide
0.155 genetic
0.130 loci
-0.116 mir
-0.116 expression
0.113 allele
0.108 schizophrenia
0.107 disease
-0.103 mirnas
-0.099 protein
-0.089 gene_expression
0.087 polymorphism
0.087 susceptibility
0.084 trait

Q: But what does -0.196 cell mean?

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 few internal nodes
  - Train to "recover" the input "x"



stats.stackexchange.com

- Related: word2vec
  - Trains an NN to recover the context of words
  - Use internal hidden node responses as a vector representation of the word

Machine Learning

Dimensionality Reduction

Principal Components Analysis (PCA)

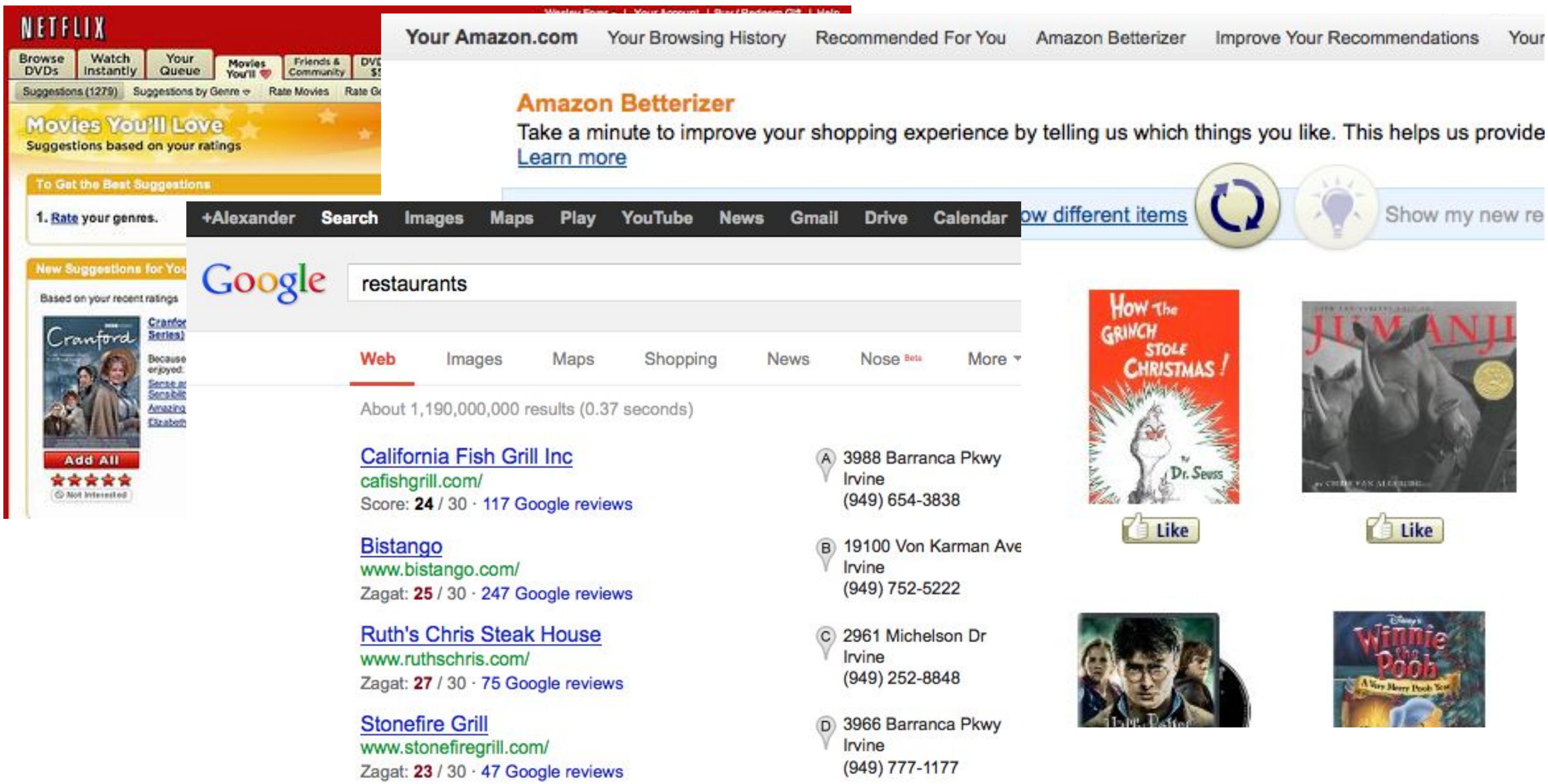
Applications of PCA: Eigenfaces & LSI

Collaborative Filtering

Recommender systems

- Automated recommendations
- Inputs
  - User information
    - Situation context, demographics, preferences, past ratings
  - Items
    - Item characteristics, or nothing at all
- Output
  - Relevance score, predicted rating, or ranking

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
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
3988 Barranca Pkwy
Irvine
(949) 654-3838
Bistango
www.bistango.com/
Zagat: 25 / 30 · 247 Google reviews
19100 Von Karman Ave
Irvine
(949) 752-5222
Ruth's Chris Steak House
www.ruthschris.com/
Zagat: 27 / 30 · 75 Google reviews
2961 Michelson Dr
Irvine
(949) 252-8848
Stonefire Grill
www.stonefiregrill.com/
Zagat: 23 / 30 · 47 Google reviews
3966 Barranca Pkwy
Irvine
(949) 777-1177

Paradigms

Recommender systems reduce information overload by estimating relevance

Recommendation
system

Item	score
I1	0.9
I2	1
I3	0.3
…	…

Recommendations

Paradigms

User profile / context

Personalized recommendations

Recommendation system

Item	score
I1	0.9
I2	1
I3	0.3
…	…

Recommendations

Paradigms

Content-based:
"Show me more of the same things that I've liked"

User profile / context

Title	Genre	Actors	…

Product / item features

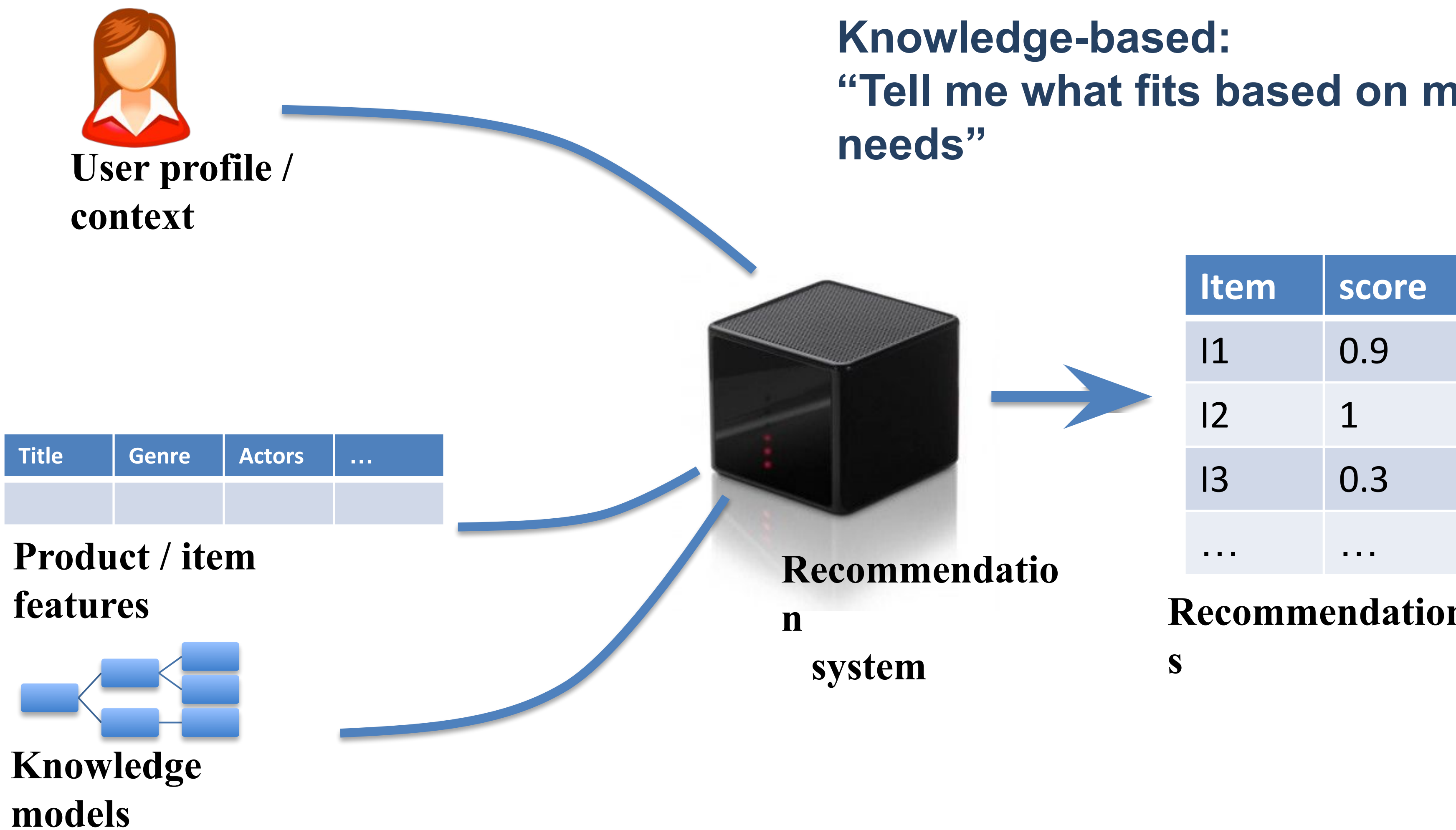
Recommendatio
n
system

Item	score
I1	0.9
I2	1
I3	0.3
…	…

Recommendation
s

Paradigms

Knowledge-based:
"Tell me what fits based on my needs"



Paradigms

Collaborative:
"Tell me what's popular among my peers"

User profile / context

Community data

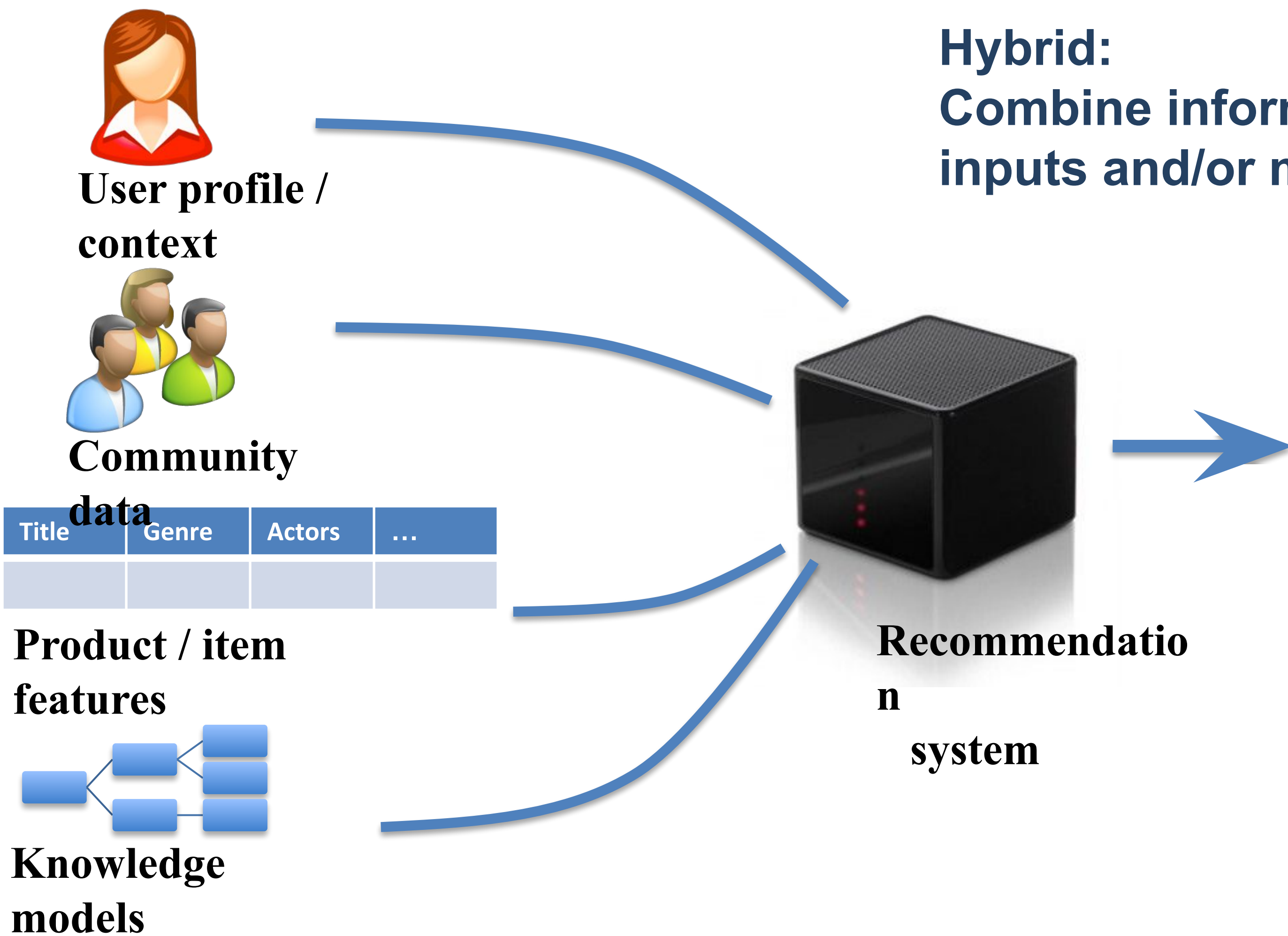
Recommendation system

Item	score
I1	0.9
I2	1
I3	0.3
…	…

Recommendations

Paradigms

Hybrid:
Combine information from many inputs and/or methods



Item	score
I1	0.9
I2	1
I3	0.3
…	…

Measuring success

- Prediction perspective
  - Predict to what degree users like the item
  - Most common evaluation for research
  - Regression vs. “top-K” ranking, etc.
- Interaction perspective
  - Promote positive “feeling” in users (“satisfaction”)
  - Educate about the products
  - Persuade users, provide explanations
- Conversion perspective
  - Commercial success
  - Increase “hits”, “click-through” rates
  - Optimize sales and profits

Why are recommenders important?

- The long tail of product appeal
  - A few items are very popular
  - Most items are popular only with a few people
    - But everybody is interested in *some* rare products
- Goal: recommend not-widely known items that the user might like!

Collaborative filtering (Netflix)

From Y. Koren of BellKor team

users

movies

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

Collaborative filtering

- Simple approach: standard regression
  - Use "user features" A_u, "item features" A_i
  - Train $f(A_u, A_i) \to r_{ui}$
  - Learn "users with my features like items with these features"

- Extreme case: per-user model / per-item model
- Issues: needs lots of side information!

Features:

1 0 1 0 0 …

0 0 1 0 0 …

…

users

movies

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

Collaborative filtering

- Example: nearest neighbor methods
  - Which data are "similar"?

- Nearby items? (based on…)

Features:

1 0 1 0 0 …

0 0 1 0 0 …

…

users

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

Collaborative filtering

- Example: nearest neighbor methods
  - Which data are "similar"?

- Nearby items? (based on…)

Based on ratings alone?

Find other items that are rated similarly…

Good match on observed ratings

users

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

Collaborative filtering

- Which data are “similar”?
- Nearby items?
- Nearby users?
  - Based on user features?
  - Based on ratings?

users

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

Collaborative filtering

- Some **very simple** examples
  - All users similar, items not similar?
  - All items similar, users not similar?
  - All users and items are equally similar?

users

movies

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

Measuring similarity

- Nearest neighbors depends significantly on distance function
  - "Default": Euclidean distance
- Collaborative filtering:
  - Cosine similarity: $\frac{x^{(i)} \cdot x^{(j)}}{\|x^{(i)}\| \|x^{(j)}\|}$ (measures angle between x^i, x^j)
  - Pearson correlation: measure correlation coefficient between x^i, x^j
  - Often perform better in recommender tasks $\frac{(x^{(i)} - \mu) \cdot (x^{(j)} - \mu)}{\|x^{(i)} - \mu\| \|x^{(j)} - \mu\|}$
- Variant: weighted nearest neighbors
  - Average over neighbors is weighted by their similarity
- Note: with ratings, need to deal with missing data!

Nearest-Neighbor methods

users

movies

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

Neighbor selection:
Identify movies similar to 1, rated by user 5

Nearest-Neighbor methods

users

movies

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

Compute similarity weights:

$s_{13}=0.2$, $s_{16}=0.3$

Nearest-Neighbor methods

users

movies

12	11	10	9	8	7	6	5	4	3	2	1	
	4		5			5	**2.6**		3		1	1
3	1	2			4			4	5			2
	5	3	4		3		2	1		4	2	**3**
	2			4			5		4	2		4
5	2					2	4	3	4			5
	4			2			3		3		1	**6**

Predict by taking weighted average:

$(0.2*2+0.3*3)/(0.2+0.3)=2.6$

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

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

~

~

items

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

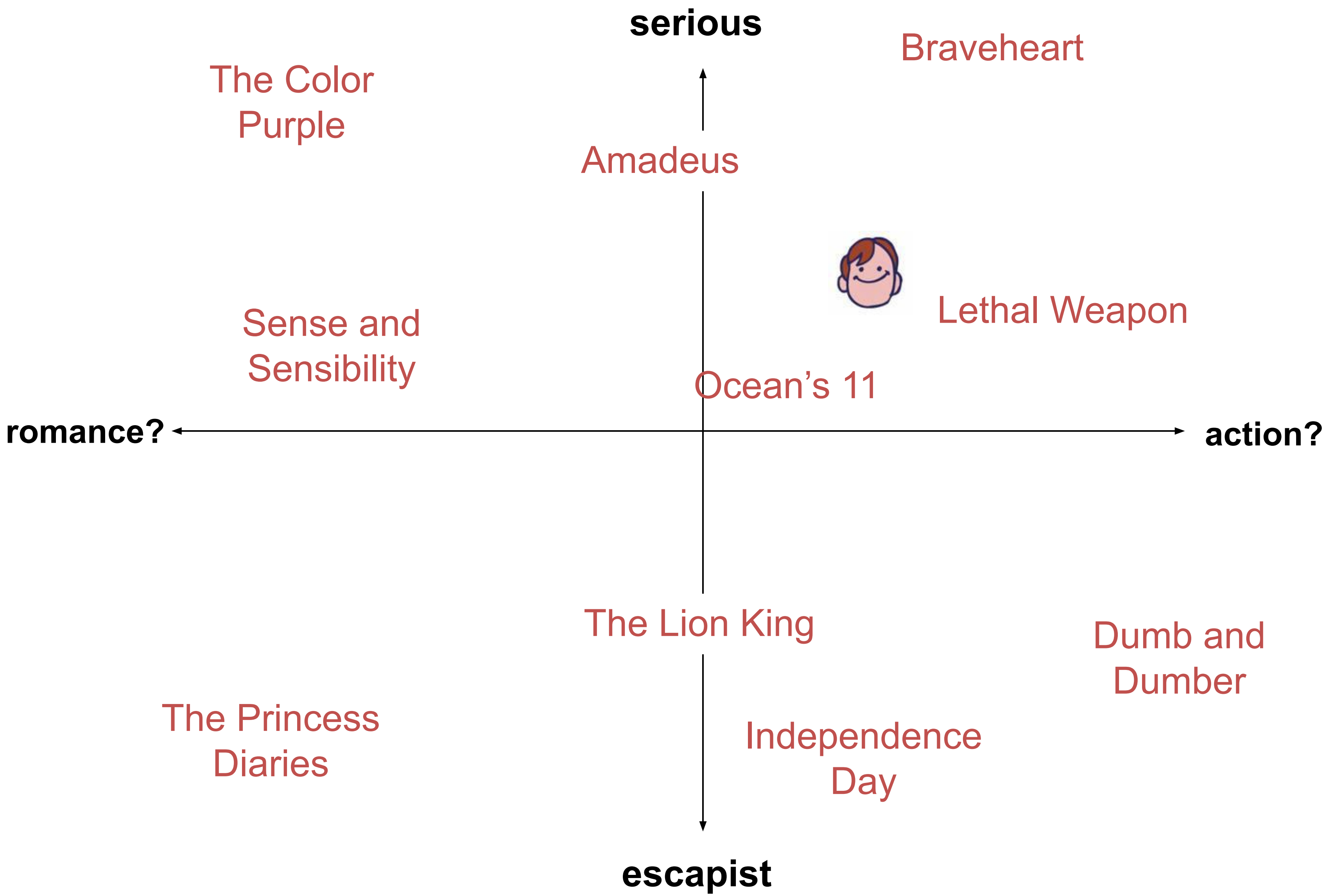
•

users

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

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

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

```
# 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)
```

Latent space models

- Can be 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)
  - (Saturating non-linearity?)
- Then, just train some loss, e.g. MSE, with SGD
  - Each (user, item, rating) is one data point

Ensembles for recommenders

- Given that we have many possible models:
  - Feature-based regression
  - (Weighted) kNN on items
  - (Weighted) kNN on users
  - Latent space representation
- Perhaps we should combine them?

- Use an ensemble average, or a stacked ensemble
  - “Stacked” : train a weighted combination of model predictions