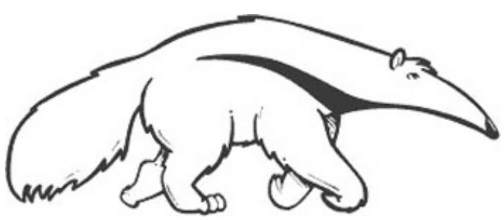
+

Machine Learning and Data Mining

Dimensionality Reduction; PCA & SVD

Prof. Alexander Ihler



BREN:ICS
INFORMATION AND COMPUTER SCIENCES

UNIVERSITY of CALIFORNIA IRVINE

Motivation

- High-dimensional data
  - Images of faces
  - Text from articles
  - All S&P 500 stocks
- Can we describe them in a “simpler” way?

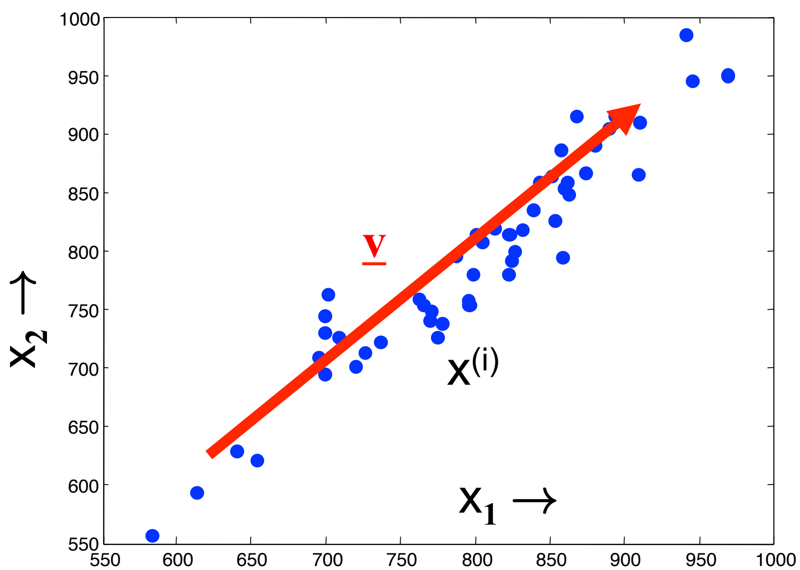
- Ex: S&P 500 – vector of 500 (change in) values per day
  - But, lots of structure
  - Some elements tend to “change together”
  - Maybe we only need a few values to approximate it?
  - “Tech stocks up 2x, manufacturing up 1.5x, …” ?

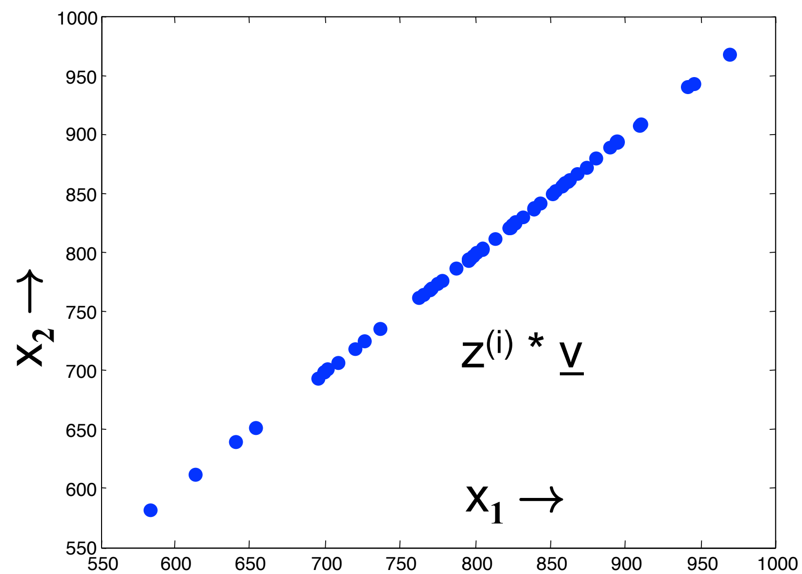
- How can we access that structure?

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



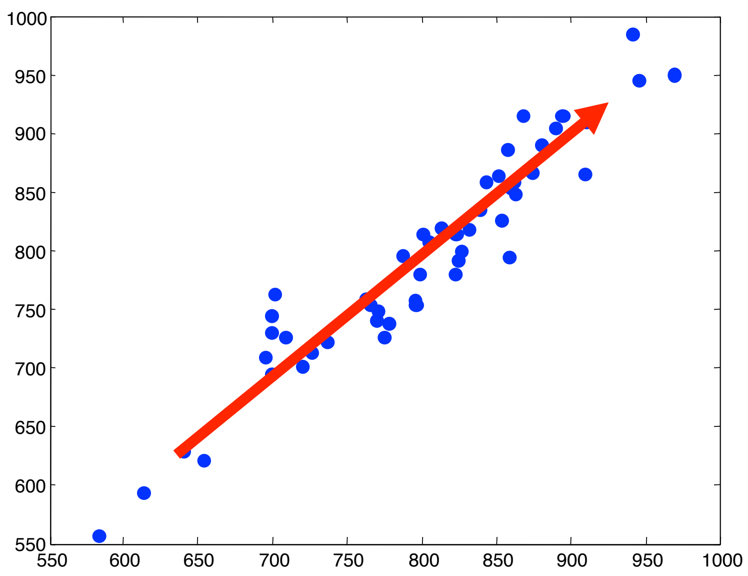


Principal Components Analysis

- What is the vector that would most closely reconstruct X?

$$\min_{a,v} \sum_i (x^{(i)} - a^{(i)} v)^2$$

  - Given v: $a^{(i)}$ is the projection of each point $x^{(i)}$ onto v
  - v chosen to minimize the residual variance
  - Equivalently, v is the direction of maximum variance
  - Extensions: best two dimensions: xi= ai*v + bi*w + m



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…

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

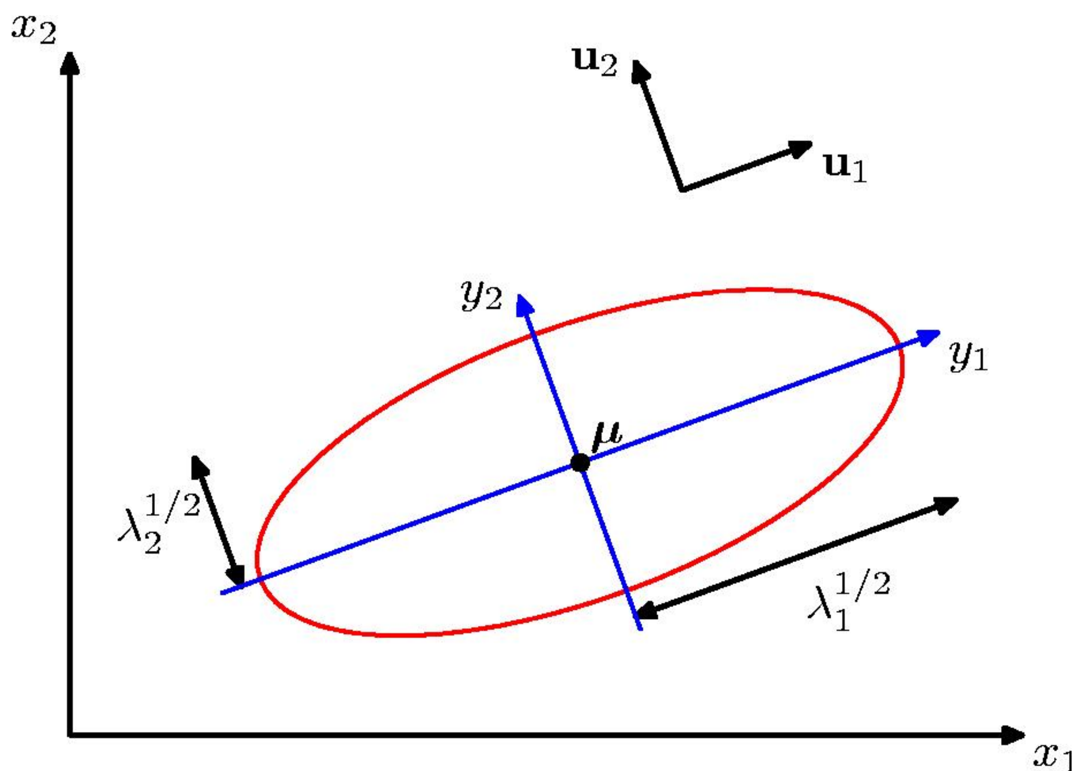
Write Σ in terms of eigenvectors…

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

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

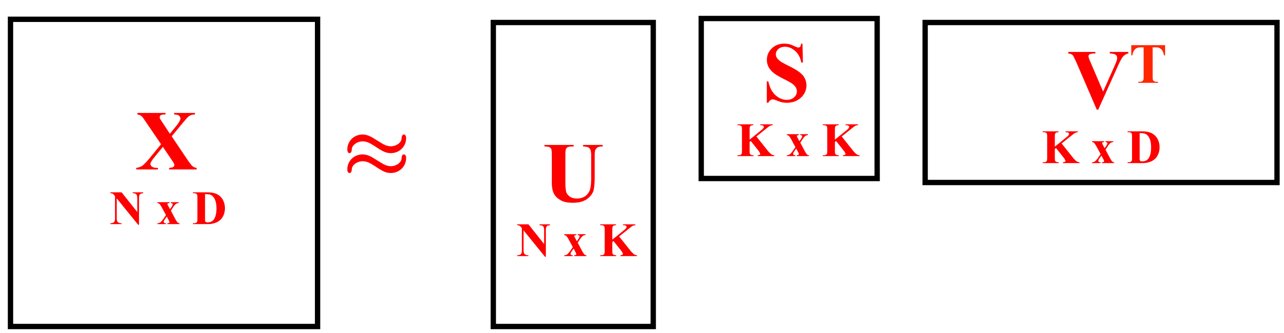
- Subtract data mean from each point
- (Typically) Scale each dimension by its variance
  - Helps pay less attention to magnitude of the variable
- Compute covariance matrix, S = 1/n \sum (xi-m)' (xi-m)
- Compute the k largest eigenvectors of S

  S = V D V^T

```
mn = mean(X,1);                % mean over examples
X0 = X – repmat(mn,[n,1]);     % subtract mean
S = X0' *X0 /n ;               % S = cov(X);
[V,D] = eig(S);                % can be slow!
v = V(: , end-k+1:end);        % k largest eigenvectors
                               %  can also find incrementally ("eigs")
```

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

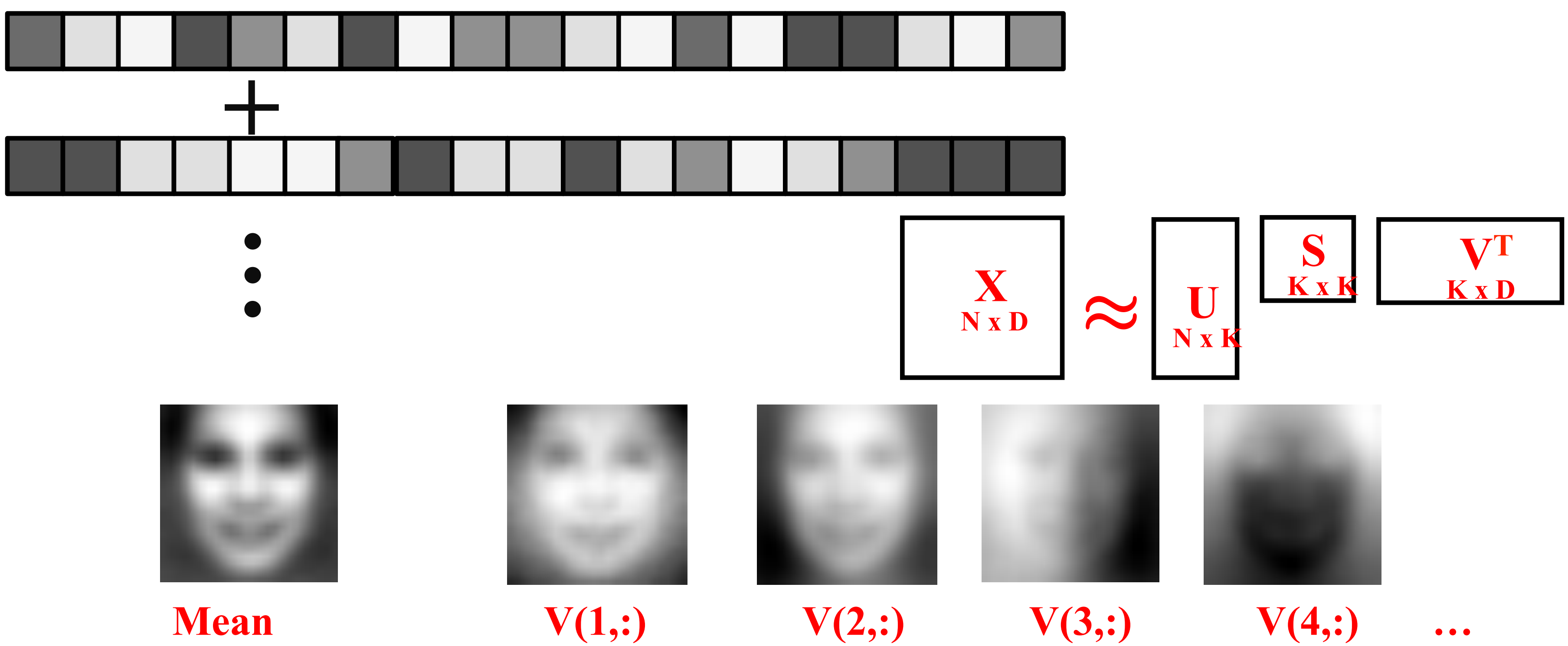


"Eigen-faces"

- "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

Mean **Dir 1** **Dir 2** **Dir 3** **Dir 4** **…**

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 order
- Example:

**Rain and chilly weather didn't keep thousa
paradegoers from camping out Friday nigh
of Roses.**

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

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

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

**### example1/20000101.0015.txt
rain
chilly
weather
didn
keep
thousands
paradegoers
camping
out
friday
night
111th
tournament
roses
spirits**

Text representations

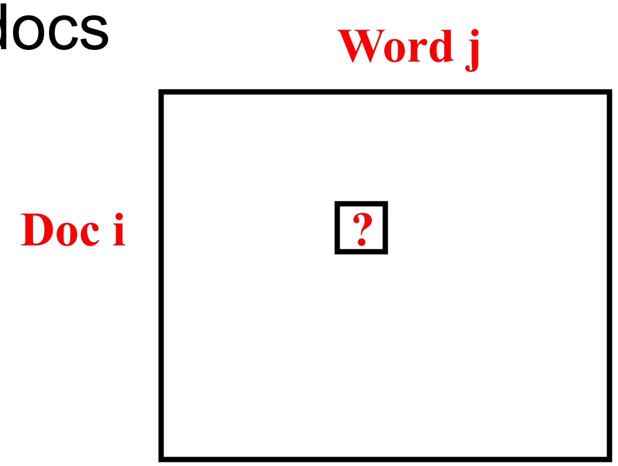
- "Bag of words"
  - Remember word counts but not order
- Example:

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
….

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
…		

Latent Semantic Indexing (LSI)

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

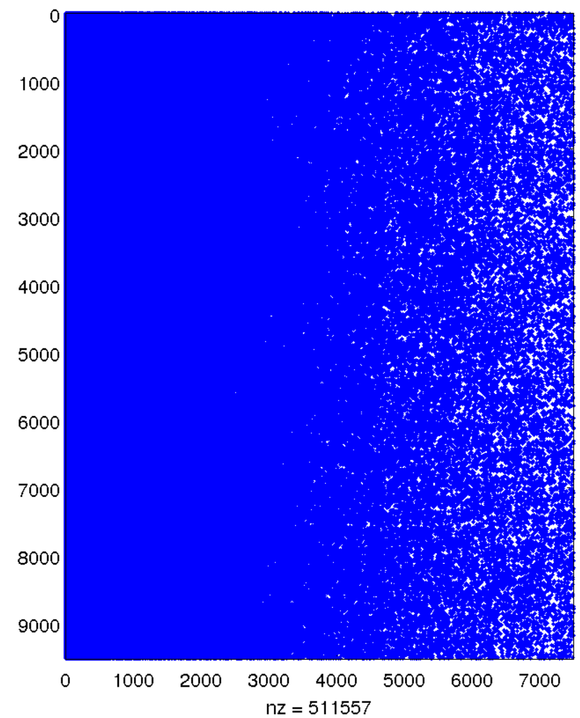


- Huge matrix (mostly zeros)
  - Typically normalize by e.g. sum over j to control for short docs
  - Typically don't subtract mean or normalize by variance
  - Might transform counts in some way (log, etc)
- PCA on this matrix provides a new representation
  - Document comparison
  - Fuzzy search ("concept"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

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 ~ 10^9

- DxW matrix (# docs x # words)
  - Looks dense, but that's just plotting
  - Each entry is non-negative
  - Typically integer / count data



Latent Semantic Indexing (LSI)

- 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

Latent Semantic Indexing (LSI)

- What do the principal components look like?

PRINCIPAL COMPONENT 1	PRINCIPAL COMPONENT 2
0.135 genetic	0.247 snp
0.134 gene	-0.196 cell
0.131 snp	0.187 variant
0.129 disease	0.181 risk
0.126 genome_wide	0.180 gwas
0.117 cell	0.162 population
0.110 variant	0.162 genome_wide
0.109 risk	0.155 genetic
0.098 population	0.130 loci
0.097 analysis	-0.116 mir
0.094 expression	-0.116 expression
0.093 gene_expression	0.113 allele
0.092 gwas	0.108 schizophrenia
0.089 control	0.107 disease
0.088 human	-0.103 mirnas
0.086 cancer	-0.099 protein

Q: But what does -0.196 cell mean?

From Y. Koren
of BellKor team

Collaborative Filtering (Netflix)

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	

$$\underset{N \times D}{X} \approx \underset{N \times K}{U} \; \underset{K \times K}{S} \; \underset{K \times D}{V^T}$$

From Y. Koren
of BellKor team

Latent Space Models

Model ratings matrix as "user" and "movie" positions

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

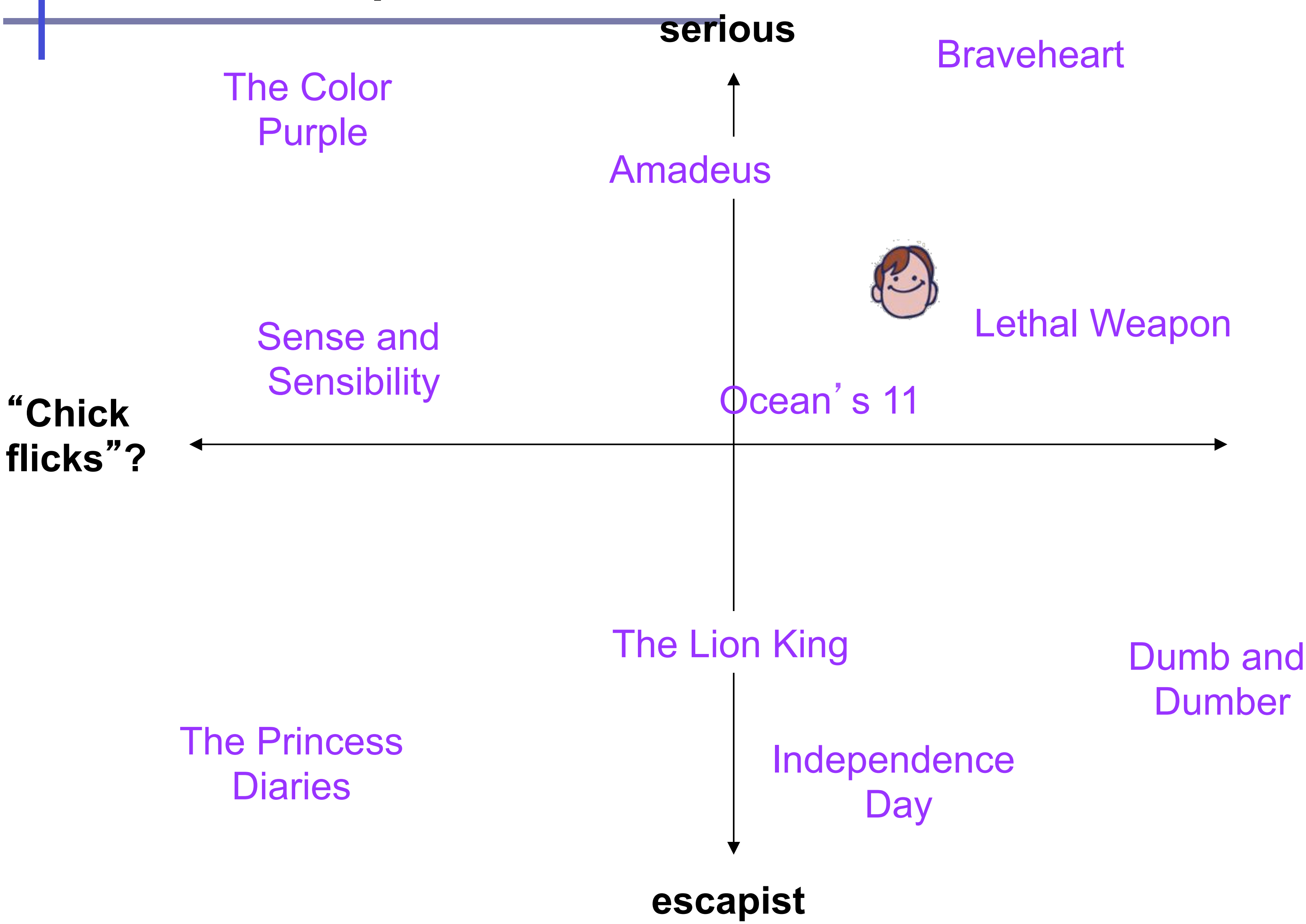
•

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

From Y. Koren of BellKor team

Latent Space Models



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
  - Typically solve using gradient descent
  - Easy algorithm (see Netflix challenge forum)

```
%For user u, movie m, find kth eigenvector & coefficient by iterating:
err = ( rating – predict(m,u) );          % predict vector-vector product
  % predict(m,u) = U(m,:) * V(:,u)
Vku = V(k,u); Umk=U(m,k);                 % save before changing
U(m,k) = U(m,k) + alpha*err*Vku;          % Update our matrices
V(k,u) = V(k,u) + alpha*err*Umk;          % (compare to least-squares gradient)
```

Summary

- Dimensionality reduction
  - Representation: basis vectors & coefficients

- Linear decomposition
  - PCA / eigendecomposition
  - Singular value decomposition

- Examples and data sets
  - Face images
  - Text documents (latent semantic indexing)
  - Movie ratings