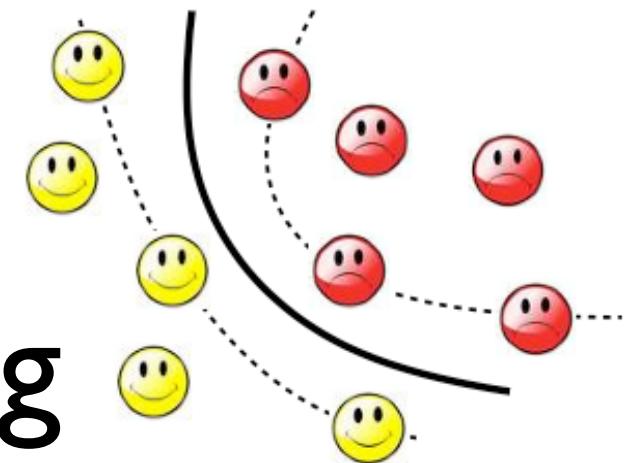


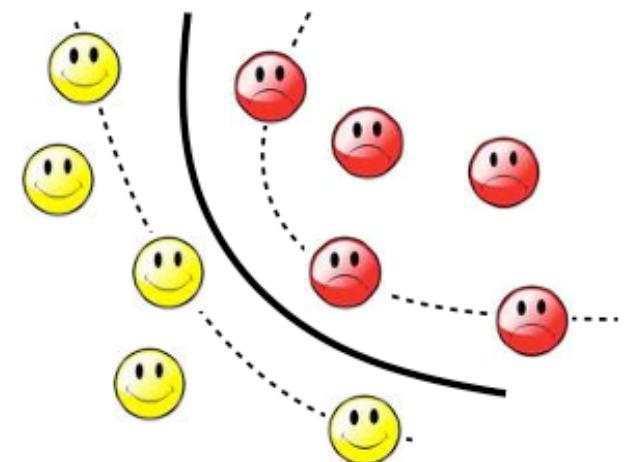
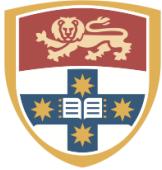
Machine Learning and Data Mining

COMP 5318



Basic Matrix Analysis and Singular Value
Decomposition

Fabio Ramos
Lionel Ott
Roman Marchant



Review



What is Machine Learning?

Informally: Knowledge discovery from data

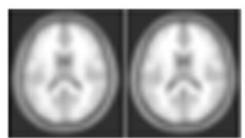
Formally: The construction of a statistical model that is an underlying distribution from which the data is drawn from.



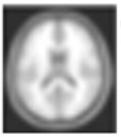
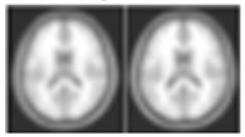
THE UNIVERSITY OF
SYDNEY

Elements of Machine Learning

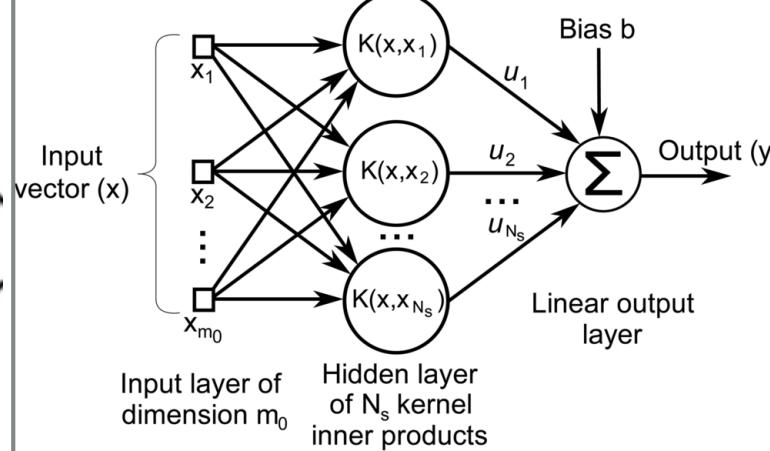
Group 1



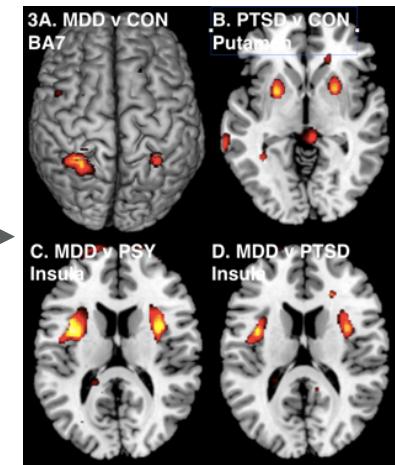
Group 2



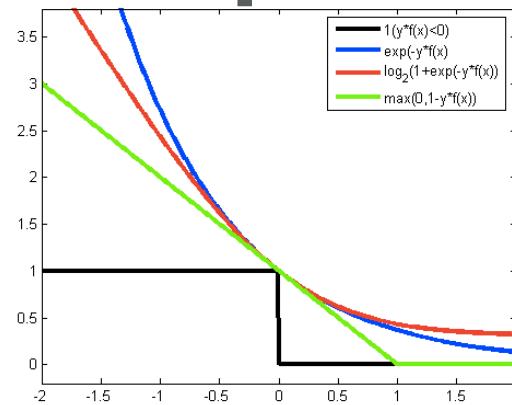
Mathematical Model



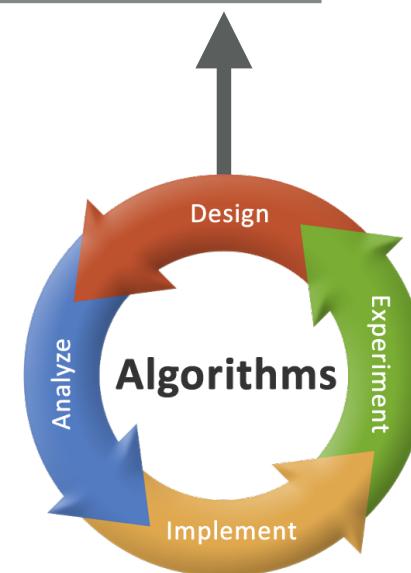
Predictions/Patterns



Data



Objective function

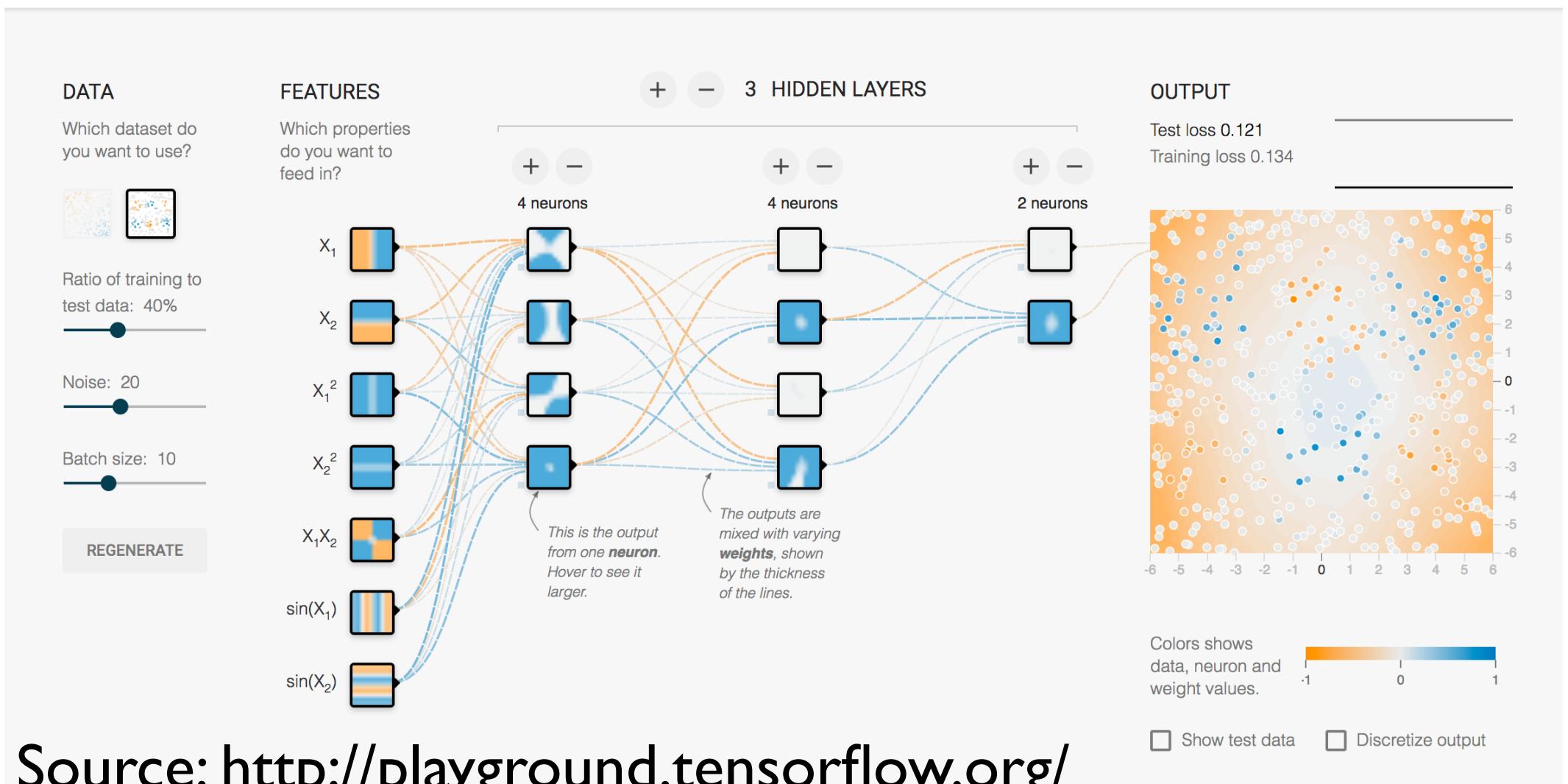




THE UNIVERSITY OF
SYDNEY



Neural Networks



Source: <http://playground.tensorflow.org/>



Common representation

IMAGE/
VIDEO

TEXT/
COMMENT

TIME
SERIES

SYSTEM
LOGS

NETWORK

TABULAR/
RATING

Is there a common way to represent data
of different modalities ?



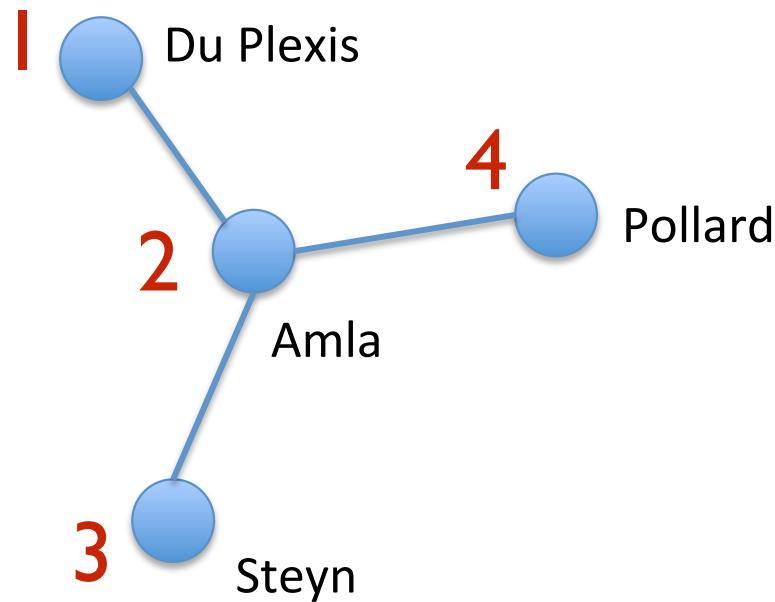
Text to matrix

- Document- Word Matrix
- Document 1: “AACCBBAAA”
- Document 2: “CCAABBDD”

$$\begin{pmatrix} A & B & C & D \\ 5 & 2 & 2 & 0 \\ 2 & 2 & 2 & 2 \end{pmatrix}$$



Network data



Nodes	Nodes	Nodes	Nodes	Nodes
0	1	0	0	0
1	0	1	1	1
0	1	0	0	0
0	1	0	0	0



THE UNIVERSITY OF
SYDNEY

Image data



www.sydney.visitorsbureau.com.au



700 x 500

4	45	6
6	12	33
22	17	44



4	45	6	6	12	33	22	17	44
---	----	---	---	----	----	----	----	----



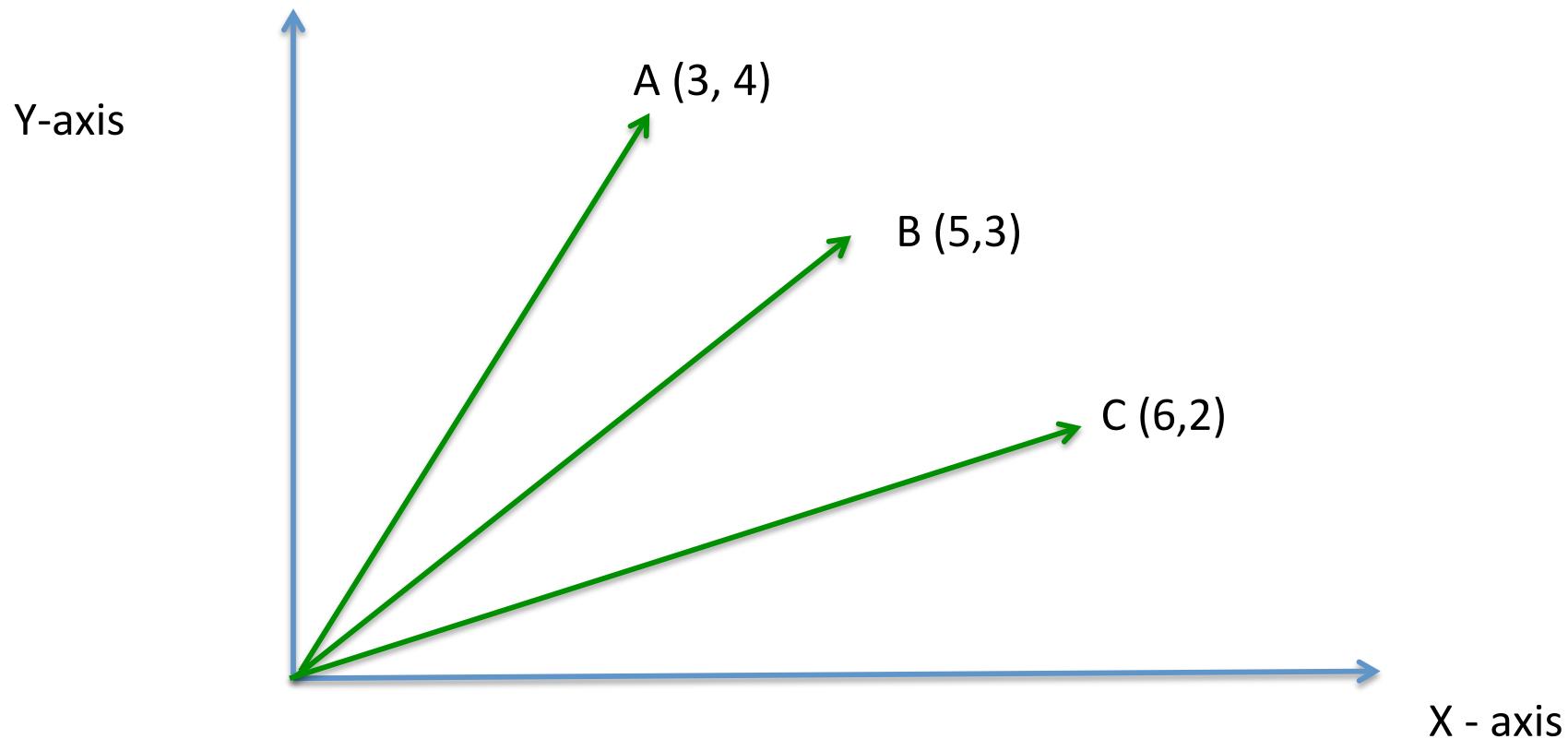
Similarity Computation

- We now can represent most data types as a matrix.
- A special case of a matrix is a vector.
- Now lets compute similarities with these objects.



Similarity Computation

How can we quantify
similarity between A, B and C ?





Similarity Computation

- Dot product

$$\mathbf{x} = (x_1, x_2, \dots, x_n); \mathbf{y} = (y_1, y_2, \dots, y_n)$$

$$\mathbf{x} \cdot \mathbf{y} = (x_1 y_1 + x_2 y_2 + \dots + x_n y_n)$$

- Norm (length) of a vector

$$\|\mathbf{x}\| = (\mathbf{x} \cdot \mathbf{x})^{1/2} = (x_1^2 + x_2^2 + \dots + x_n^2)^{1/2}$$

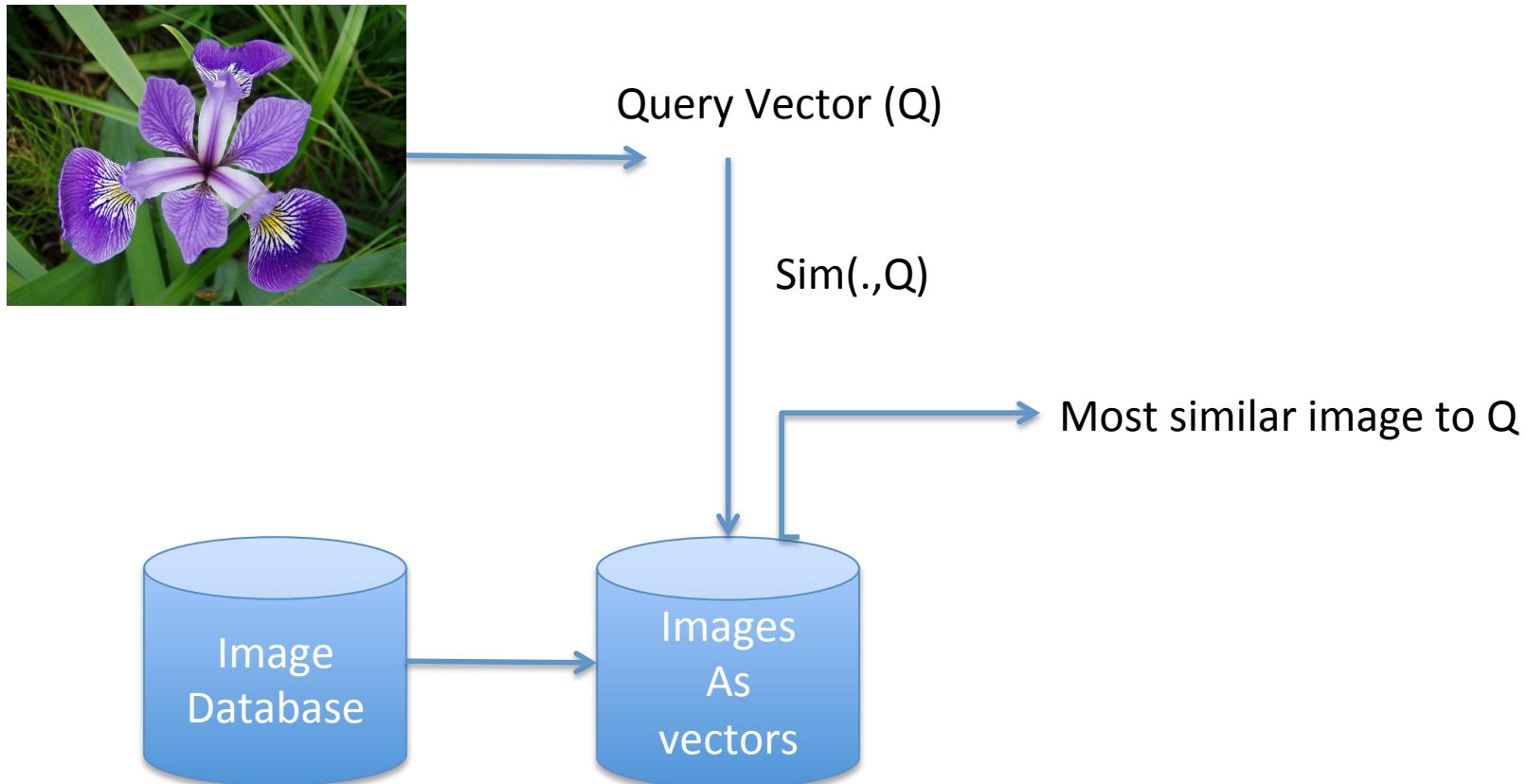


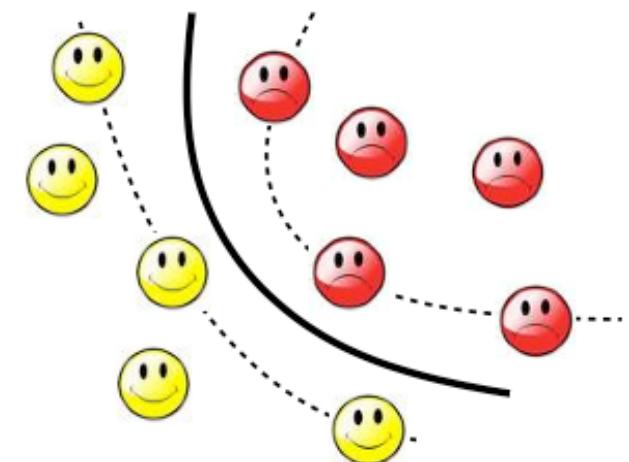
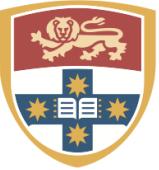
Similarity Computation

- The similarity between two vectors x and y is given by
 - $\text{sim}(x,y) = \mathbf{x} \cdot \mathbf{y} / (\|\mathbf{x}\| \|\mathbf{y}\|)$



Image search engine





Matrix Algebra and Decompositions



Why Matrix Algebra?

- Data Mining: Computation for large data sets
- Key idea: Data Reduction leads to ``Knowledge Discovery''
- Matrix algebra provides simple algorithms with great power for data reduction or data summarisation.

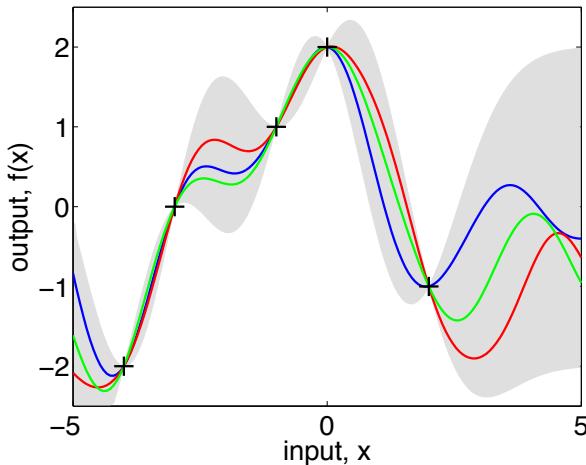


ML algorithms as linear algebra operations

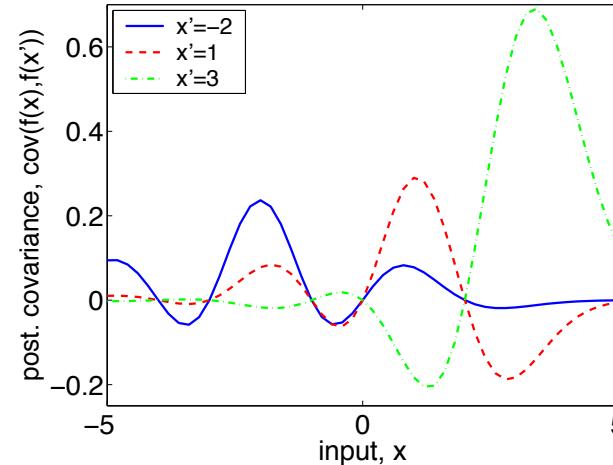
input: X (inputs), \mathbf{y} (targets), k (covariance function), σ_n^2 (noise level),
 \mathbf{x}_* (test input)

- 2: $L := \text{cholesky}(K + \sigma_n^2 I)$
 - $\boldsymbol{\alpha} := L^\top \backslash (L \backslash \mathbf{y})$
 - 4: $\bar{f}_* := \mathbf{k}_*^\top \boldsymbol{\alpha}$
 - $\mathbf{v} := L \backslash \mathbf{k}_*$
 - 6: $\mathbb{V}[f_*] := k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{v}^\top \mathbf{v}$
 - $\log p(\mathbf{y}|X) := -\frac{1}{2}\mathbf{y}^\top \boldsymbol{\alpha} - \sum_i \log L_{ii} - \frac{n}{2} \log 2\pi$
 - 8: **return:** \bar{f}_* (mean), $\mathbb{V}[f_*]$ (variance), $\log p(\mathbf{y}|X)$ (log marginal likelihood)
- } predictive mean eq. (2.25)
} predictive variance eq. (2.26)
eq. (2.30)

Algorithm 2.1: Predictions and log marginal likelihood for Gaussian process regression. The implementation addresses the matrix inversion required by eq. (2.25) and (2.26) using Cholesky factorization, see section A.4. For multiple test cases lines 4-6 are repeated. The log determinant required in eq. (2.30) is computed from the Cholesky factor (for large n it may not be possible to represent the determinant itself). The computational complexity is $n^3/6$ for the Cholesky decomposition in line 2, and $n^2/2$ for solving triangular systems in line 3 and (for each test case) in line 5.



(a), posterior



(b), posterior covariance



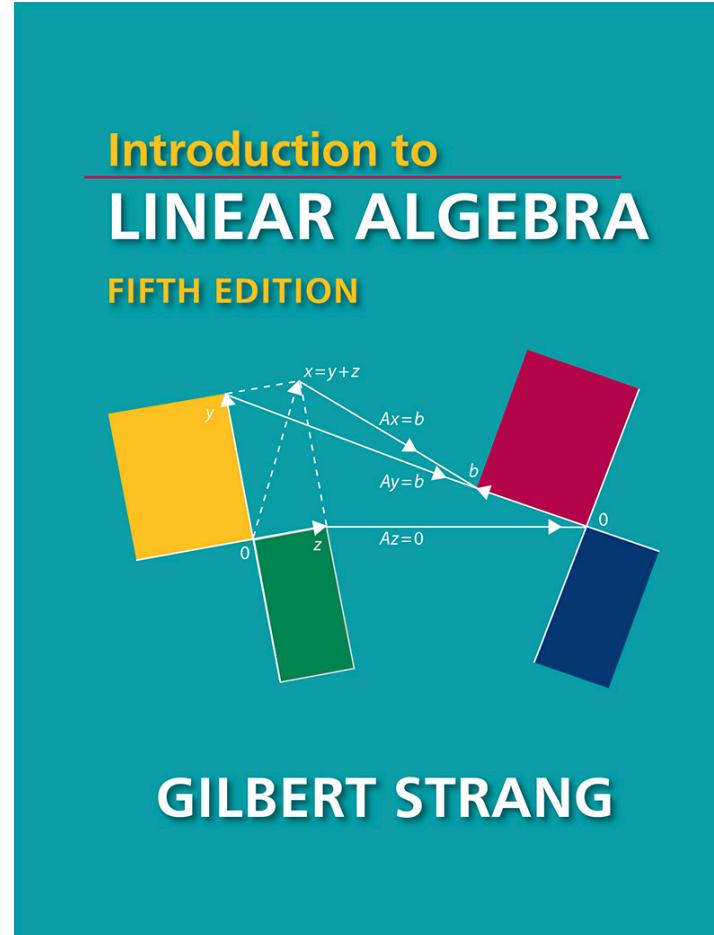
Matrix Algebra

- In database management, the fundamental entity is a *relation*.
 - SQL (relational algebra) is a collection of operations on relations – Select, project, join, group-by
- In machine learning, the fundamental entity is a *matrix*
 - We therefore need to know the basic operations on matrices
 - Our goal is to summarise data or extract patterns from data or compress data
 - In SIT several people apply data mining for different domains: Chinese medicine, bioinformatics, student learning, multimedia, image processing, quality of cloud computing service, text analysis, medical imaging, robotics



THE UNIVERSITY OF
SYDNEY

Suggested book



<http://math.mit.edu/~gs/linearalgebra/>



Numbers vs Matrices

Numbers

- Can add and subtract numbers
- Multiply numbers
- Divide two number a/b as long as b is not equal to 0.
- Can factorize positive numbers into product of primes.

Matrices

- Can add and subtract compatible matrices
- Multiply and divide matrices.
- Division of matrices is complicated
- Can factorize any matrix to get data patterns (using a technique called Singular Value Decomposition)



Linear Algebra

- Area in maths that deals with *vector spaces* and *linear mappings* between these spaces
- The generalisation of LA to infinite dimensions is known as *functional analysis*

2	4	7	3	6
---	---	---	---	---

$$D = 5$$

Linear algebra

2	4	7	3	6	9	...
---	---	---	---	---	---	-----

$$D = \infty$$

Functional analysis



Basics I

$$S = \begin{bmatrix} 3 & 0 & 1 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

- This is 3×3 matrix.
- In general $m \times n$.
 - m rows and n columns
 - Square matrix when $m=n$
- Each row or column could represent one object. If rows are objects then columns are features/attributes/components



Basics II

- Identity matrix I

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- If A is a square matrix, $AI = IA = A$
- I is an example of a **diagonal** matrix.
- If $A = [a_1 \dots a_m]$ is matrix where a_i are the columns. Then
 - A is orthogonal if $a_i \cdot a_j = 0$ for $i \neq j$
 - A is orthonormal if above and $a_i \cdot a_i = 1$



Basics III

- Every vector can be written as a linear combination of some finitely many “special” vectors.
 - These are called basis-vectors

$$S = \begin{bmatrix} 3 \\ 2 \\ 2 \end{bmatrix} = 3 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + 2 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + 2 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$



Linear independence

- Intuitively, a set of vectors is linearly independent if any element of the set cannot be expressed as a linear combination of the others.
- The columns are not linearly independent:

$$S = \begin{bmatrix} 3 & 0 & 1 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$



Determinant

- The determinant of a square matrix is a single number telling us if the matrix is invertible, $\det A^{-1} = 1/(\det A)$
- For 2 by 2 matrix

$$|A| = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc.$$

- For 3 by 3 matrix

$$\begin{aligned}|A| &= \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = a \begin{vmatrix} e & f \\ h & i \end{vmatrix} - b \begin{vmatrix} d & f \\ g & i \end{vmatrix} + c \begin{vmatrix} d & e \\ g & h \end{vmatrix} \\ &= aei + bfg + cdh - ceg - bdi - afh.\end{aligned}$$



Rank of a matrix I

- Given a matrix X , the **rank** of a matrix is the maximum number of linearly independent columns.

- A rank 2 matrix:

$$S = \begin{bmatrix} 3 & 0 & 1 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$



Rank of a matrix II

- Can we automatically determine the rank of a matrix

$$S = \begin{bmatrix} 3 & 1 & 3.5 \\ 2 & 2 & 3 \\ 4 & 2 & 5 \end{bmatrix}$$

- Why is rank important anyways..?



Transpose of a matrix

- The transpose of a matrix A, denoted A^t is another matrix B such that $B(i,j) = A(j,i)$ [just flip the matrix]
- The transpose of S is S^t

$$S = \begin{bmatrix} 3 & 3 & 1 \\ 0 & 2 & 4 \\ 0 & 0 & 0 \end{bmatrix} \quad S^t = \begin{bmatrix} 3 & 0 & 0 \\ 3 & 2 & 0 \\ 1 & 4 & 0 \end{bmatrix}$$



Exercise

- What is the rank of the following matrix:

$$\begin{bmatrix} 1 & 2 & 3 \\ 3 & 4 & 5 \\ 5 & 4 & 3 \\ 0 & 2 & 4 \\ 1 & 3 & 5 \end{bmatrix}$$



Eigen Decomposition

- For any square matrix A , we say that λ is an eigenvalue and \mathbf{u} is its eigenvector if

$$A\mathbf{u} = \lambda\mathbf{u}, \quad \mathbf{u} \neq 0 .$$

- Stacking up all eigenvectors/values gives

$$AU = U\Lambda = \begin{pmatrix} & & & \\ | & | & \cdots & | \\ u_1 & u_2 & \cdots & u_n \\ | & | & \cdots & | \end{pmatrix} \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{pmatrix}$$



Eigen Decomposition

- If A is symmetric, all its e-vals are real, and all its e-vecs are orthonormal, $u_i^T u_j = \delta_{ij}$
- Hence $U^T U = UU^T = I$, $|U| = 1$.
- and $A = U\Lambda U^T = \sum_{i=1}^n \lambda_i u_i u_i^T$

$$\begin{aligned} A &= \left(\begin{array}{cccc} | & | & & | \\ u_1 & u_2 & \cdots & u_n \\ | & | & & | \end{array} \right) \left(\begin{array}{ccccc} \lambda_1 & & & & \\ & \lambda_2 & & & \\ & & \ddots & & \\ & & & \lambda_n & \end{array} \right) \left(\begin{array}{ccc} - & u_1^T & - \\ - & u_2^T & - \\ \vdots & \vdots & \vdots \\ - & u_n^T & - \end{array} \right) \\ &= \lambda_1 \left(\begin{array}{c} | \\ u_1 \\ | \end{array} \right) (- \ u_1^T \ -) + \cdots + \lambda_n \left(\begin{array}{c} | \\ u_n \\ | \end{array} \right) (- \ u_n^T \ -) \end{aligned}$$



Exercise

- Find the eigenvectors and eigenvalues for the following matrix

$$\begin{bmatrix} 3 & 2 \\ 2 & 6 \end{bmatrix}$$



Singular Value Decomposition

- Given **any** real matrix X of size (m,n) , it can be expressed as:

$$X = U\Sigma V^T$$

The diagram shows the SVD formula $X = U\Sigma V^T$. Below the formula, three boxes indicate dimensions: the first box contains $m \times r$, the second $r \times r$, and the third $r \times n$. Arrows point from each dimension box to its corresponding term in the formula: the $m \times r$ box points to U , the $r \times r$ box points to Σ , and the $r \times n$ box points to V^T .

- r is the rank of matrix X
- U is a (m,r) column-orthonormal matrix
- V is a (n, r) column-orthonormal matrix
- Σ is diagonal $r \times r$ matrix



SVD in Python

- One line command:

```
>>> U, s, V = np.linalg.svd(a, full_matrices=True)
```

	Wed	Thur	Friday	Sat	Sun
ABC Ltd	1	1	1	0	0
DEF Ltd	2	2	2	0	0
GHI Inc	1	1	1	0	0
KLM Co	5	5	5	0	0
Smith	0	0	0	2	2
Johnson	0	0	0	3	3
Thompson	0	0	0	1	1

$$X = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \times \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \times \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$



Adhoc Querying on X

- Suppose X was very very large..[all mobile users in China]
- Now the query is “Find the amount spent by Jack Ma on Friday”
 - The original answer $X(\text{JackMa}, \text{Friday})$
 - The approximate answer $\hat{X}(\text{JackMa}, \text{Friday})$
 - Is it a good approximation ?



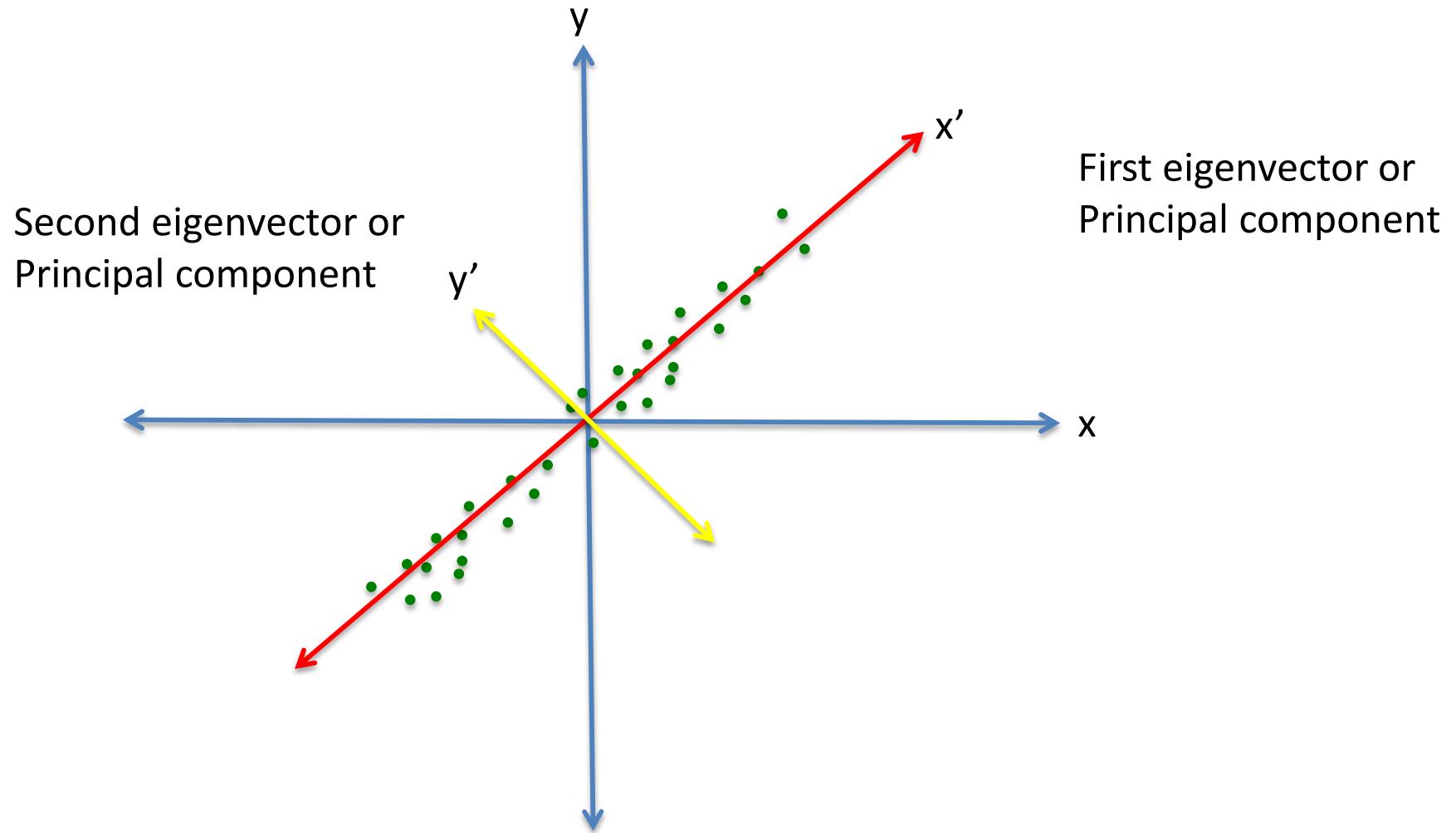
Qualitative use of SVD

$$\mathbf{X} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \times \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \times \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

- Two kinds of customers (businesses and individuals)
- Two kinds of days (weekday and weekends)
- \mathbf{U} is a customer-pattern matrix
- \mathbf{V} is a day-pattern matrix
- $V(1,2) = 0$ means Wednesday has zero similarity with the “weekend pattern.”

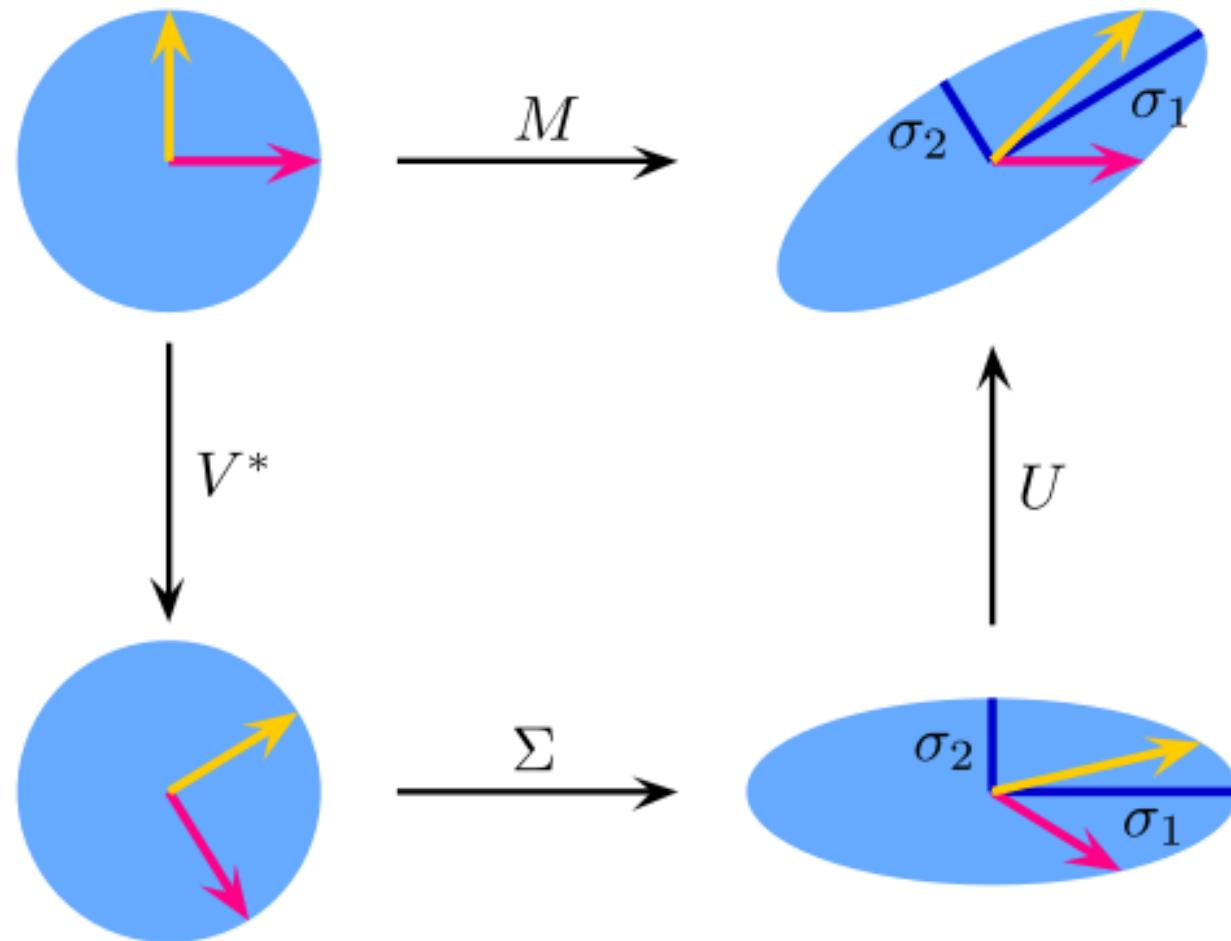


Geometric intuition





SVD as a sequence of operations



$$M = U \cdot \Sigma \cdot V^*$$



Spectral representation

- Matrix X can also be written as:

$$X = \lambda_1 u_1 \times v_1^t + \lambda_2 u_2 \times v_2^t + \dots + \lambda_r u_r \times v_r^t$$

- The above is called **spectral** representation.

$$X = 9.64 \times \begin{bmatrix} 0.18 \\ 0.36 \\ 0.18 \\ 0.90 \\ 0 \\ 0 \\ 0 \end{bmatrix} \times [0.58, \quad 0.58, \quad 0.58, \quad 0, \quad 0] + 5.29 \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.53 \\ 0.80 \\ 0.27 \end{bmatrix} \times [0, \quad 0, \quad 0.71, \quad 0.71]$$



Data reduction with SVD

- Now how can we use an SVD decomposition..
- The first way is qualitative..
 - Customer-day Pattern; or Customer-Pattern-day matrix..
- The second way is quantitative..
 - When X is very large, we can compress X into a smaller matrix but still retain important information..



Example: compression of X

$$X = U \Sigma V^T$$

The diagram illustrates the dimensions of the matrices in the Singular Value Decomposition (SVD) of matrix X. Matrix X is shown as a large rectangle at the top, with arrows pointing down to three smaller rectangles below it. The first rectangle is labeled 'm x r', the second is 'r x r', and the third is 'r x n'.

- Size of $X = mn$
- Size of $U + \Sigma + V$ is $mr + r + nr$
- Thus compression ratio is

$$\frac{mr + r + nr}{mn} = \frac{r(m+1+n)}{mn} \approx \frac{rm}{mn} = \frac{r}{n}$$

- Can we do better ?



Example: compression of X

- To get better compression we should look at the λ values.
 - These are called singular values.
- We can arrange them in descending order:
 - $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_r > 0$
- Now recall...

$$X = \lambda_1 u_1 \times v_1^t + \lambda_2 u_2 \times v_2^t + \dots + \lambda_r u_r \times v_r^t$$



Example: compression of X

- Now a compact way of writing the spectral representation is:

$$\mathbf{X} = \sum_{i=1}^r \lambda_i \mathbf{u}_i \times \mathbf{v}_i^t$$

- However, can approximate it as

$$\widehat{\mathbf{X}} = \sum_{i=1}^k \lambda_i \mathbf{u}_i \times \mathbf{v}_i^t$$

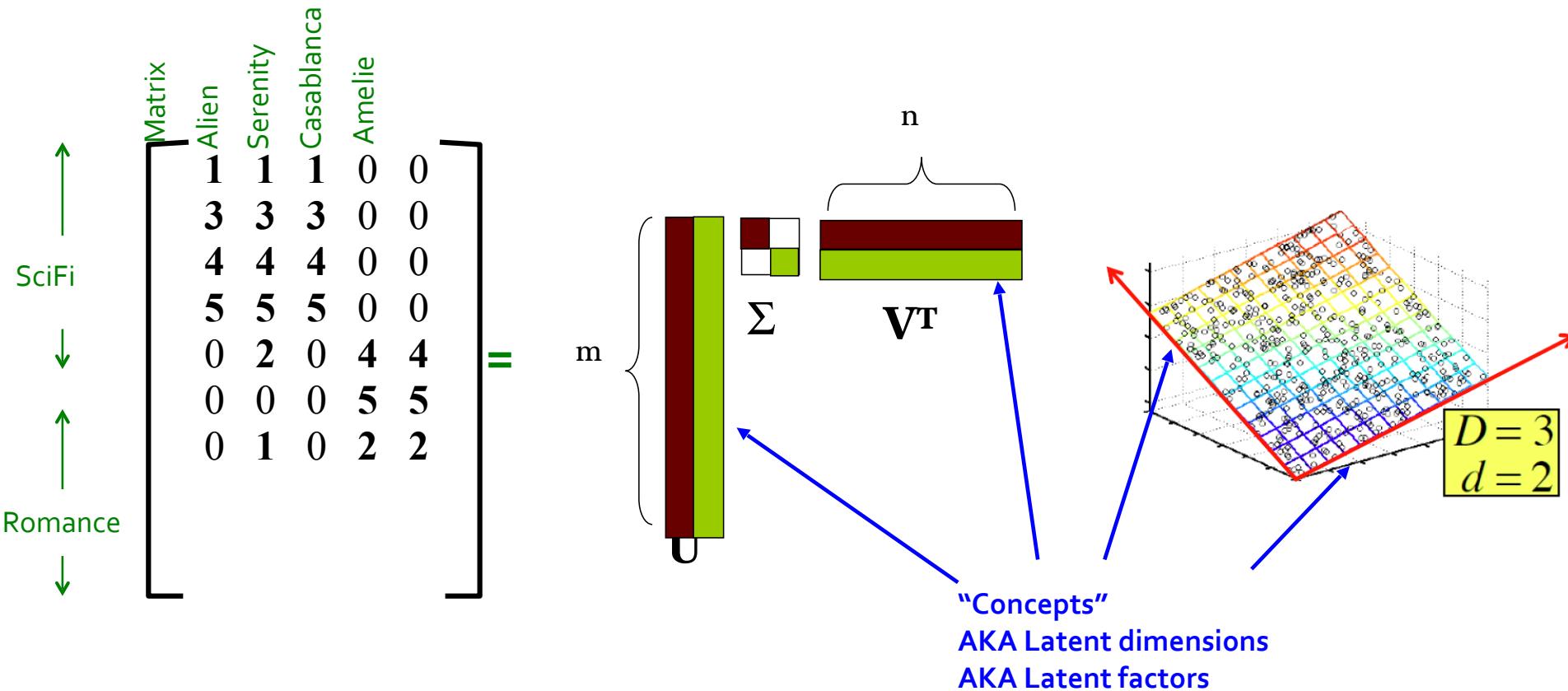
- Thus new compression ratio is:

$$\frac{mk + k + nk}{mn} = \frac{k(m+1+n)}{mn} \approx \frac{km}{mn} = \frac{k}{n} \leq \frac{r}{n}$$



Example: Users to Movies

- $A = U \Sigma V^T$ - example: Users to Movies





Example: Users to Movies

- $A = U \Sigma V^T$ - example: Users to Movies

U is “user-to-concept”
similarity matrix

Matrix

	Alien	Serenity	Casablanca	Amelie
SciFi	1 1 1 0 0	0.13 0.02 -0.01		
	3 3 3 0 0	0.41 0.07 -0.03		
	4 4 4 0 0	0.55 0.09 -0.04		
	5 5 5 0 0	0.68 0.11 -0.05		
	0 2 0 4 4	0.15 -0.59 0.65		
	0 0 0 5 5	0.07 -0.73 -0.67		
Romance	0 1 0 2 2	0.07 -0.29 0.32		

=

	SciFi-concept	Romance-concept
	0.13 0.02 -0.01	
	0.41 0.07 -0.03	
	0.55 0.09 -0.04	
	0.68 0.11 -0.05	
	0.15 -0.59 0.65	
	0.07 -0.73 -0.67	
	0.07 -0.29 0.32	

“strength” of the SciFi-concept

12.4 0 0
0 9.5 0
0 0 1.3

x

0.56 0.59 0.56 0.09 0.09
0.12 -0.02 0.12 -0.69 -0.69
0.40 -0.80 0.40 0.09 0.09

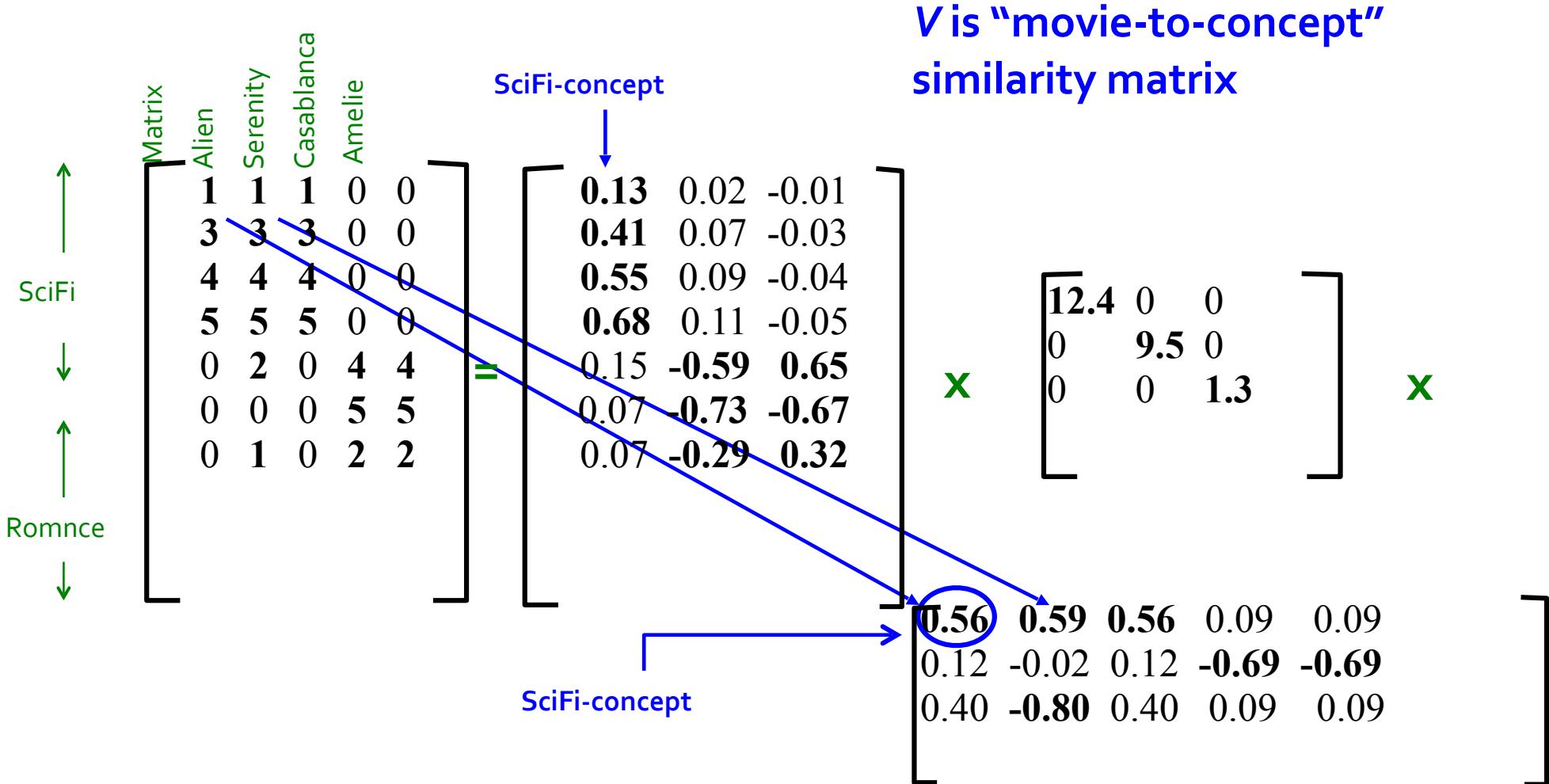
x

[]



Example: Users to Movies

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





Interpretation

‘movies’, ‘users’ and ‘concepts’:

- U : user-to-concept similarity matrix
- V : movie-to-concept similarity matrix
- Σ : its diagonal elements:
‘strength’ of each concept



Tutorials: Monday 8-9pm

- SIT 114 Tutor: Gilad
- SIT 115 Tutor: Prasad
- SIT 116 Tutor: Rafael
- SIT 117 Tutor: Dongang
- SIT 130b Tutor: Philippe
- SIT 437 Tutor: Kelvin



THE UNIVERSITY OF
SYDNEY

Tutorials: Tuesday 5-6pm

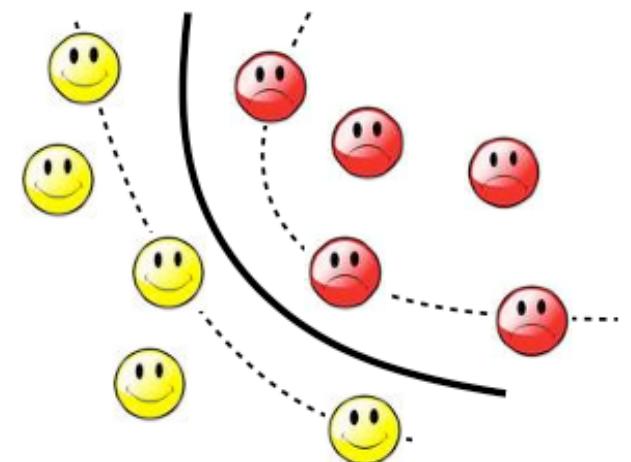
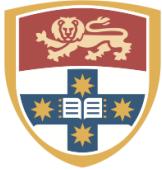
- SIT 115 Tutor: Dongang
- SIT 116 Tutor: Sheila
- SIT 118 Tutor: Prasad



Quiz on Week 4

- 1) If your lab is on Monday 8-9pm, go to your Lab at the usual time.
- 2) If your lab is on Tuesday 5-6pm, the Quiz will be on Monday, 7-8pm. See the table below:

SIT 115	SIT 115, Mon 7pm
SIT 116	SIT 116, Mon 7pm
SIT 118	SIT 114, Mon 7pm



Research Topics