Matrix Decomposition and Latent Semantic Indexing (LSI)

Introduction to Information Retrieval
INF 141
Donald J. Patterson

Last Lecture

Efficient Cosine Ranking

- Find the k docs in the corpus “nearest” to the query
  - the k largest query-doc cosines

```
$\textsc{CosineScore}(q)$
$\textsc{Initialize}(Scores[d \in D])$
$\textsc{Initialize}(Magnitude[d \in D])$
$\textbf{for } each\ term(t \in q)$
    $\textbf{do } p \leftarrow \textsc{FetchPostingsList}(t)$
        $df_t \leftarrow \textsc{GetCorpusWideStats}(p)$
        $\alpha_{t,q} \leftarrow \textsc{WeightInQuery}(t, q, df_t)$
        $\textbf{for } each\ \{d, tf_{t,d}\} \in p$
            $\textbf{do } Scores[d] += \ \alpha_{t,q} \cdot \textsc{WeightInDocument}(t, q, df_t)$
$\textbf{for } d \in Scores$
    $\textbf{do } \textsc{Normalize}(Scores[d], Magnitude[d])$
$\textbf{return } top\ K \in Scores$
```

Outline

- Introduction
- Linear Algebra Refresher

Latent Semantic Indexing - Introduction

Star Cluster NGC 290 - ESA & NASA

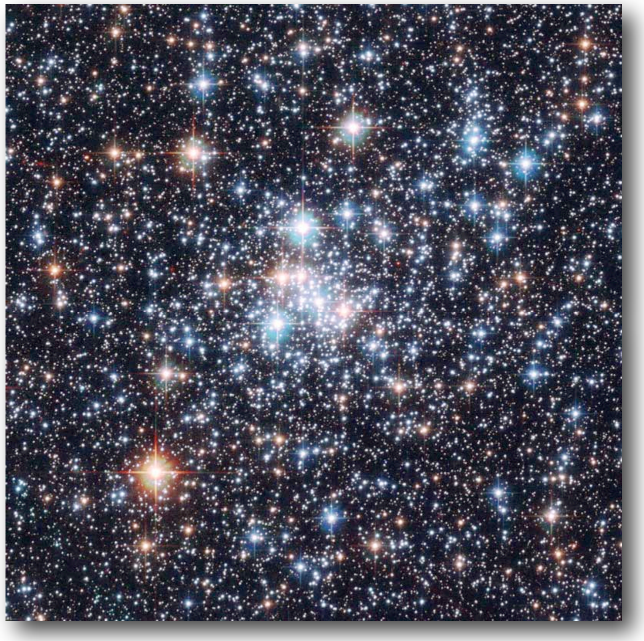
Latent Semantic Indexing - Introduction

Star Cluster NGC 290 - ESA & NASA

- A picture of the sky is two dimensional
- The stars are not in two dimensions
- When we take a photo of stars we are projecting them into 2-D
    - projecting can be defined mathematically
- When we see two stars that are close..
    - They may not be close in space
- When we see two stars that appear far...
    - They may not be far in 3-D space

Star Cluster NGC 290 - ESA & NASA

- A picture of the sky is two dimensional
- The stars are not in two dimensions
- When we take a photo of stars we are projecting them into 2-D
    - projecting can be defined mathematically
- When we see two stars that are close..
    - They may not be close in space
- When we see two stars that appear far...
    - They may not be far in 3-D space



Star Cluster NGC 290 - ESA & NASA

- When we see two stars that are close in a photo
  - They really are close for some applications
  - For example pointing a big telescope at them
  - Large shared telescopes order their views according to how "close" they are.

Latent Semantic Indexing - Introduction

Overhead projector example

Overhead projector example

- Depending on where we put the light (and the wall) we can make things in three dimensions appear close or far away in two dimensions.
- Even though the “real” position of the 3-d objects never moved.

Latent Semantic Indexing - Introduction

Mathematically speaking

- This is taking a 3-D point and projecting it into 2-D

$$\begin{array}{ccc} (x, y, z) & & (x, y) \\ (10, 10, 10) & \Longrightarrow & (10, 10) \\ \begin{bmatrix} 10 \\ 10 \\ 10 \end{bmatrix} & & \begin{bmatrix} 10 \\ 10 \end{bmatrix} \end{array}$$

- The arrow in this picture acts like the overhead projector

Latent Semantic Indexing - Introduction

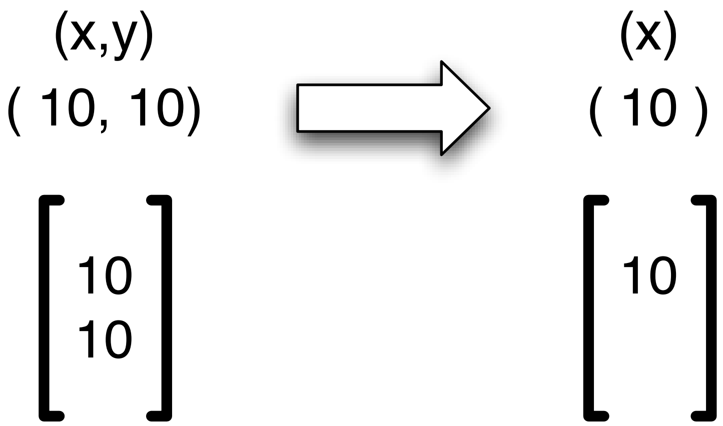
Mathematically speaking

- We can project from any number of dimensions into any other number of dimensions.
- Increasing dimensions adds redundant information
  - But sometimes useful
  - Support Vector Machines (kernel methods) do this effectively
- Latent Semantic Indexing always reduces the number of dimensions

Latent Semantic Indexing - Introduction

Mathematically speaking

- Latent Semantic Indexing always reduces the number of dimensions



(x,y)

(10, 10)

$$\begin{bmatrix} 10 \\ 10 \end{bmatrix}$$

(x)

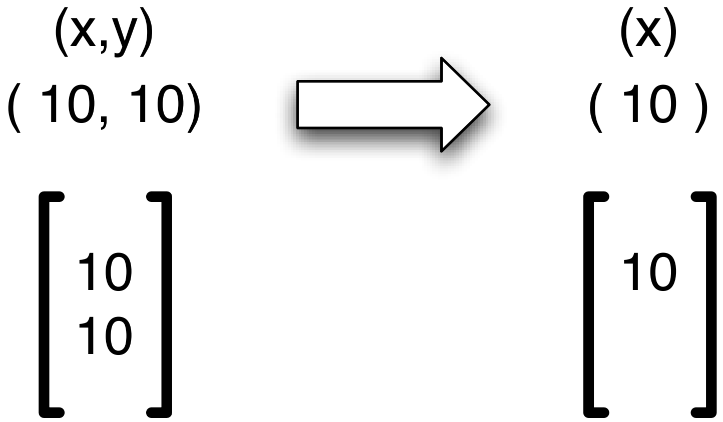
(10)

$$\begin{bmatrix} 10 \end{bmatrix}$$

Latent Semantic Indexing - Introduction

Mathematically speaking

- Latent Semantic Indexing always reduces the number of dimensions



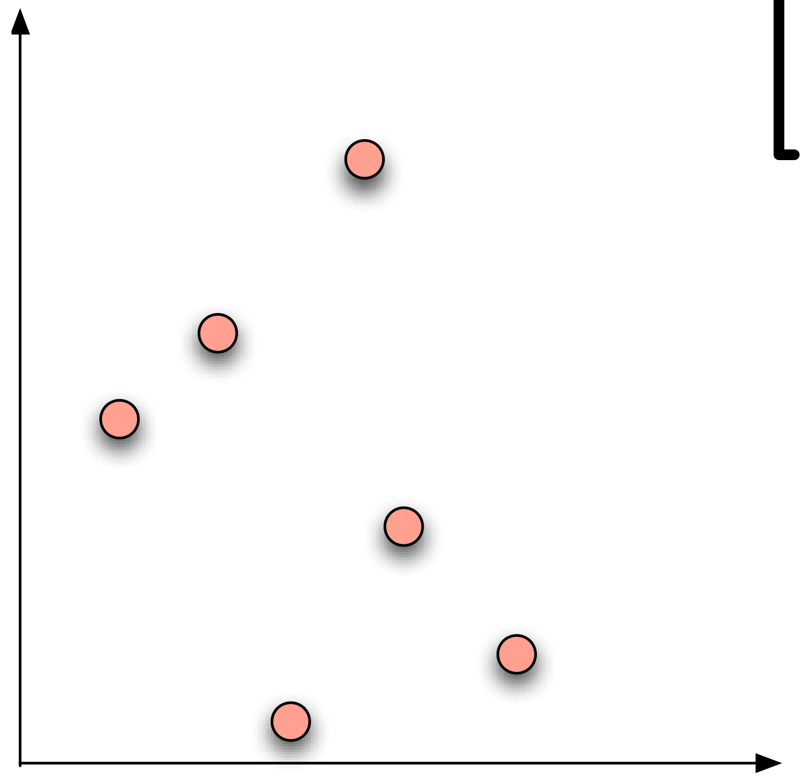
(x,y)

(10, 10)

$$\begin{bmatrix} 10 \\ 10 \end{bmatrix} \longrightarrow \begin{bmatrix} 10 \end{bmatrix}$$

(x)

(10)



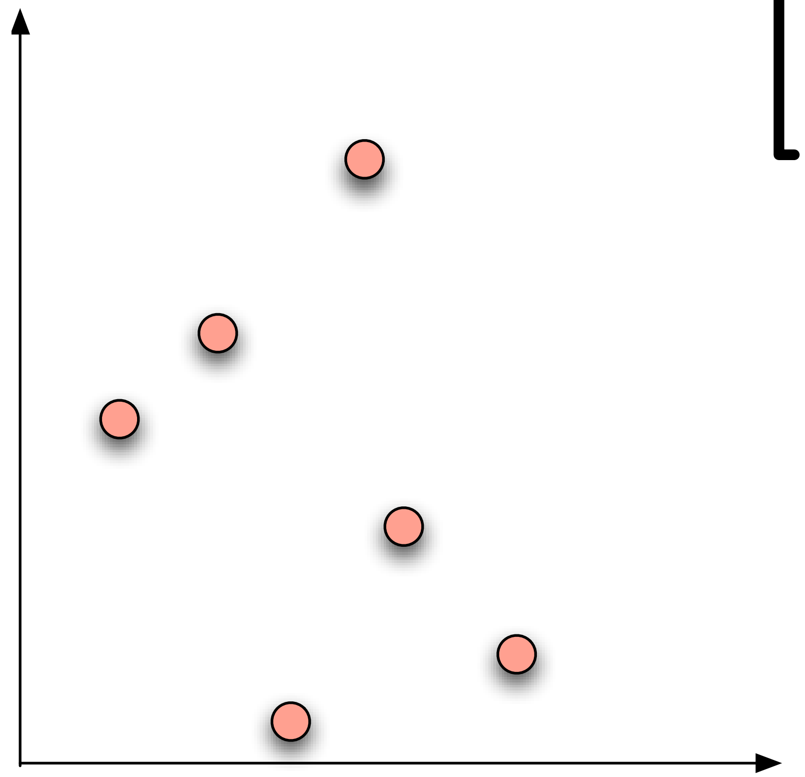
Mathematically speaking

- Latent Semantic Indexing always reduces the number of dimensions

(x,y) → (x)

(10, 10) → (10)

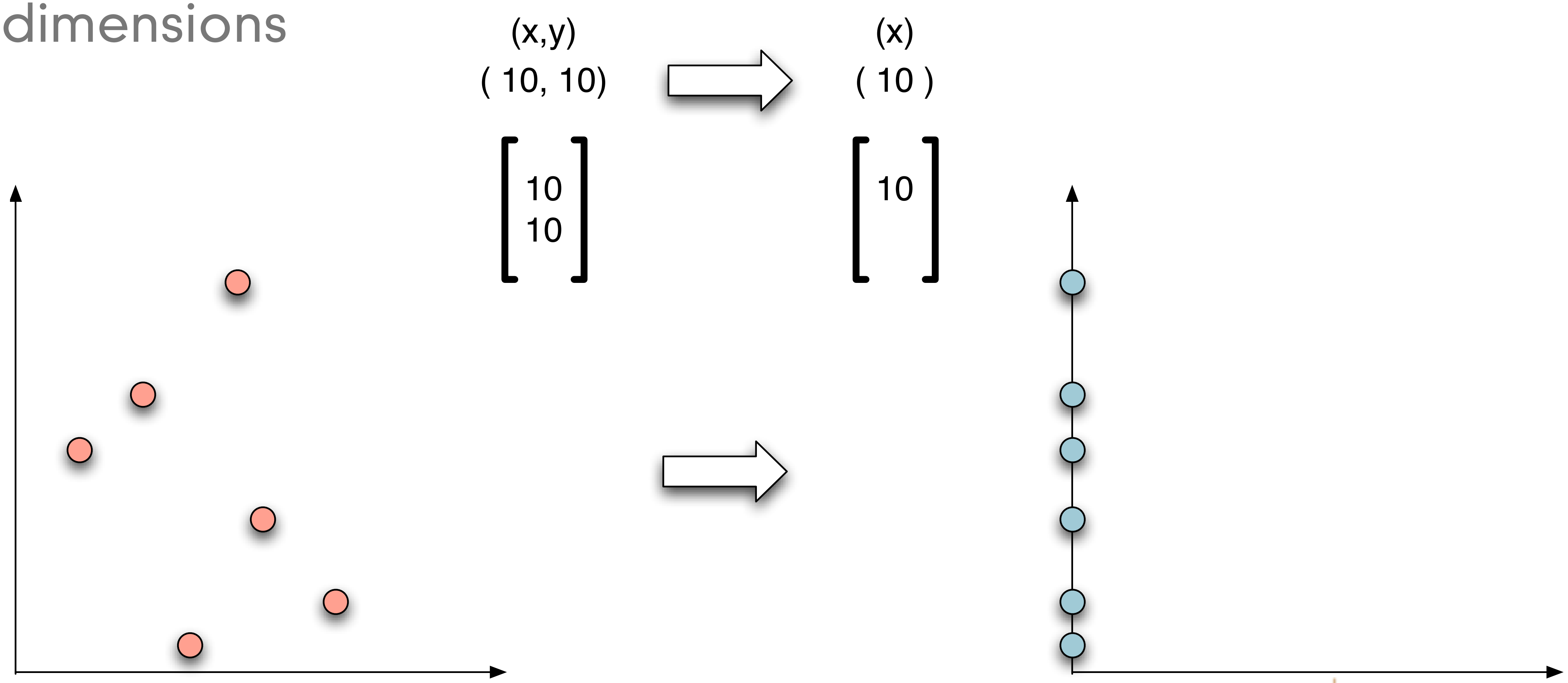
$$\begin{bmatrix}10\\10\end{bmatrix} \qquad \begin{bmatrix}10\end{bmatrix}$$



Latent Semantic Indexing - Introduction

Mathematically speaking

- Latent Semantic Indexing always reduces the number of dimensions



(x,y)

(10, 10)

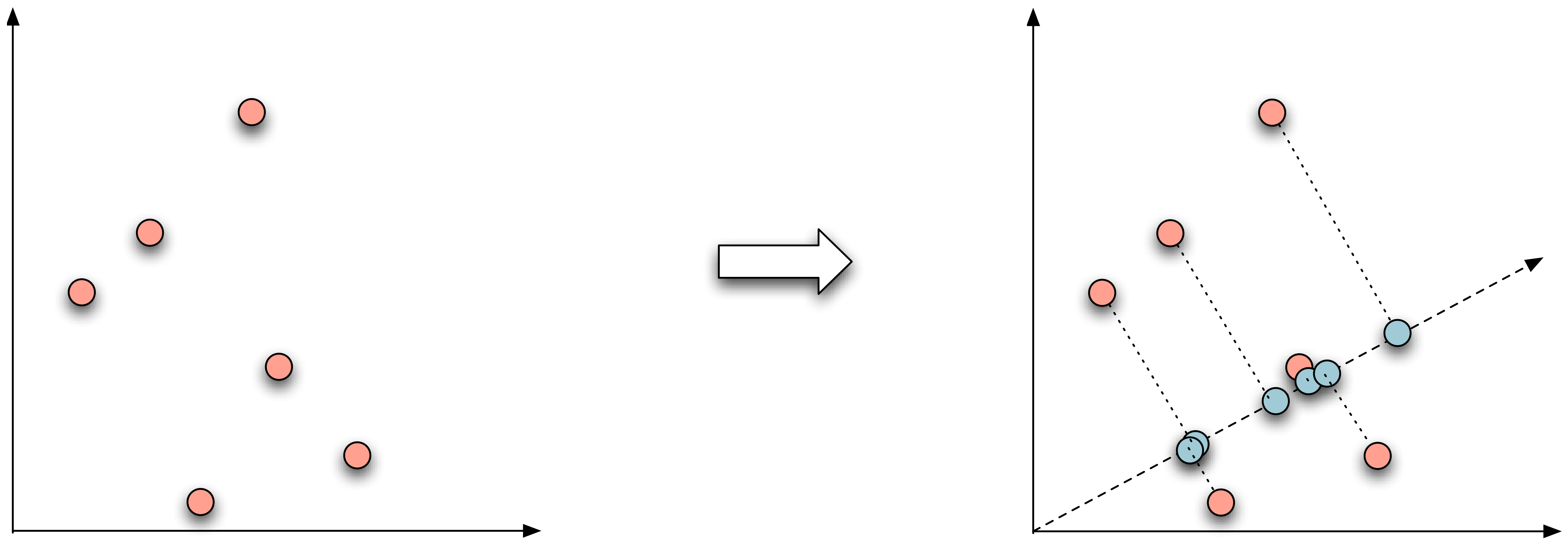
$$\begin{bmatrix} 10 \\ 10 \end{bmatrix} \Rightarrow \begin{bmatrix} 10 \end{bmatrix}$$

(x)

(10)

Mathematically speaking

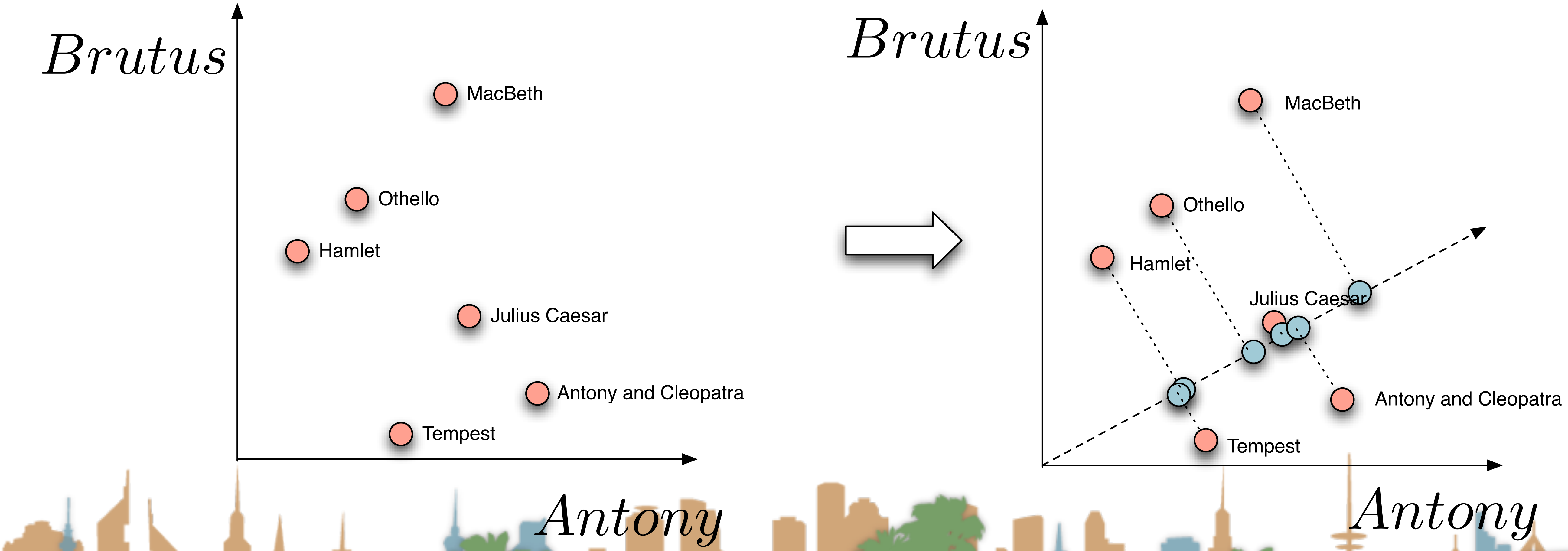
- Latent Semantic Indexing can project on an arbitrary axis, not just a principal axis



Latent Semantic Indexing - Introduction

Mathematically speaking

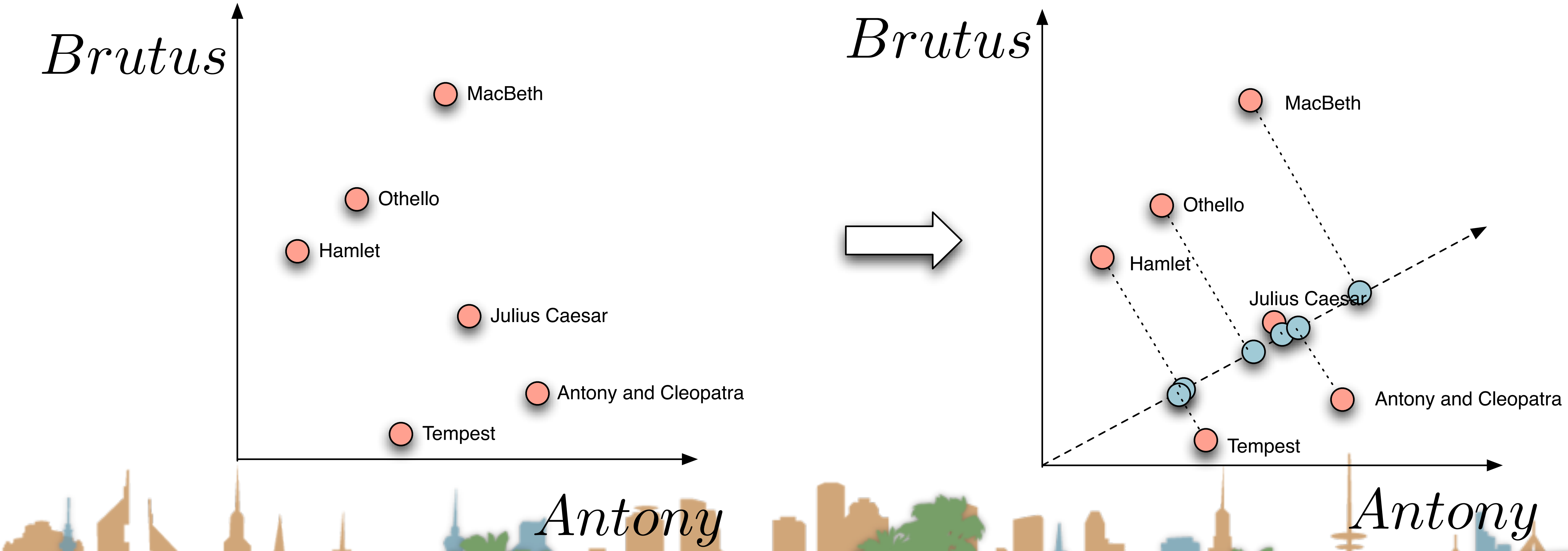
- Our documents were just points in an N-dimensional term space
- We can project them also



Latent Semantic Indexing - Introduction

Mathematically speaking

- Latent Semantic Indexing makes the claim that these new axes represent semantics - deeper meaning than just a term



Mathematically speaking

- A term vector that is projected on new vectors may uncover deeper meanings
  - For example
    - Transforming the 3 axes of a term matrix from “ball” “bat” and “cave” to
      - An axis that merges “ball” and “bat”
      - An axis that merges “bat” and “cave”
    - Should be able to separate differences in meaning of the term “bat”
    - Bonus: less dimensions is faster

Linear Algebra Refresher

- Let C be an M by N matrix with real-valued entries
  - for example our term document matrix
- A matrix with the same number of rows and columns is called a square matrix
- An M by M matrix with elements only on the diagonal is called a diagonal matrix
- The identity matrix is a diagonal matrix with ones on the main diagonal

Linear Algebra Refresher

- Let C be an M by N matrix with real-valued entries
  - for example our term document matrix

$$\begin{array}{cc} & N=5 \\ M=3 & \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 0 & 2 & 3 & 2 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{bmatrix} \\ & C \end{array}$$

- A matrix with the same number of rows and columns is called a square matrix
- An M by M matrix with elements only on the diagonal is called a diagonal matrix
- The identity matrix is a diagonal matrix with ones on the main diagonal

Linear Algebra Refresher

- Let C be an M by N matrix with real-valued entries
  - for example our term document matrix
- A matrix with the same number of rows and columns is called a square matrix
- An M by M matrix with elements only on the diagonal is called a diagonal matrix
- The identity matrix is a diagonal matrix with ones on the main diagonal

$$M=3 \quad \underset{C}{\overset{N=5}{\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 0 & 2 & 3 & 2 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{bmatrix}}}$$

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

Linear Algebra Refresher

- Let C be an M by N matrix with real-valued entries
  - for example our term document matrix
- A matrix with the same number of rows and columns is called a square matrix
- An M by M matrix with elements only on the diagonal is called a diagonal matrix
- The identity matrix is a diagonal matrix with ones on the main diagonal

$$M=3 \; \underset{C}{\overset{N=5}{\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 0 & 2 & 3 & 2 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{bmatrix}}}$$

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

$$\begin{bmatrix} 5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

Linear Algebra Refresher

- Let C be an M by N matrix with real-valued entries
    - for example our term document matrix

N=5

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

C

- A matrix with the same number of rows and columns is called a square matrix

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

- An M by M matrix with elements only on the diagonal is called a diagonal matrix

$$\begin{bmatrix} 5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

- The identity matrix is a diagonal matrix with ones on the main diagonal

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

Matrix Decomposition

- Singular Value Decomposition
  - Splits a matrix into three matrices
  - Such that
    - If
    - then
    - and
    - and
    - also Sigma is almost a diagonal matrix

Matrix Decomposition

- Singular Value Decomposition
  - Splits a matrix into three matrices $U \quad \Sigma \quad V^T$
  - Such that
    - If
    - then
    - and
    - and
    - also Sigma is almost a diagonal matrix

Matrix Decomposition

- Singular Value Decomposition
  - Splits a matrix into three matrices $U \quad \Sigma \quad V^T$
  - Such that $C = U\Sigma V^T$
    - If
    - then
    - and
    - and
    - also Sigma is almost a diagonal matrix

Matrix Decomposition

- Singular Value Decomposition
  - Splits a matrix into three matrices

    $$U \quad \Sigma \quad V^T$$

  - Such that

    $$C \quad = \quad U\Sigma V^T$$

    - If

      $$C \ is \ (M \ by \ N)$$

    - then
    - and
    - and
    - also Sigma is almost a diagonal matrix

Matrix Decomposition

- Singular Value Decomposition
  - Splits a matrix into three matrices $U \quad \Sigma \quad V^T$
  - Such that $C = U\Sigma V^T$
    - If $C \; is \; (M \; by \; N)$
    - then $U \; is \; (M \; by \; M)$
    - and
    - and
    - also Sigma is almost a diagonal matrix

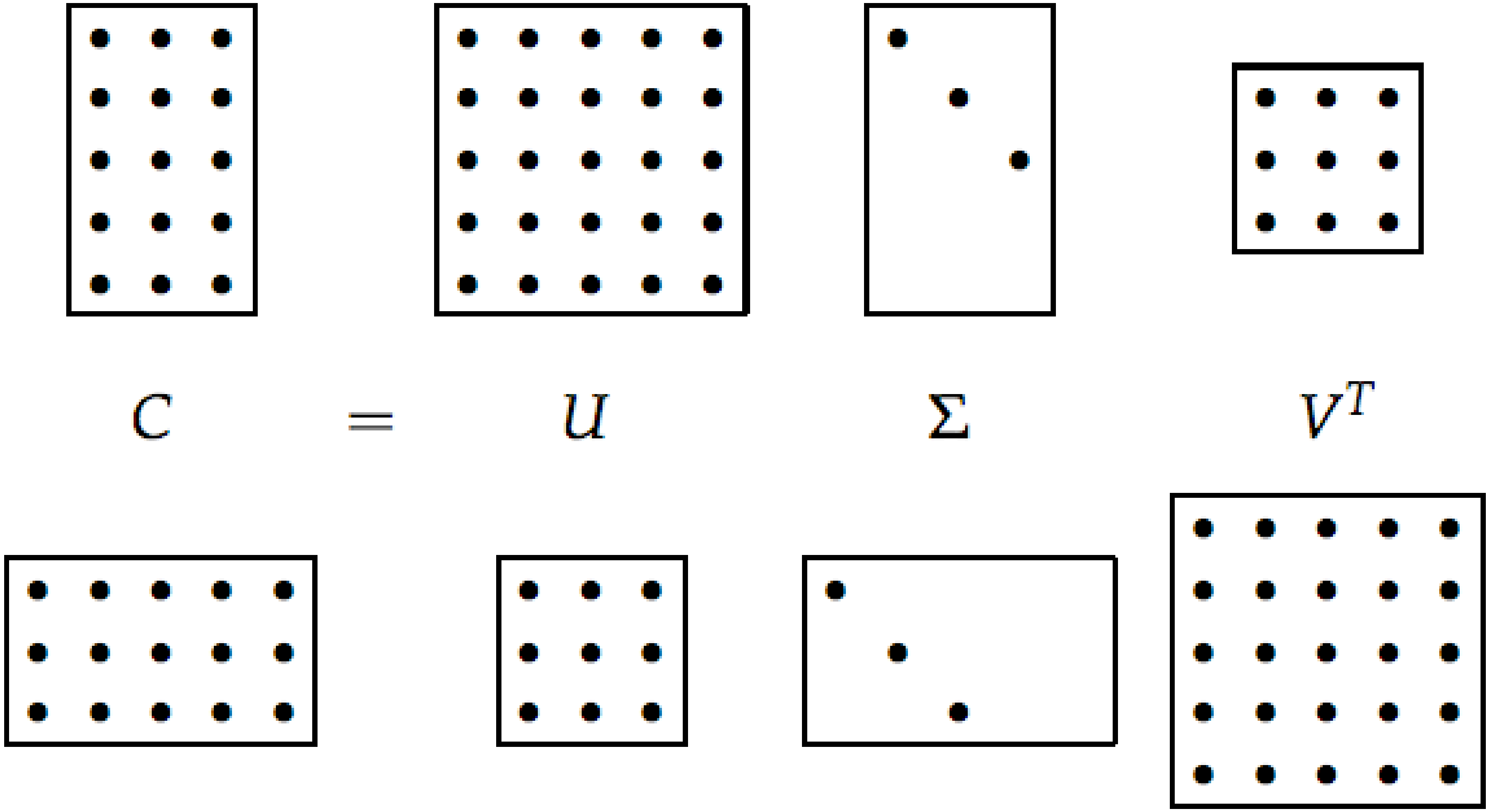
Matrix Decomposition

- Singular Value Decomposition
  - Splits a matrix into three matrices $U \quad \Sigma \quad V^T$
  - Such that $C = U\Sigma V^T$
    - If $C\ is\ (M\ by\ N)$
    - then $U\ is\ (M\ by\ M)$
    - and $\Sigma\ is\ (M\ by\ N)$
    - and
    - also Sigma is almost a diagonal matrix

Matrix Decomposition

- Singular Value Decomposition
  - Splits a matrix into three matrices $U \quad \Sigma \quad V^T$
  - Such that $C \quad = \quad U\Sigma V^T$
    - If $C \ is \ (M \ by \ N)$
    - then $U \ is \ (M \ by \ M)$
    - and $\Sigma \ is \ (M \ by \ N)$
    - and $V^T \ is \ (N \ by \ N)$
    - also Sigma is almost a diagonal matrix

Matrix Decomposition



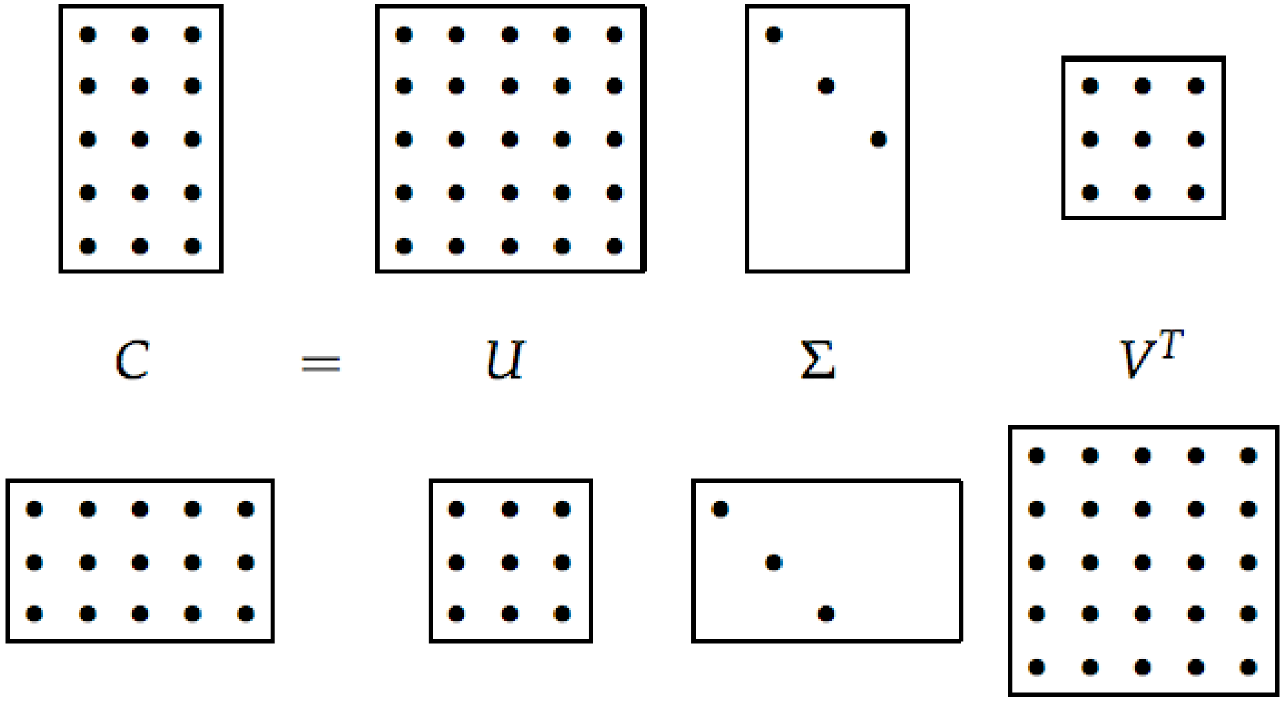
$$C = U \Sigma V^T$$

Matrix Decomposition

- Singular Value Decomposition
    - Is a technique that splits a matrix into three components with these properties.
    - They also have some other properties which are relevant to latent semantic indexing

Matrix Decomposition

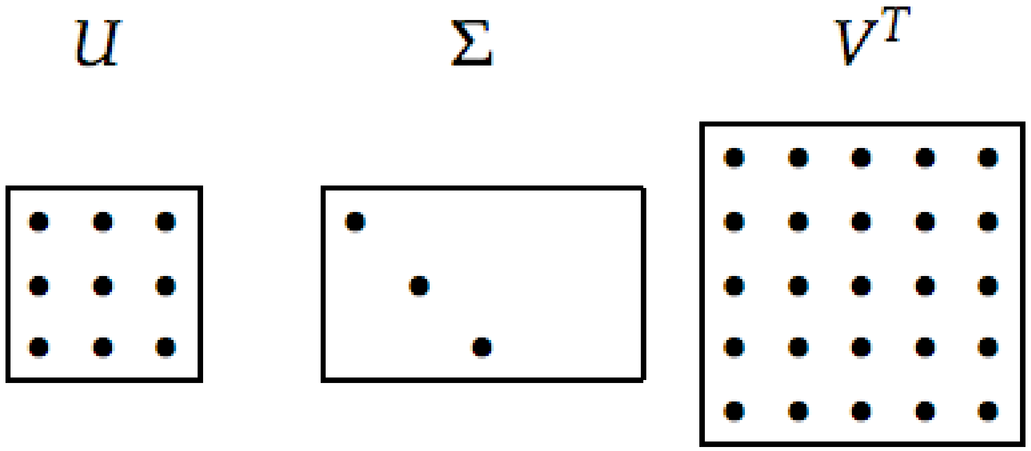
- Singular Value Decomposition
  - Is a technique that splits a matrix into three components with these properties.



Latent Semantic Indexing - Linear Algebra Refresher

Matrix Decomposition

- Singular Value Decomposition
  - SVD enables lossy compression of your term-document matrix
    - reduces the dimensionality or the rank
    - you can arbitrarily reduce the dimensionality by putting zeros in the bottom right of sigma
    - this is a mathematically optimal way of reducing dimensions



Matrix Decomposition

- Singular Value Decomposition
  - If the old dimensions were based on terms
    - after reducing the rank of the matrix the dimensionality is based on concepts or semantics
    - a concept is a linear combination of terms

$$SVD_{dimension_1} = a * td_{dim_1} + b * td_{dim_2} + c * td_{dim_3} + d * td_{dim_4}$$

$$SVD_{dimension_2} = a' * td_{dim_1} + b' * td_{dim_2} + c' * td_{dim_3} + d' * td_{dim_4}$$

$$SVD_{dimension_3} = a'' * td_{dim_1} + b'' * td_{dim_2} + c'' * td_{dim_3} + d'' * td_{dim_4}$$

Latent Semantic Indexing - Linear Algebra Refresher

Matrix Decomposition

- Singular Value Decomposition

$$SVD_{dimension_1} = a * td_{dim_1} + b * td_{dim_2} + c * td_{dim_3} + d * td_{dim_4}$$

$$SVD_{dimension_2} = a' * td_{dim_1} + b' * td_{dim_2} + c' * td_{dim_3} + d' * td_{dim_4}$$

$$SVD_{dimensions_3} = a'' * td_{dim_1} + b'' * td_{dim_2} + c'' * td_{dim_3} + d'' * td_{dim_4}$$

- 4 dimensions to 3 dimensions

$$\begin{vmatrix} a & b & c & d \\ a' & b' & c' & d' \\ a'' & b'' & c'' & d'' \end{vmatrix}$$

Matrix Decomposition

- Singular Value Decomposition

$$SVD_{dimension_1} = a * td_{dim_1} + b * td_{dim_2} + c * td_{dim_3} + d * td_{dim_4}$$

$$SVD_{dimension_2} = a' * td_{dim_1} + b' * td_{dim_2} + c' * td_{dim_3} + d' * td_{dim_4}$$

$$SVD_{dimensions_3} = a'' * td_{dim_1} + b'' * td_{dim_2} + c'' * td_{dim_3} + d'' * td_{dim_4}$$

- 4 dimensions to 3 dimensions

$$\begin{vmatrix} a & b & c & d \\ a' & b' & c' & d' \\ a'' & b'' & c'' & d'' \end{vmatrix} * \begin{vmatrix} td_{dim_1} \\ td_{dim_2} \\ td_{dim_3} \\ td_{dim_4} \end{vmatrix}$$

Matrix Decomposition

- Singular Value Decomposition

$$SVD_{dimension_1} = a * td_{dim_1} + b * td_{dim_2} + c * td_{dim_3} + d * td_{dim_4}$$

$$SVD_{dimension_2} = a' * td_{dim_1} + b' * td_{dim_2} + c' * td_{dim_3} + d' * td_{dim_4}$$

$$SVD_{dimension_3} = a'' * td_{dim_1} + b'' * td_{dim_2} + c'' * td_{dim_3} + d'' * td_{dim_4}$$

$$\begin{vmatrix} SVD_{dim_1} \\ SVD_{dim_2} \\ SVD_{dim_3} \end{vmatrix} = \begin{vmatrix} a & b & c & d \\ a' & b' & c' & d' \\ a'' & b'' & c'' & d'' \end{vmatrix} * \begin{vmatrix} td_{dim_1} \\ td_{dim_2} \\ td_{dim_3} \\ td_{dim_4} \end{vmatrix}$$

Latent Semantic Indexing - Linear Algebra Refresher

Matrix Decomposition

- Singular Value Decomposition

$$SVD_{dimension_1} = a * td_{dim_1} + b * td_{dim_2} + c * td_{dim_3} + d * td_{dim_4}$$

$$SVD_{dimension_2} = a' * td_{dim_1} + b' * td_{dim_2} + c' * td_{dim_3} + d' * td_{dim_4}$$

$$SVD_{dimension_3} = a'' * td_{dim_1} + b'' * td_{dim_2} + c'' * td_{dim_3} + d'' * td_{dim_4}$$

$$\begin{vmatrix} SVD_{dim_1} \\ SVD_{dim_2} \\ SVD_{dim_3} \end{vmatrix} = \begin{vmatrix} a & b & c & d \\ a' & b' & c' & d' \\ a'' & b'' & c'' & d'' \end{vmatrix} * \begin{vmatrix} td_{dim_1} \\ td_{dim_2} \\ td_{dim_3} \\ td_{dim_4} \end{vmatrix}$$

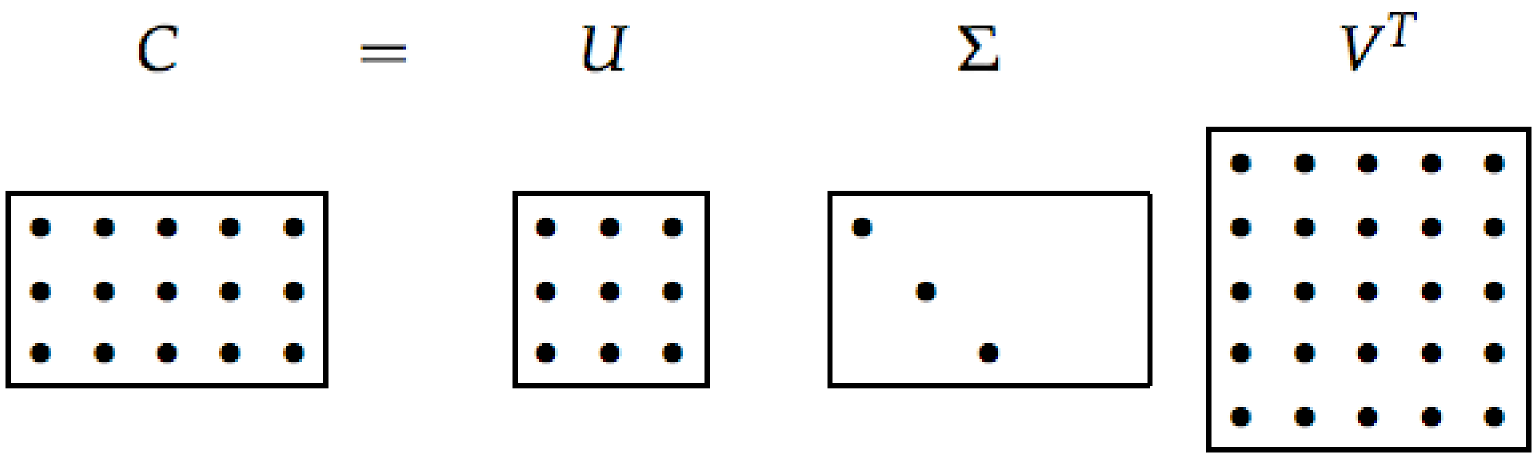
$$SVD_{ConceptSpace} = M * query_{TermSpace}$$

Matrix Decomposition

- Singular Value Decomposition

$$\begin{vmatrix} SVD_{dim_1} \\ SVD_{dim_2} \\ SVD_{dim_3} \end{vmatrix} = \begin{vmatrix} a & b & c & d \\ a' & b' & c' & d' \\ a'' & b'' & c'' & d'' \end{vmatrix} * \begin{vmatrix} td_{dim_1} \\ td_{dim_2} \\ td_{dim_3} \\ td_{dim_4} \end{vmatrix}$$

$$SVD_{ConceptSpace} = M * query_{TermSpace}$$



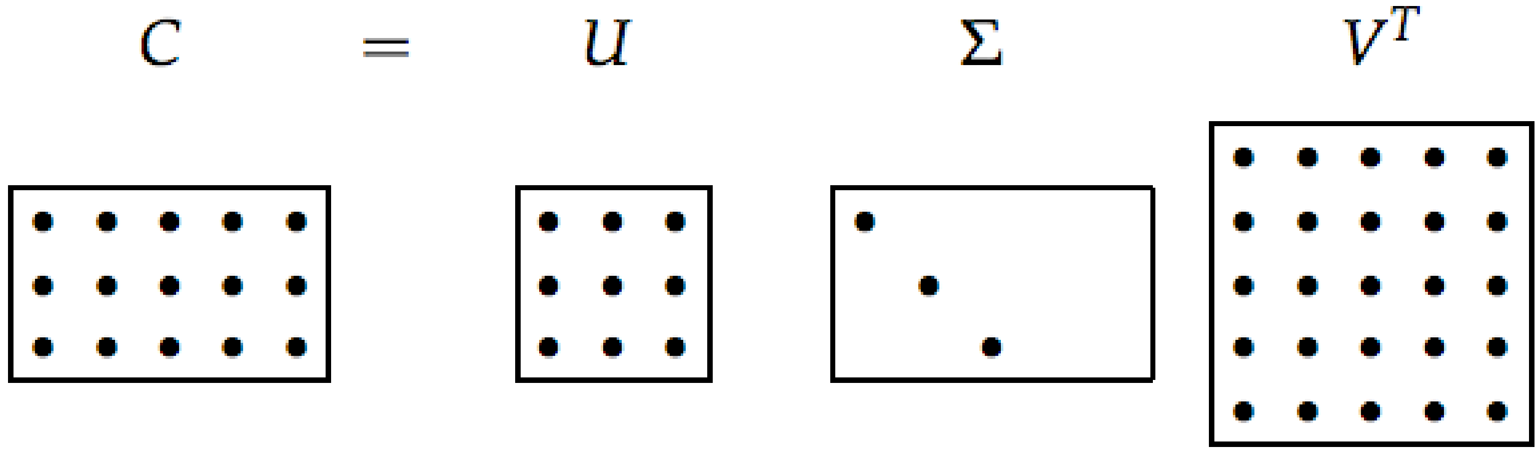
Matrix Decomposition

- Singular Value Decomposition

$$\begin{vmatrix} SVD_{dim_1} \\ SVD_{dim_2} \\ SVD_{dim_3} \end{vmatrix} = \begin{vmatrix} a & b & c & d \\ a' & b' & c' & d' \\ a'' & b'' & c'' & d'' \end{vmatrix} * \begin{vmatrix} td_{dim_1} \\ td_{dim_2} \\ td_{dim_3} \\ td_{dim_4} \end{vmatrix}$$

$$SVD_{ConceptSpace} = M * query_{TermSpace}$$

$$M = \Sigma_k^{-1} U_k^T$$



Matrix Decomposition

- Singular Value Decomposition

$$\begin{vmatrix} SVD_{dim_1} \\ SVD_{dim_2} \\ SVD_{dim_3} \end{vmatrix} = \begin{vmatrix} a & b & c & d \\ a' & b' & c' & d' \\ a'' & b'' & c'' & d'' \end{vmatrix} * \begin{vmatrix} td_{dim_1} \\ td_{dim_2} \\ td_{dim_3} \\ td_{dim_4} \end{vmatrix}$$

$$SVD_{ConceptSpace} = M * query_{TermSpace}$$

$$M = \Sigma_k^{-1} U_k^T$$

$$query_{ConceptSpace} = \Sigma_k^{-1} U_k^T query_{TermSpace}$$

Latent Semantic Indexing - Linear Algebra Refresher

Matrix Decomposition

- Singular Value Decomposition
  - SVD is an algorithm that gives us $\Sigma \; U \; V^T$
  - With these quantities we can reduce dimensionality
  - With reduced dimensionality
    - synonyms are mapped onto the same location
      - “bat” “chiroptera”
    - polysemies are mapped onto different locations
      - “bat” (baseball) vs. “bat” (small furry mammal)

Latent Semantic Indexing - Linear Algebra Refresher

- Computing SVD takes a significant amount of CPU
- It is possible to add documents to a corpus without recalculating SVD
  - The result becomes an approximation
  - To get mathematical guarantees the whole SVD needs to be computed from scratch
- LSI doesn’t support negation queries
- LSI doesn’t support boolean queries

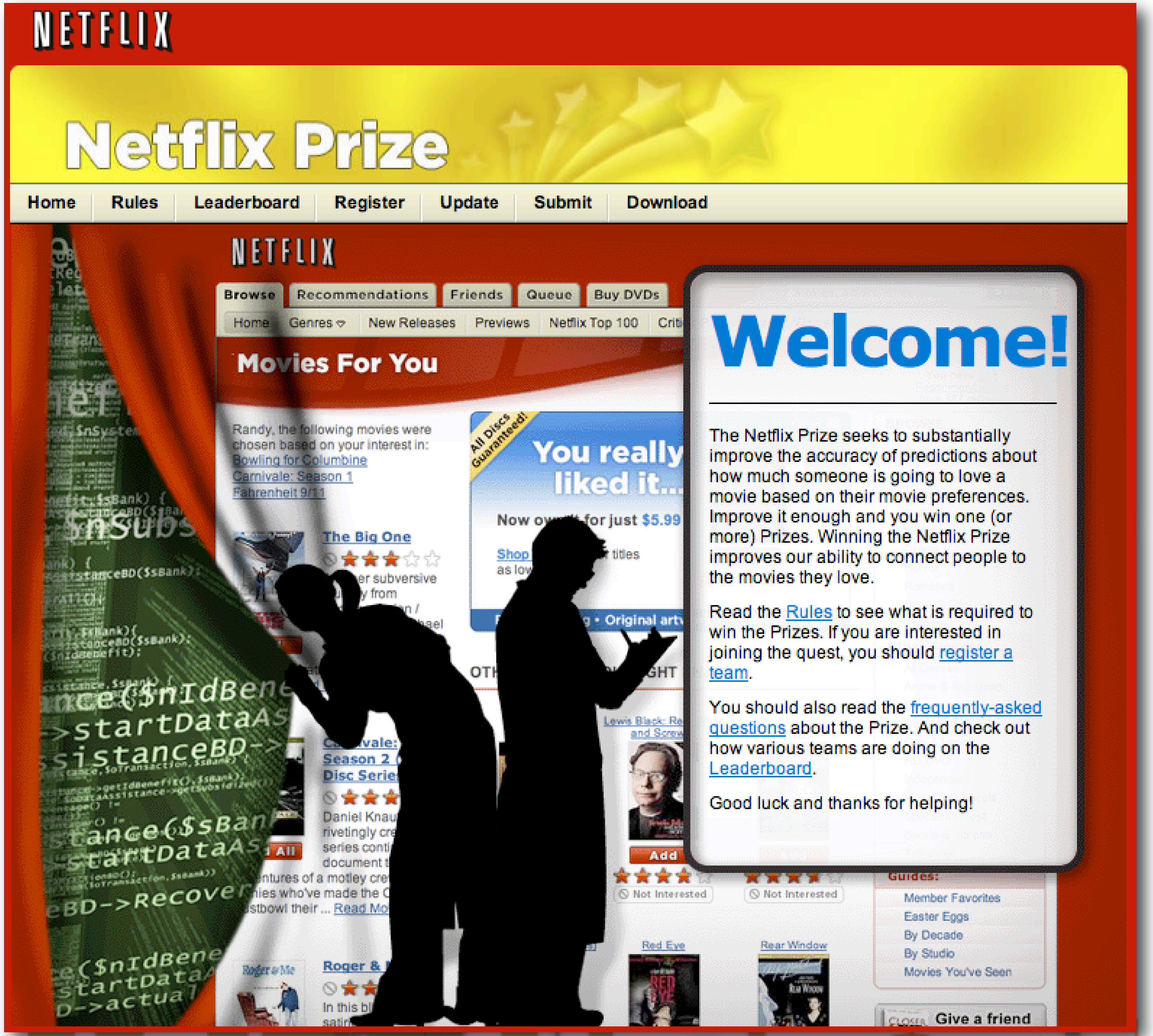
Latent Semantic Indexing - Linear Algebra Refresher

Matrix Decomposition

- “I am not crazy”
    - Netflix

Matrix Decomposition

- "I am not crazy"
  - Netflix



Welcome!

The Netflix Prize seeks to substantially improve the accuracy of predictions about how much someone is going to love a movie based on their movie preferences. Improve it enough and you win one (or more) Prizes. Winning the Netflix Prize improves our ability to connect people to the movies they love.

Read the Rules to see what is required to win the Prizes. If you are interested in joining the quest, you should register a team.

You should also read the frequently-asked questions about the Prize. And check out how various teams are doing on the Leaderboard.

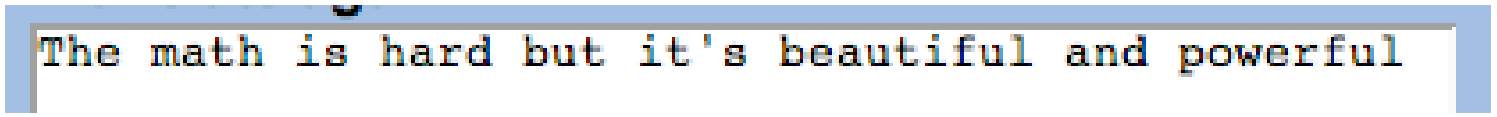
Good luck and thanks for helping!

Matrix Decomposition

- “I am not crazy”
  - Netflix
  - Machine translations
    - Just like “bat” and “chiroptera” map the same
    - “bat” and “murciélago” can map to the same thing

Matrix Decomposition

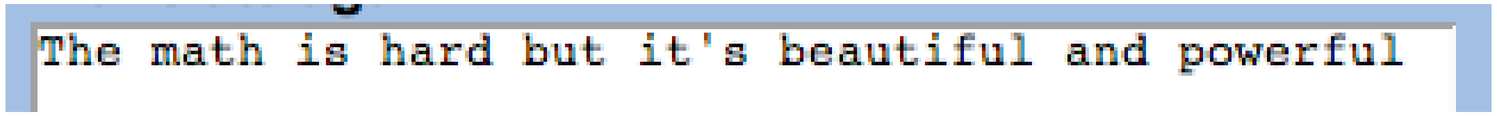
- “I am not crazy”
  - Netflix
  - Machine translations
    - Just like “bat” and “chiroptera” map the same
    - “bat” and “murciélago” can map to the same thing



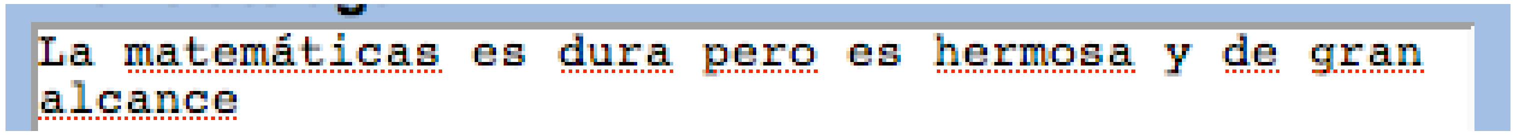
The math is hard but it's beautiful and powerful

Matrix Decomposition

- "I am not crazy"
  - Netflix
  - Machine translations
    - Just like "bat" and "chiroptera" map the same
    - "bat" and "murciélago" can map to the same thing



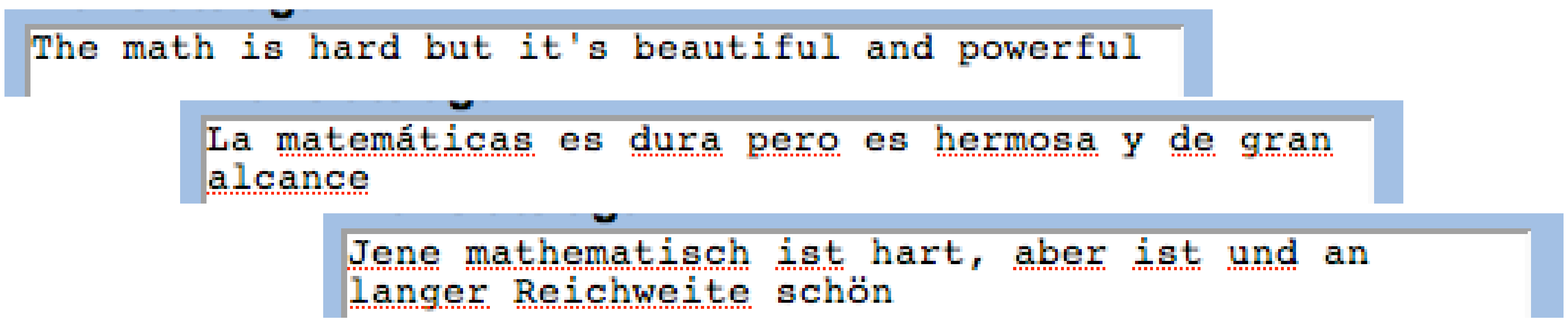
The math is hard but it's beautiful and powerful



La matemáticas es dura pero es hermosa y de gran alcance

Matrix Decomposition

- “I am not crazy”
  - Netflix
  - Machine translations
    - Just like “bat” and “chiroptera” map the same
    - “bat” and “murciélago” can map to the same thing



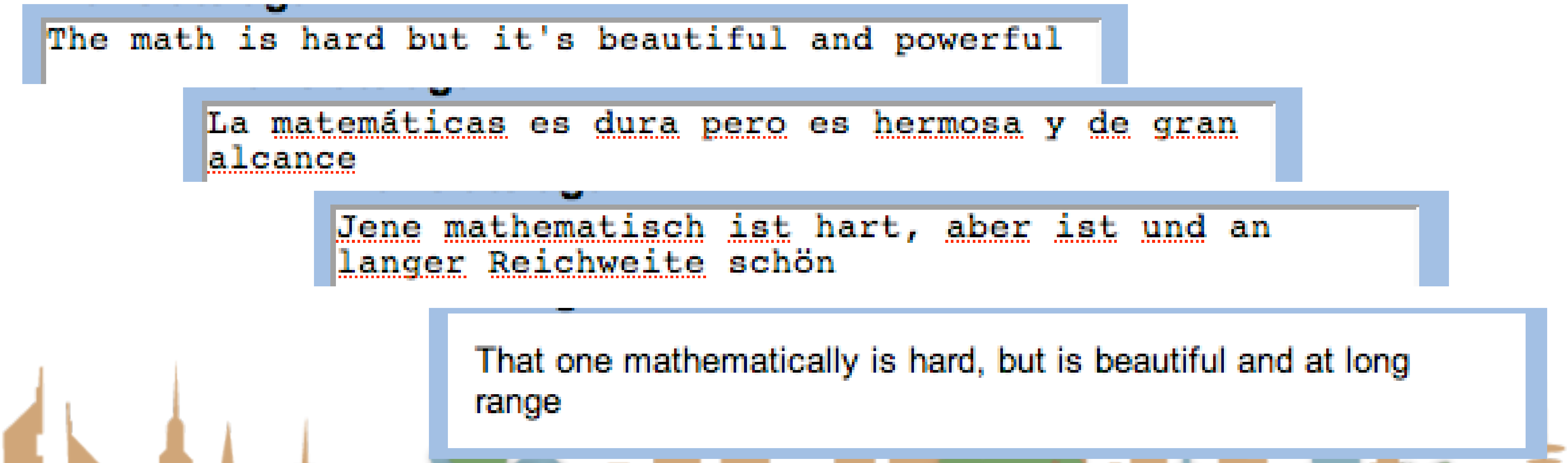
The math is hard but it's beautiful and powerful

La matemáticas es dura pero es hermosa y de gran alcance

Jene mathematisch ist hart, aber ist und an langer Reichweite schön

Matrix Decomposition

- “I am not crazy”
  - Netflix
  - Machine translations
    - Just like “bat” and “chiroptera” map the same
    - “bat” and “murciélago” can map to the same thing



The math is hard but it's beautiful and powerful

La matemáticas es dura pero es hermosa y de gran alcance

Jene mathematisch ist hart, aber ist und an langer Reichweite schön

That one mathematically is hard, but is beautiful and at long range next...

LUCI