

# Distributional Semantic Models

## Part 4: Elements of matrix algebra

Stefan Evert<sup>1</sup>

with Alessandro Lenci<sup>2</sup>, Marco Baroni<sup>3</sup> and Gabriella Lapesa<sup>4</sup>

<sup>1</sup>Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

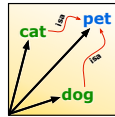
<sup>2</sup>University of Pisa, Italy

<sup>3</sup>University of Trento, Italy

<sup>4</sup>University of Stuttgart, Germany

<http://wordspace.collocations.de/doku.php/course:start>

Copyright © 2009–2019 Evert, Lenci, Baroni & Lapesa | Licensed under CC-by-sa version 3.0



## Outline

### Matrix algebra

Roll your own DSM

Matrix multiplication

Association scores & normalization

### Geometry

Metrics and norms

Angles and orthogonality

### Dimensionality reduction

Orthogonal projection

PCA & SVD

## Outline

### Matrix algebra

Roll your own DSM

Matrix multiplication

Association scores & normalization

### Geometry

Metrics and norms

Angles and orthogonality

### Dimensionality reduction

Orthogonal projection

PCA & SVD

## Matrices and vectors

- ▶  $k \times n$  matrix  $\mathbf{M} \in \mathbb{R}^{k \times n}$  is a rectangular array of real numbers

$$\mathbf{M} = \begin{bmatrix} m_{11} & \cdots & m_{1n} \\ \vdots & & \vdots \\ m_{k1} & \cdots & m_{kn} \end{bmatrix}$$

- ▶ Each row  $\mathbf{m}_i \in \mathbb{R}^n$  is an  $n$ -dimensional vector

$$\mathbf{m}_i = (m_{i1}, m_{i2}, \dots, m_{in})$$

- ▶ Similarly, each column is a  $k$ -dimensional vector  $\in \mathbb{R}^k$

```
> options(digits=3)
> M <- DSM_TermTerm$M
> M[2, ] # row vector  $\mathbf{m}_2$  for "dog"
> M[, 5] # column vector for "important"
```

## Matrices and vectors

- ▶ Vector  $\mathbf{x} \in \mathbb{R}^n$  as single-row or single-column matrix
  - ▶  $\mathbf{x} = \mathbf{x}^{TT} = n \times 1$  matrix ("vertical")
  - ▶  $\mathbf{x}^T = 1 \times n$  matrix ("horizontal")
  - ▶ **transposition** operator  $\cdot^T$  swaps rows & columns of matrix
- ▶ We need vectors  $\mathbf{r} \in \mathbb{R}^k$  and  $\mathbf{c} \in \mathbb{R}^n$  of marginal frequencies
- ▶ Notation for cell  $ij$  of co-occurrence matrix:
  - ▶  $m_{ij} = O \dots$  observed co-occurrence frequency
  - ▶  $r_i = R \dots$  row marginal (target)
  - ▶  $c_j = C \dots$  column marginal (feature)
  - ▶  $N \dots$  sample size

```
> r <- DSM_TermTerm$rows$f
> c <- DSM_TermTerm$cols$f
> N <- DSM_TermTerm$globals$N
> t(r)      # "horizontal" vector
> t(t(r))  # "vertical" vector
```

## Scalar operations

- ▶ **Scalar** operations perform the same transformation on each element of a vector or matrix, e.g.
  - ▶ add / subtract fixed shift  $\mu \in \mathbb{R}$
  - ▶ multiply / divide by fixed factor  $\sigma \in \mathbb{R}$
  - ▶ apply function ( $\log, \sqrt{\cdot}, \dots$ ) to each element
- ▶ Allows efficient processing of large sets of values
- ▶ Element-wise binary operators on matching vectors / matrices
  - ▶  $\mathbf{x} + \mathbf{y} =$  **vector addition**
  - ▶  $\mathbf{x} \odot \mathbf{y} =$  element-wise multiplication (**Hadamard product**)

```
> log(M + 1) # discounted log frequency weighting
> (M["cause", ] + M["effect", ]) / 2 # centroid vector
```

## The outer product

- ▶ Compute matrix  $\mathbf{E} \in \mathbb{R}^{k \times n}$  of expected frequencies

$$e_{ij} = \frac{r_i c_j}{N}$$

i.e.  $\mathbf{r}[i] * \mathbf{c}[j]$  for each cell  $ij$

- ▶ This is the **outer product** of  $\mathbf{r}$  and  $\mathbf{c}$

$$\begin{bmatrix} r_1 \\ \vdots \\ r_k \end{bmatrix} \cdot \begin{bmatrix} c_1 & c_2 & \dots & c_n \end{bmatrix} = \begin{bmatrix} r_1 c_1 & r_1 c_2 & \dots & r_1 c_n \\ \vdots & \vdots & & \vdots \\ r_k c_1 & r_k c_2 & \dots & r_k c_n \end{bmatrix}$$

- ▶ The **inner product** of  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$  is the sum  $x_1 y_1 + \dots + x_n y_n$

```
> outer(r, c) / N
```

## Outline

### Matrix algebra

Roll your own DSM

Matrix multiplication

Association scores & normalization

### Geometry

Metrics and norms

Angles and orthogonality

### Dimensionality reduction

Orthogonal projection

PCA & SVD

## Matrix multiplication

$$\begin{bmatrix} a_{ij} \end{bmatrix} = \begin{bmatrix} b_{i1} & \cdots & b_{in} \end{bmatrix} \cdot \begin{bmatrix} c_{1j} \\ \vdots \\ c_{nj} \end{bmatrix}$$

$$\mathbf{A} = \mathbf{B} \cdot \mathbf{C}$$

$(k \times m) \quad = \quad (k \times n) \quad \cdot \quad (n \times m)$

- ▶  $\mathbf{B}$  and  $\mathbf{C}$  must be **conformable** (in dimension  $n$ )
- ▶ Element  $a_{ij}$  is the inner product of the  $i$ -th row of  $\mathbf{B}$  and the  $j$ -th column of  $\mathbf{C}$

$$a_{ij} = b_{i1}c_{1j} + \dots + b_{in}c_{nj} = \sum_{t=1}^n b_{it}c_{tj}$$

## Some properties of matrix multiplication

Associativity:  $\mathbf{A}(\mathbf{BC}) = (\mathbf{AB})\mathbf{C} =: \mathbf{ABC}$

Distributivity:  $\mathbf{A}(\mathbf{B} + \mathbf{B}') = \mathbf{AB} + \mathbf{AB}'$   
 $(\mathbf{A} + \mathbf{A}')\mathbf{B} = \mathbf{AB} + \mathbf{A'B}$

Scalar multiplication:  $(\lambda\mathbf{A})\mathbf{B} = \mathbf{A}(\lambda\mathbf{B}) = \lambda(\mathbf{AB}) =: \lambda\mathbf{AB}$

- ▶ Not commutative in general:  $\mathbf{AB} \neq \mathbf{BA}$
- ▶ The  $k$ -dimensional square-diagonal **identity matrix**

$$\mathbf{I}_k := \begin{bmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{bmatrix} \quad \text{with} \quad \mathbf{I}_k \cdot \mathbf{A} = \mathbf{A} \cdot \mathbf{I}_n = \mathbf{A}$$

is the **neutral element** of matrix multiplication

## Transposition and multiplication

- ▶ The **transpose**  $\mathbf{A}^T$  of a matrix  $\mathbf{A}$  swaps rows and columns:

$$\begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \end{bmatrix}^T = \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{bmatrix}$$

- ▶ Properties of the transpose:
  - ▶  $(\mathbf{A} + \mathbf{B})^T = \mathbf{A}^T + \mathbf{B}^T$
  - ▶  $(\lambda\mathbf{A})^T = \lambda(\mathbf{A}^T) =: \lambda\mathbf{A}^T$
  - ▶  $(\mathbf{A} \cdot \mathbf{B})^T = \mathbf{B}^T \cdot \mathbf{A}^T$  [note the different order of  $\mathbf{A}$  and  $\mathbf{B}$ !]
  - ▶  $\mathbf{I}^T = \mathbf{I}$
- ▶  $\mathbf{A}$  is called **symmetric** iff  $\mathbf{A}^T = \mathbf{A}$ 
  - ▶ symmetric matrices have many special properties that will become important later (e.g. eigenvalues)

## The outer product as matrix multiplication

- ▶ The outer product is a special case of matrix multiplication

$$\mathbf{E} = \frac{1}{N}(\mathbf{r} \cdot \mathbf{c}^T)$$

- ▶ The other special case is the **inner product**

$$\mathbf{x}^T \mathbf{y} = \sum_{i=1}^n x_i y_i$$

- ▶ NB:  $\mathbf{x} \cdot \mathbf{x}$  and  $\mathbf{x}^T \cdot \mathbf{x}^T$  are not conformable

# three ways to compute the matrix of expected frequencies

```
> E <- outer(r, c) / N
> E <- (r %*% t(c)) / N
> E <- tcrossprod(r, c) / N
> E
```

## Outline

### Matrix algebra

Roll your own DSM  
Matrix multiplication  
Association scores & normalization

### Geometry

Metrics and norms  
Angles and orthogonality

### Dimensionality reduction

Orthogonal projection  
PCA & SVD

## Normalizing vectors

- ▶ Compute Euclidean norm of vector  $\mathbf{x} \in \mathbb{R}^n$ :

$$\|\mathbf{x}\|_2 = \sqrt{x_1^2 + \dots + x_n^2}$$

- ▶ Normalized vector  $\|\mathbf{x}_0\|_2 = 1$  by scalar multiplication:

$$\mathbf{x}_0 = \frac{1}{\|\mathbf{x}\|_2} \mathbf{x}$$

```
> x <- S[2, ]
> b <- sqrt(sum(x ^ 2)) # Euclidean norm of x
> x0 <- x / b          # normalized vector
> sqrt(sum(x0 ^ 2))
```

## Computing association scores

- ▶ Association scores = element-wise combination of  $\mathbf{M}$  and  $\mathbf{E}$ , e.g. for (pointwise) Mutual Information

$$\mathbf{S} = \log_2(\mathbf{M} \oslash \mathbf{E})$$

- ▶  $\oslash$  = element-wise division similar to Hadamard product  $\odot$
- ▶ For sparse AMs such as PPMI, we need to compute  $\max\{s_{ij}, 0\}$  for each element of the scored matrix  $\mathbf{S}$

```
> log2(M / E)
> S <- pmax(log2(M / E), 0) # not max() !
> S
```

## Normalizing matrix rows

- ▶ Compute vector  $\mathbf{b} \in \mathbb{R}^k$  of norms of row vectors of  $\mathbf{S}$
- ▶ Can you find an elegant way to multiply each row of  $\mathbf{S}$  with the corresponding normalization factor  $b_i^{-1}$ ?
- ▶ Multiplication with **diagonal matrix**  $\mathbf{D}_b^{-1}$

$$\mathbf{S}_0 = \mathbf{D}_b^{-1} \cdot \mathbf{S}$$

$$\mathbf{S}_0 = \begin{bmatrix} b_1^{-1} & & \\ & \ddots & \\ & & b_k^{-1} \end{bmatrix} \cdot \begin{bmatrix} s_{11} & \cdots & s_{1n} \\ \vdots & & \vdots \\ s_{k1} & \cdots & s_{kn} \end{bmatrix}$$

- ☞ What about multiplication with diagonal matrix on the right?

## Normalizing matrix rows

- ▶ Compute vector  $\mathbf{b} \in \mathbb{R}^k$  of norms of row vectors of  $\mathbf{S}$
- ▶ Can you find an elegant way to multiply each row of  $\mathbf{S}$  with the corresponding normalization factor  $b_i^{-1}$ ?
- ▶ Multiplication with **diagonal matrix**  $\mathbf{D}_b^{-1}$

$$\mathbf{S}_0 = \mathbf{D}_b^{-1} \cdot \mathbf{S}$$

```
> b <- sqrt(rowSums(S^2))
> b <- rowNorms(S, method="euclidean") # more efficient

> S0 <- diag(1 / b) %*% S
> S0 <- scaleMargins(S, rows=(1 / b)) # much more efficient

> S0 <- normalize.rows(S, method="euclidean") # the easy way
```

## Outline

### Matrix algebra

- Roll your own DSM
- Matrix multiplication
- Association scores & normalization

### Geometry

- Metrics and norms
- Angles and orthogonality

### Dimensionality reduction

- Orthogonal projection
- PCA & SVD

## Metric: a measure of distance

- ▶ A **metric** is a general measure of the distance  $d(\mathbf{u}, \mathbf{v})$  between points  $\mathbf{u}$  and  $\mathbf{v}$ , which satisfies the following axioms:
  - ▶  $d(\mathbf{u}, \mathbf{v}) = d(\mathbf{v}, \mathbf{u})$
  - ▶  $d(\mathbf{u}, \mathbf{v}) > 0$  for  $\mathbf{u} \neq \mathbf{v}$
  - ▶  $d(\mathbf{u}, \mathbf{u}) = 0$
  - ▶  $d(\mathbf{u}, \mathbf{w}) \leq d(\mathbf{u}, \mathbf{v}) + d(\mathbf{v}, \mathbf{w})$  (**triangle inequality**)
- ▶ Metrics form a very broad class of distance measures, some of which do not fit in well with our geometric intuitions
- ▶ Useful: family of **Minkowski**  $p$ -metrics

$$d_p(\mathbf{u}, \mathbf{v}) := (|u_1 - v_1|^p + \dots + |u_n - v_n|^p)^{1/p} \quad p \geq 1$$

$$d_p(\mathbf{u}, \mathbf{v}) := |u_1 - v_1|^p + \dots + |u_n - v_n|^p \quad 0 \leq p < 1$$

## Norm: a measure of length

- ▶ A general **norm**  $\|\mathbf{u}\|$  for the length of a vector  $\mathbf{u}$  must satisfy the following axioms:
  - ▶  $\|\mathbf{u}\| > 0$  for  $\mathbf{u} \neq \mathbf{0}$
  - ▶  $\|\lambda \mathbf{u}\| = |\lambda| \cdot \|\mathbf{u}\|$  (**homogeneity**)
  - ▶  $\|\mathbf{u} + \mathbf{v}\| \leq \|\mathbf{u}\| + \|\mathbf{v}\|$  (**triangle inequality**)
- ▶ Every norm **induces** a metric

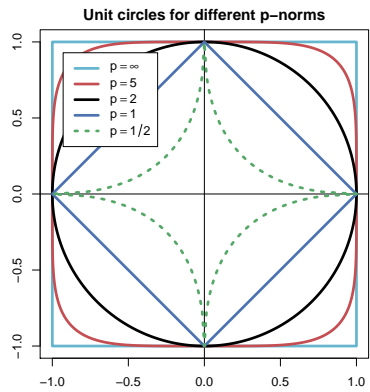
$$d(\mathbf{u}, \mathbf{v}) := \|\mathbf{u} - \mathbf{v}\|$$

with two desirable properties

- ▶ **translation-invariant**:  $d(\mathbf{u} + \mathbf{x}, \mathbf{v} + \mathbf{x}) = d(\mathbf{u}, \mathbf{v})$
- ▶ **scale-invariant**:  $d(\lambda \mathbf{u}, \lambda \mathbf{v}) = |\lambda| \cdot d(\mathbf{u}, \mathbf{v})$
- ▶  $d_p(\mathbf{u}, \mathbf{v})$  is induced by the **Minkowski norm** for  $p \geq 1$ :

$$\|\mathbf{u}\|_p := (|u_1|^p + \dots + |u_n|^p)^{1/p}$$

## Norm: a measure of length



- ▶ Visualisation of norms in  $\mathbb{R}^2$  by plotting **unit circle**, i.e. points  $\mathbf{u}$  with  $\|\mathbf{u}\| = 1$
- ▶ Here:  $p$ -norms  $\|\cdot\|_p$  for different values of  $p$
- ▶ Triangle inequality  $\iff$  unit circle is **convex**
- ▶ This shows that  $p$ -norms with  $p < 1$  would violate the triangle inequality

☞ Consequence for DSM:  $p \ll 2$  sensitive to small differences in many coordinates,  $p \gg 2$  to larger differences in few coord.

## Outline

### Matrix algebra

- Roll your own DSM
- Matrix multiplication
- Association scores & normalization

### Geometry

- Metrics and norms
- Angles and orthogonality

### Dimensionality reduction

- Orthogonal projection
- PCA & SVD

## Euclidean norm & inner product

- ▶ The Euclidean norm  $\|\mathbf{u}\|_2 = \sqrt{\mathbf{u}^T \mathbf{u}}$  is special because it can be derived from the **inner product**:

$$\mathbf{x}^T \mathbf{y} = x_1 y_1 + \dots + x_n y_n$$

- ▶ The inner product is a **positive definite** and **symmetric bilinear form** with an important geometric interpretation:

$$\cos \phi = \frac{\mathbf{u}^T \mathbf{v}}{\|\mathbf{u}\|_2 \cdot \|\mathbf{v}\|_2}$$

for the **angle**  $\phi$  between vectors  $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$

- ▶ the value  $\cos \phi$  is known as the **cosine similarity** measure
- ▶ In particular,  $\mathbf{u}$  and  $\mathbf{v}$  are **orthogonal** iff  $\mathbf{u}^T \mathbf{v} = 0$

## Cosine similarity in R

- ▶ Cosine similarities can be computed very efficiently if vectors are pre-normalized, so that  $\|\mathbf{u}\|_2 = \|\mathbf{v}\|_2 = 1$
- ☞ just need all inner products  $\mathbf{m}_i^T \mathbf{m}_j$  between row vectors of  $\mathbf{M}$

$$\mathbf{M} \cdot \mathbf{M}^T = \begin{bmatrix} \dots & \mathbf{m}_1 & \dots \\ \dots & \mathbf{m}_2 & \dots \\ \dots & \mathbf{m}_k & \dots \end{bmatrix} \cdot \begin{bmatrix} \vdots & \vdots & \vdots \\ \mathbf{m}_1 & \mathbf{m}_2 & \mathbf{m}_k \\ \vdots & \vdots & \vdots \end{bmatrix}$$

$$\implies (\mathbf{M} \cdot \mathbf{M}^T)_{ij} = \mathbf{m}_i^T \mathbf{m}_j$$

```
# cosine similarities for row-normalized matrix:
> sim <- tcrossprod(S0)
> angles <- acos(pmin(sim, 1)) * (180 / pi)
```

## Euclidean distance or cosine similarity?

- ▶ Proof that Euclidean distance and cosine similarity are equivalent looks much simpler in matrix algebra
- ▶ Assuming that  $\|\mathbf{u}\|_2 = \|\mathbf{v}\|_2 = 1$ , we have:

$$\begin{aligned} d_2(\mathbf{u}, \mathbf{v}) &= \|\mathbf{u} - \mathbf{v}\|_2 = \sqrt{(\mathbf{u} - \mathbf{v})^T(\mathbf{u} - \mathbf{v})} \\ &= \sqrt{\mathbf{u}^T\mathbf{u} + \mathbf{v}^T\mathbf{v} - 2\mathbf{u}^T\mathbf{v}} \\ &= \sqrt{\|\mathbf{u}\|_2^2 + \|\mathbf{v}\|_2^2 - 2\mathbf{u}^T\mathbf{v}} \\ &= \sqrt{2 - 2\cos\phi} \end{aligned}$$

☞  $d_2(\mathbf{u}, \mathbf{v})$  is a monotonically increasing function of  $\phi$

## Outline

### Matrix algebra

- Roll your own DSM
- Matrix multiplication
- Association scores & normalization

### Geometry

- Metrics and norms
- Angles and orthogonality

### Dimensionality reduction

- Orthogonal projection
- PCA & SVD

## Linear subspace & basis

- ▶ A linear **subspace**  $B \subseteq \mathbb{R}^n$  of rank  $r \leq n$  is spanned by a set of  $r$  linearly independent basis vectors

$$B = \{\mathbf{b}_1, \dots, \mathbf{b}_r\}$$

- ▶ Every point  $\mathbf{u}$  in the subspace is a unique linear combination of the basis vectors

$$\mathbf{u} = x_1\mathbf{b}_1 + \dots + x_r\mathbf{b}_r$$

with coordinate vector  $\mathbf{x} \in \mathbb{R}^r$

- ▶ Basis matrix  $\mathbf{V} \in \mathbb{R}^{n \times r}$  with column vectors  $\mathbf{b}_i$ :

$$\mathbf{u} = \mathbf{V}\mathbf{x}$$

## Linear subspace & basis

- ▶ Basis matrix  $\mathbf{V} \in \mathbb{R}^{n \times r}$  with column vectors  $\mathbf{b}_i$ :

$$\mathbf{u} = x_1\mathbf{b}_1 + \dots + x_r\mathbf{b}_r = \mathbf{V}\mathbf{x}$$

$$\begin{bmatrix} x_1 b_{11} + \dots + x_r b_{r1} \\ x_1 b_{12} + \dots + x_r b_{r2} \\ \vdots \\ x_1 b_{1n} + \dots + x_r b_{rn} \end{bmatrix} = \begin{bmatrix} b_{11} & \dots & b_{r1} \\ b_{12} & \dots & b_{r2} \\ \vdots & & \vdots \\ b_{1n} & \dots & b_{rn} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ \vdots \\ x_r \end{bmatrix}$$

$$\begin{matrix} \mathbf{u} \\ (n \times 1) \end{matrix} = \begin{matrix} \mathbf{V} \\ (n \times r) \end{matrix} \cdot \begin{matrix} \mathbf{x} \\ (r \times 1) \end{matrix}$$

## Orthonormal basis

- Particularly convenient with orthonormal basis:

$$\begin{aligned} \|\mathbf{b}_i\|_2 &= 1 \\ \mathbf{b}_i^T \mathbf{b}_j &= 0 \quad \text{for } i \neq j \end{aligned}$$

- Corresponding basis matrix  $\mathbf{V}$  is (column)-**orthogonal**

$$\mathbf{V}^T \mathbf{V} = \mathbf{I}_r$$

and defines a **Cartesian coordinate system** in the subspace

## The mathematics of projections

- 1-d subspace spanned by basis vector  $\|\mathbf{b}\|_2 = 1$
- For any point  $\mathbf{u}$ , we have

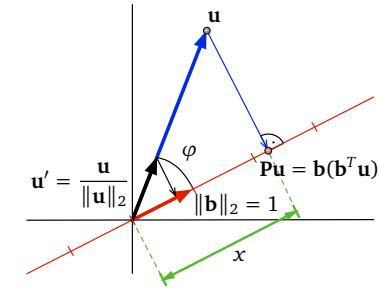
$$\cos \varphi = \frac{\mathbf{b}^T \mathbf{u}}{\|\mathbf{b}\|_2 \cdot \|\mathbf{u}\|_2} = \frac{\mathbf{b}^T \mathbf{u}}{\|\mathbf{u}\|_2}$$

- Trigonometry: coordinate of point on the line is  $x = \|\mathbf{u}\|_2 \cdot \cos \varphi = \mathbf{b}^T \mathbf{u}$

- The projected point in original space is then given by

$$\mathbf{b} \cdot x = \mathbf{b}(\mathbf{b}^T \mathbf{u}) = (\mathbf{b}\mathbf{b}^T) \mathbf{u} = \mathbf{P}\mathbf{u}$$

where  $\mathbf{P}$  is a **projection matrix** of rank 1



## The mathematics of projections

- For an orthogonal basis matrix  $\mathbf{V}$  with columns  $\mathbf{b}_1, \dots, \mathbf{b}_r$ , the projection into the rank- $r$  subspace  $B$  is given by

$$\mathbf{P}\mathbf{u} = \left( \sum_{i=1}^r \mathbf{b}_i \mathbf{b}_i^T \right) \mathbf{u} = \mathbf{V}\mathbf{V}^T \mathbf{u}$$

and its subspace coordinates are  $\mathbf{x} = \mathbf{V}^T \mathbf{u}$

- Projection can be seen as decomposition into the projected vector and its orthogonal complement

$$\mathbf{u} = \mathbf{P}\mathbf{u} + (\mathbf{u} - \mathbf{P}\mathbf{u}) = \mathbf{P}\mathbf{u} + (\mathbf{I} - \mathbf{P})\mathbf{u} = \mathbf{P}\mathbf{u} + \mathbf{Q}\mathbf{u}$$

- Because of orthogonality, this also applies to the squared Euclidean norm (according to the Pythagorean theorem)

$$\|\mathbf{u}\|^2 = \|\mathbf{P}\mathbf{u}\|^2 + \|\mathbf{Q}\mathbf{u}\|^2$$

## Aside: the matrix cross-product

- We already know that the (transpose) cross-product  $\mathbf{M}\mathbf{M}^T$  computes all inner products between the row vectors of  $\mathbf{M}$
- But  $\mathbf{V}\mathbf{V}^T$  it can also be understood as a **superposition** of the **outer products** of the columns of  $\mathbf{V}$  with themselves

$$\mathbf{V}\mathbf{V}^T = \begin{bmatrix} b_{11} \\ \vdots \\ b_{1n} \end{bmatrix} \cdot [b_{11} \cdots b_{1n}] + \dots + \begin{bmatrix} b_{r1} \\ \vdots \\ b_{rn} \end{bmatrix} \cdot [b_{r1} \cdots b_{rn}]$$



## Projections in R

```
# column basis vector for "animal" subspace
> b <- t(t(c(1, 1, 1, 1, .5, 0, 0)))
> b <- normalize.cols(b) # basis vectors must be normalized

> (x <- M %*% b) # projection of data points into subspace coordinates
> x %*% t(b) # projected points in original space
> tcrossprod(x, b) # outer() only works for plain vectors

> P <- b %*% t(b) # projection operator
> P - t(P) # note that P is symmetric
> M %*% P # projected points in original space
```

## Optimal projections and subspaces

- ▶ Optimal subspace maximises  $R^2$  across a data set  $\mathbf{M}$ , which is now specified in terms of row vectors  $\mathbf{m}_i^T$ :

$$\begin{aligned} \mathbf{x}_i^T &= \mathbf{m}_i^T \mathbf{V} & \mathbf{m}_i^T \mathbf{P} &= \mathbf{m}_i^T \mathbf{V} \mathbf{V}^T \\ \mathbf{X} &= \mathbf{M} \mathbf{V} & \mathbf{M} \mathbf{P} &= \mathbf{M} \mathbf{V} \mathbf{V}^T \end{aligned}$$

- ▶ We will now show that the overall projection quality is

$$R^2 = \frac{\sum_{i=1}^k \|\mathbf{m}_i^T \mathbf{P}\|^2}{\sum_{i=1}^k \|\mathbf{m}_i^T\|^2} = \frac{\|\mathbf{M} \mathbf{P}\|_F^2}{\|\mathbf{M}\|_F^2}$$

with the (squared) **Frobenius norm**

$$\|\mathbf{M}\|_F^2 = \sum_{ij} (m_{ij})^2 = \sum_{i=1}^k \|\mathbf{m}_i\|^2$$

## Optimal projections and subspaces

- ▶ Orthogonal decomposition of squared distances btw vectors

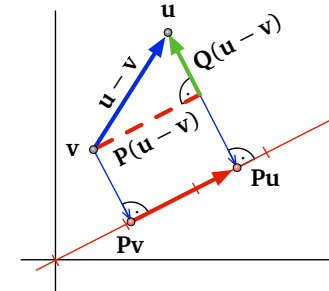
$$\|\mathbf{u} - \mathbf{v}\|^2 = \|\mathbf{P}\mathbf{u} - \mathbf{P}\mathbf{v}\|^2 + \|\mathbf{Q}\mathbf{u} - \mathbf{Q}\mathbf{v}\|^2$$

- ▶ Define projection **loss** as difference btw squared distances

$$\begin{aligned} & \left| \|\mathbf{P}(\mathbf{u} - \mathbf{v})\|^2 - \|\mathbf{u} - \mathbf{v}\|^2 \right| \\ &= \|\mathbf{u} - \mathbf{v}\|^2 - \|\mathbf{P}(\mathbf{u} - \mathbf{v})\|^2 \\ &= \|\mathbf{Q}(\mathbf{u} - \mathbf{v})\|^2 \end{aligned}$$

- ▶ Projection quality measure:

$$R^2 = \frac{\|\mathbf{P}(\mathbf{u} - \mathbf{v})\|^2}{\|\mathbf{u} - \mathbf{v}\|^2}$$



## Optimal projections and subspaces

- ▶ For a **centered** data set with  $\sum_i \mathbf{m}_i = \mathbf{0}$ , the Frobenius norm corresponds to the average (squared) distance between points

$$\begin{aligned} \sum_{i,j=1}^k \|\mathbf{m}_i - \mathbf{m}_j\|^2 &= \sum_{i,j=1}^k (\mathbf{m}_i - \mathbf{m}_j)^T (\mathbf{m}_i - \mathbf{m}_j) \\ &= \sum_{i,j=1}^k (\|\mathbf{m}_i\|^2 + \|\mathbf{m}_j\|^2 - 2\mathbf{m}_i^T \mathbf{m}_j) \\ &= \sum_{j=1}^k \|\mathbf{M}\|_F^2 + \sum_{i=1}^k \|\mathbf{M}\|_F^2 - 2 \sum_{i=1}^k \mathbf{m}_i^T \underbrace{\left( \sum_{j=1}^k \mathbf{m}_j \right)}_0 \\ &= 2k \cdot \|\mathbf{M}\|_F^2 \end{aligned}$$

- ▶ Similarly for the overall projection loss and quality  $R^2$ :

$$R^2 = \frac{\sum_{i,j=1}^k \|\mathbf{P}(\mathbf{m}_i - \mathbf{m}_j)\|^2}{\sum_{i,j=1}^k \|\mathbf{m}_i - \mathbf{m}_j\|^2} = \frac{2k \cdot \|\mathbf{M} \mathbf{P}\|_F^2}{2k \cdot \|\mathbf{M}\|_F^2}$$

## Outline

### Matrix algebra

- Roll your own DSM
- Matrix multiplication
- Association scores & normalization

### Geometry

- Metrics and norms
- Angles and orthogonality

### Dimensionality reduction

- Orthogonal projection
- PCA & SVD

## Singular value decomposition

- Fundamental result of matrix algebra: **singular value decomposition (SVD)** factorises any matrix  $\mathbf{M}$  into

$$\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

where  $\mathbf{U}$  and  $\mathbf{V}$  are orthogonal and  $\mathbf{\Sigma}$  is a diagonal matrix of **singular values**  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_m > 0$

$$\begin{bmatrix} & n \\ k & \mathbf{M} \end{bmatrix} = \begin{bmatrix} & m \\ k & \mathbf{U} \end{bmatrix} \cdot \begin{bmatrix} \sigma_1 & m \\ m & \ddots \\ & \mathbf{\Sigma} & \sigma_m \end{bmatrix} \cdot \begin{bmatrix} & n \\ m & \mathbf{V}^T \end{bmatrix}$$

## Singular value decomposition

- $m \leq \min\{k, n\}$  is the inherent dimensionality (**rank**) of  $\mathbf{M}$
- Columns  $\mathbf{a}_i$  of  $\mathbf{U}$  are called left singular vectors, columns  $\mathbf{b}_i$  of  $\mathbf{V}$  (= rows of  $\mathbf{V}^T$ ) are right singular vectors
- Recall the “outer product” view of matrix multiplication:

$$\mathbf{UV}^T = \sum_{i=1}^m \mathbf{a}_i \mathbf{b}_i^T$$

- Hence the SVD corresponds to a sum of rank-1 components

$$\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \sum_{i=1}^m \sigma_i \mathbf{a}_i \mathbf{b}_i^T$$

## Singular value decomposition

- Key property of SVD: the first  $r$  components give the best rank- $r$  approximation to  $\mathbf{M}$  with respect to the Frobenius norm, i.e. they minimize the loss

$$\|\mathbf{U}_r \mathbf{\Sigma}_r \mathbf{V}_r^T - \mathbf{M}\|_F^2 = \|\mathbf{M}_r - \mathbf{M}\|_F^2$$

### Truncated SVD

- $\mathbf{U}_r, \mathbf{V}_r$  = first  $r$  columns of  $\mathbf{U}, \mathbf{V}$
- $\mathbf{\Sigma}_r$  = diagonal matrix of first  $r$  singular values
- It can be shown that

$$\|\mathbf{M}\|_F^2 = \sum_{i=1}^m \sigma_i^2 \quad \text{and} \quad \|\mathbf{M}_r\|_F^2 = \sum_{i=1}^r \sigma_i^2$$

## SVD dimensionality reduction

- ▶ Columns of  $\mathbf{V}_r$  form an orthogonal basis of the optimal rank- $r$  subspace because

$$\mathbf{M}\mathbf{P} = \mathbf{M}\mathbf{V}_r\mathbf{V}_r^T = \mathbf{U}\underbrace{\mathbf{\Sigma}\mathbf{V}_r^T\mathbf{V}_r}_{=\mathbf{I}_r}\mathbf{V}_r^T = \mathbf{U}_r\mathbf{\Sigma}_r\mathbf{V}_r^T = \mathbf{M}_r$$

- ▶ **Dimensionality reduction** uses the subspace coordinates

$$\mathbf{M}\mathbf{V}_r = \mathbf{U}_r\mathbf{\Sigma}_r$$

- ▶ If  $\mathbf{M}$  is centered, this also gives the best possible preservation of pairwise distances → **principal component analysis (PCA)**
  - ☞ but centering is usually omitted in order to maintain sparseness, so SVD preserves vector lengths rather than distances

## Scaling SVD dimensions

- ▶ Singular values  $\sigma_i$  can be seen as weighting of the latent dimensions, which determines their contribution to

$$\|\mathbf{M}\mathbf{V}_r\|_F = \sigma_1^2 + \dots + \sigma_r^2$$

- ▶ Weighting adjusted by **power scaling** of the singular values:

$$\mathbf{U}_r\mathbf{\Sigma}_r^p = \begin{bmatrix} \vdots & & \vdots \\ \sigma_1^p \mathbf{a}_1 & \cdots & \sigma_r^p \mathbf{a}_r \\ \vdots & & \vdots \end{bmatrix}$$

- ▶  $p = 1$ : normal SVD projection
- ▶  $p = 0$ : dimension weights equalized
- ▶  $p = 2$ : more weight given to first latent dimensions
- ▶ Other weighting schemes possible (e.g. skip first dimensions)

## SVD projection in R

```
> fact <- svd(S0)      # SVD decomposition of S0
> round(fact$u, 3)     # left singular vectors (columns) = U
> round(fact$v, 3)     # right singular vectors (columns) = V
> round(fact$d, 3)     # singular values = diagonal of Σ
# note that S0 has effective rank 6 because σ7 ≈ 0
> barplot(fact$d ~ 2) # R² contributions

> r <- 2               # truncated rank-2 SVD
> (U.r <- fact$u[, 1:r])
> (Sigma.r <- diag(fact$d[1:r], nrow=r))
> (V.r <- fact$v[, 1:r])
```

## SVD projection in R

```
> (X.r <- S0 %*% V.r) # project into latent coordinates
> U.r %*% Sigma.r     # same result
> scaleMargins(U.r, cols=fact$d[1:r]) # the wordspace way

> rownames(X.r) <- rownames(S0)      # NB: keep row labels

> S0r <- U.r %*% Sigma.r %*% t(V.r)  # rank-2 matrix approx.
> round(S0r, 3)
# compare with S0: where are the differences?

> round(X.r %*% t(V.r), 3)           # same result
```

- ☞ see example code for comparison against PCA with centering