Vektoren und Matrizen

- Masterkurs: Vertiefung der Grundlagen der Computerlinguistik
- Referenten: Shuzhou Yuan, Erika Worm

Gliederung

- Vektoren
  - Begriffe
  - Beispiele
- Matrizen
  - Begriffe
  - Beispiele
- Anwendung in der Computerlinguistik

Gliederung

- Operationen und Algorithmen mit Vektoren und Matrizen
  - Matrix-Vektor Multiplikation
  - Matrix-Matrix Multiplikation
  - Vektornormierung
- Lineare Unabhängigkeit

Vektoren

- Skalar, reelle Zahl
- Vektor: Größe und Richtung

$\underset{v}{\rightarrow} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ zwei Komponenten = zwei Dimensionen

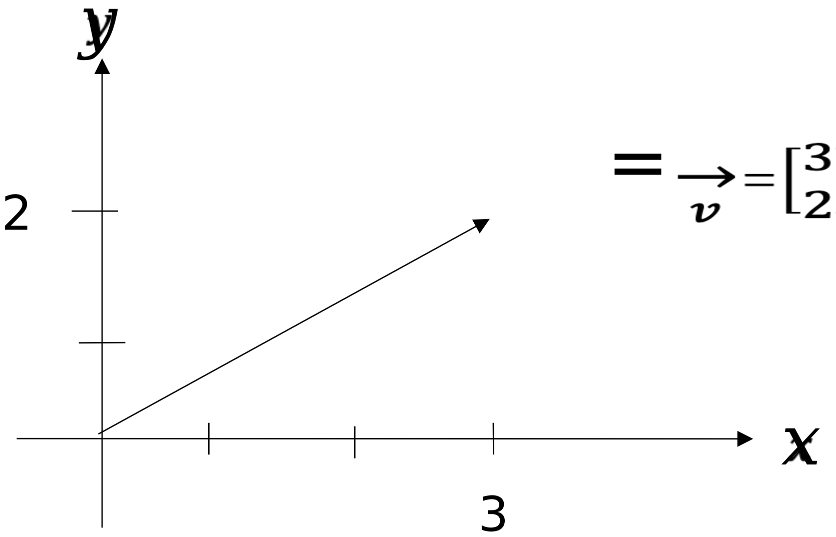
ein Vektor besteht aus beliebige reelle Zahlen/Komponenten

Zahl der Komponenten = Zahl der Dimensionen

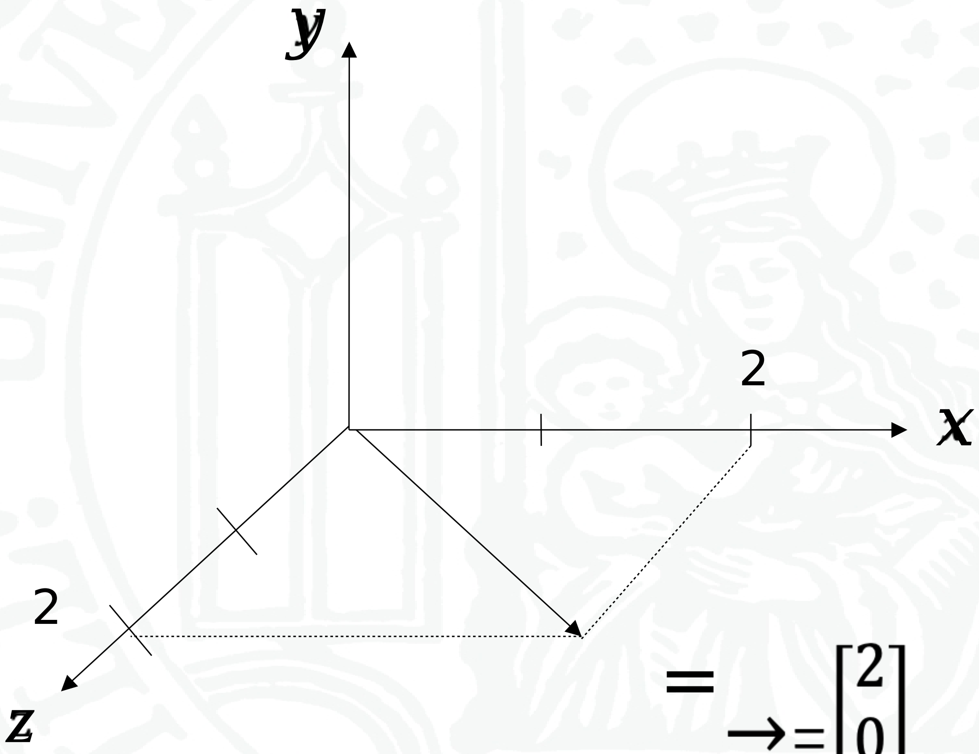
column vector: $\underset{v}{\rightarrow} = \begin{bmatrix} v_1 \\ . \\ . \\ v_n \end{bmatrix}$ row vector: $\underset{v}{\rightarrow} = [v_1 \dots v_n]$

Beispiel

- Die Punkte in x-y-Achse könnten als Vektoren darstellen



$= \vec{v} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$



$= \vec{v} = \begin{bmatrix} 2 \\ 0 \\ 2 \end{bmatrix}$

Beispiel

- Vektor ⟺ Liste von Zahlen

In NLP könnten wir die Vektoren benutzen, um die Texte zu verarbeiten.

Text 1: I like you.

$$\vec{v}_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

Text 2: He hates me.

$$\vec{v}_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Matrizen

- $A = \begin{bmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{m1} & \cdots & A_{mn} \end{bmatrix}$

A ist die Matrizen mit Größe m*n A_{ij} : die i-te Zeile/row, j-te Spalte/column

A ist die Matrizen mit Größe m*n A_{ij} : die i-te Zeile/row, j-te Spalte/column

- quadratische Matrix/ square matrix: m=n

- quadratische Matrix/ square matrix: m=n

- Diagonalmatrix: Als Diagonalmatrix bezeichnet man in der linearen Algebra eine quadratische Matrix, bei der alle Elemente außerhalb der Hauptdiagonale Null sind.

- Diagonalmatrix: Als Diagonalmatrix bezeichnet man in der linearen Algebra eine quadratische Matrix, bei der alle Elemente außerhalb der Hauptdiagonale Null sind.

$$\begin{bmatrix} 2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 2 \end{bmatrix}$$

Beispiel

10*10 Pixel schwarzweißes Foto, 1 für schwarz, 0für weiß
10*10 Pixel schwarzweißes Foto , 1 für schwarz, 0 für weiß

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

Anwendung in der Computerlinguistik

Term-document matrices are used in *information retrieval*. Consider the following selection of five documents. Key words, which we call *t erms*, are marked in boldface.

Document 1: The **Google matrix** P is a model of the **Internet**.

Document 2: P_{ij} is nonzero if there is a **link** from **Web page** j to i.

Document 3: The **Google matrix** is used to **rank** all **Web pages**.

Document 4: The **ranking** is done by solving a **matrix eigenvalue** pr oblem.

Document 5: **England** dropped out of the top 10 in the **FIFA ranking**.

If we count the frequency of terms in each document we get th
e following result:

Term	Doc 1	Doc 2	Doc 3	Doc 4	Doc 5
eigenvalue	0	0	0	1	0
England	0	0	0	0	1
FIFA	0	0	0	0	1
Google	1	0	1	0	0
Internet	1	0	0	0	0
link	0	1	0	0	0
matrix	1	0	1	1	0
page	0	1	1	0	0
rank	0	0	1	1	1
Web	0	1	1	0	0

Thus each document is represented by a vector, or a point, in $\mathbb{R}^{10}$, and we can organize all documents into a term-document matrix.

Thus each document is represented by a vector, or a point, in ^{10} , and we can organize all documents into a term-document ma trix.

$$A = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

ow assume that we want to find all documents that are relevant to th
uery“**ranking** of **Web pages**”.This is represented by a query vector,
onstructed in a way analogous to the term-document matrix:

$$A = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

$$q = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} \in \mathbb{R}^{10}.$$

Thus the query itself is considered as a document. The information retrieval task can now be formulated as a mathematical problem: find the columns of A that are close to the vector q. To solve this problem we must use some distance measure in ^{10}.

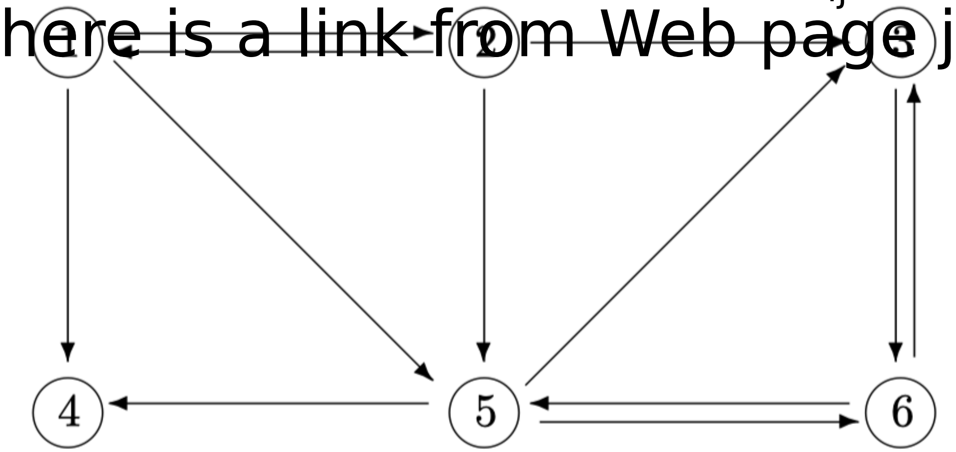
Thus the query itself is considered as a document. The information retrieval task can now be formulated as a mathematical problem: find the columns of A that are close to the vector q. To solve this problem we must use some distance measure in $\mathbb{R}^{10}$.

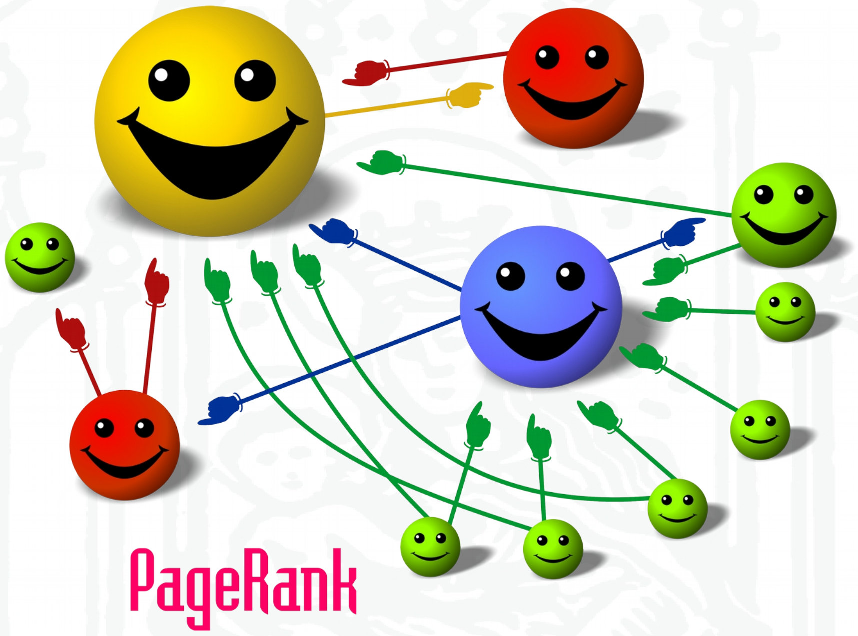
$$A = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

$$q = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} \in \mathbb{R}^{10}.$$

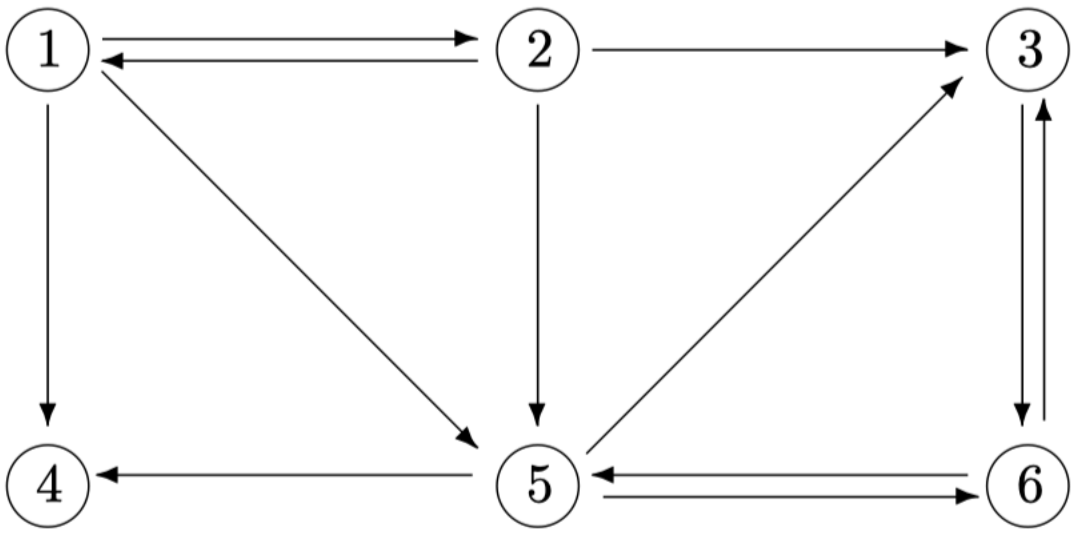
Anwendung in PageRanking

The core of the Google search engine is a matrix computation, probably the largest that is performed routinely. The Google matrix P is of the order billions, i.e., close to the total number of Web pages on the Internet. The matrix is constructed based on the link structure of the Web, and element P_{ij} is nonzero if there is a link from Web page j to i.





Anwendung in PageRank



$$P = \begin{pmatrix} 0 & \frac{1}{3} & 0 & 0 & 0 & 0 \\ \frac{1}{3} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} & 0 \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 1 & 0 & \frac{1}{3} & 0 \end{pmatrix}$$

Operationen und Algorithmen mit Vekt oren und Matrizen

- Schritte im Algorithmus/in der Operation nicht immer in selb er Reihenfolge
- Definition eines Algorithmus setzt nicht genaue Berechnungs struktur voraus

Matrix-Vektor Multiplikation

Reihenweise:
A als m x n Matrix

Reihenweise:
A als m x n Matrix

$$y = Ax, \qquad y_i = \sum_{j=1}^{n} a_{ij} x_j, \qquad i = 1, \ldots, m$$

Matrix-Vektor Multiplikation

Spaltenweise:

Spaltenweise:

$$y = Ax = \begin{pmatrix} a_{\cdot 1} & a_{\cdot 2} & \dots & a_{\cdot n} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix} = \sum_{j=1}^{n} a_{\cdot j}\, x_j$$

Anwendungsbeispiel:

Basisvektoren Koordinaten

Anwendungsbeispiel:

$a_{\cdot j}$ → Basisvektoren x_j → Koordinaten

Matrix-Matrix Multiplikation

- Geg. zwei Matrizen:
- Definition:

- Geg. zwei Matrizen: $A \in \mathbb{R}^{mxk},\ B \in \mathbb{R}^{kxn}$
- Definition:

$$\mathbb{R}^{mxn} \ni C = AB$$

$$c_{ij} = \sum_{s=1}^{k} a_{is} b_{sj}, i = 1, \dots, m, \qquad j = 1, \dots, n$$

Berechnung eines Skalarprodukts für unabhängig von Reihenfol ge

$\rightarrow$ Berechnung eines Skalarprodukts für c_{ij} unabhängig von Reihenfolge

Matrix-Matrix Multiplikation

Analog zu V x M Multiplikation:

Analog zu V x M Multiplikation:

$$C = AB = \begin{pmatrix} a_{\cdot 1} & a_{\cdot 2} & \dots & a_{\cdot k} \end{pmatrix} \begin{pmatrix} b_{1\cdot} \\ b_{2\cdot} \\ \dots \\ b_{k\cdot} \end{pmatrix}^T = \sum_{s=1}^{k} a_{\cdot s} b_{s\cdot}^T$$

Auch genannt: Dyadisches Produkt

Auch genannt: Dyadisches Produkt

Vektornormierung

Übergeordnete Normierung: *p-Norm*

Übergeordnete Normierung: *p-Norm*

$$\|x\|_{p} = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}$$

Vektornormierung

Verschiedene Normierungen:

$\|x\|_1 = \sum_{i=1}^{n} \|x_i\|$	Summennorm
$\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$	Euklidische Norm
$\|x\|_\infty = \max_{1 \le i \le n} \|x_i\|$	Maximumsnorm

$\|x\|_1 = \sum_{i=1}^{n} \|x_i\|$	Summennorm
$\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$	Euklidische Norm
$\|x\|_\infty = \max_{1 \le i \le n} \|x_i\|$	Maximumsnorm

$\|x\|_1 = \sum_{i=1}^{n} \|x_i\|$	Summennorm
$\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$	Euklidische Norm
$\|x\|_\infty = \max_{1 \le i \le n} \|x_i\|$	Maximumsnorm

Summennorm

Euklidische Norm

Maximumsnorm

Vektornormierung

- Mithilfe von Normierung sind Fehlerberechnungen möglich
- Vektoren stellen z.B. Wörter/Dokumente dar

- Mithilfe von Normierung sind Fehlerberechnungen möglich
- Vektoren stellen z.B. Wörter/Dokumente dar

Absolute Fehler:

Absoluter Fehler:

$$\|\delta x\| = \|\bar{x} - x\|$$

Relativer Fehler:

Relativer Fehler:

$$\frac{\|\delta x\|}{\|x\|} = \frac{\|\bar{x} - x\|}{\|x\|}$$

Vektornormierung

- Vergleich der Ähnlichkeit zweier Vektoren

→ Kosinus ihrer Winkel

$$\cos\theta(x,y) = \frac{x^T y}{\|x\|_2 \|y\|_2}$$

- Bsp.: Vektoren stellen *Dokumente* dar
- Wenn Kosinus nahe an 1: *Dokumente* besitzen ähnlichen Inhalt

→ Anwendung in **Information Retrieval**

- Vergleich der Ähnlichkeit zweier Vektoren

→ Kosinus ihrer Winkel

- Bsp.: Vektoren stellen *Dokumente* dar
- Wenn Kosinus nahe an 1: *Dokumente* besitzen ähnlichen Inhalt

→ Anwendung in **Information Retrieval**

Lineare Unabhängigkeit

- Basisvektoren (z.B. x, y, z Achsen im klassischen Koordinaten system) als Referenzen für Vektor im Raum
- Def.: Lineare Unabhängigkeit

  Geg.: Vektoren $\left(v_j\right)_{j=1}^{n}$

  $$\sum_{j=1}^{n} \alpha_j v_j = 0, \text{ wenn } \alpha_j = 0 \text{ für } j = 1, 2, \dots, n.$$

Wenn Set von Vektoren dieser Art gegeben: **Basis**

Lineare Unabhängigkeit

- Anwendungsbeispiel:
- Geg.: Set von linear abhängigen Vektoren (V)

    Subset von unabhängigen V
- Ausdrücken der abhängigen V mithilfe des Subsets
- Anzahl der unabhängigen V als Maß der Information, die im S
  et beinhaltet wird
- Komprimierung der Vektoren

Datenreduktion

Lineare Unabhängigkeit

Wichtig:

- In Realität ist lineare Unabhängigkeit fast nie möglich
- Stattdessen wichtiges Kriterium für Basisvektoren: **Orthogon alität**

Vielen Dank!