VERİ MADENCİLİĞİ
Metin Madenciliği

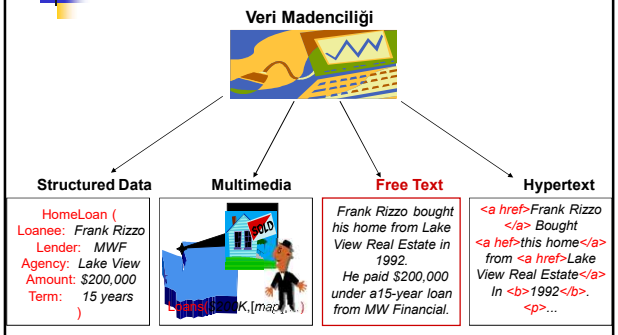
Prof. Dr. Şule Gündüz Öğüdücü

http://www3.itu.edu.tr/~sgunduz/courses/verimaden/

Konular

- Metin madenciliği & Bilgi erişim sistemleri
- Dizinleme
- Metin ön işleme
- İçerik tabanlı sıralama
- Olasılıklı bilgi erişim sistemleri
- Gizli anlamsal inceleme
- Boyut azaltma

Metin için Veri Madenciliği



Metin için Veri Madenciliği

- Metin madenciliği: Veri madenciliği teknikleri ile yazılı belgeler arasındaki (içindeki) ilişkileri, örüntüleri bulmak.
  - doğal dilde yazılmış metinler
  - Aynı konudaki belgeleri bulmak
  - Birbiriyle ilişkili belgeleri bulmak
  - Bulunan belgeleri sıralamak
- Bilgi elde etme sistemleri birleştirilebilir
  - Ortak ön işleme işlemleri: doğal dil işleme yöntemleri ile
  - bilgi elde etme sistemleri metin madenciliği sonuçlarını değerlendirmeli, yorumlamalı

Metin Madenciliğinde Sorunlar

- Metin yapısal değil
  - Sorgulama sonucuyla ilintili metinleri bulmak zor
  - Önemsiz veri (sözcük) çok fazla
- Hatalar var
  - Metin içinde hatalar var
  - Kavram oluşturmak zor: eş anlamlı, eşsesli, üst grup
  - Anlam çıkarmak zor: X'in iyi bir bilgisayar olduğunu düşünüyorum - X'in iyi bir bilgisayar olduğundan şüpheliyim
- Sonuç
  - Belgeler arasındaki ilişkilendirme hatalı

Metin Madenciliği Araçları

- Ticari ürünler
  - http://www.clearforest.com/
  - http://www.trl.ibm.com/projects/textmining/takmi/takmi_e.htm
  - http://www.megaputer.com/
- Metin inceleme
  - http://www.textanalysis.info/
- Problem
  - Dile özel çözümler
  - Sonuçlar yetersiz

Metin Veritabanları & Bilgi Erişim Sistemleri

- Metin Veri tabanları (belge veri tabanları)
  - Farklı kaynaklardan dokümanlar: haber, makale, kitap, elektronik kütüphane, elektronik posta, web sayfaları..
  - Veri genelde yapısal değil
  - Bilgi erişim sistemleri büyük miktardaki veri üzerinde başarılı değil
- Bilgi Erişim (Information Retrival) Sistemleri
  - Veri tabanları ile birlikte gelişmiş bir araştırma alanı
  - Bilgi belgeler şeklinde yer alıyor

7

Bilgi Erişim Sistemi

- Kullanıcının ilgi alanına ve isteğine en uygun belgeleri bulma
  - Kullanıcın girdiği bir sorgulamaya göre
  - Kullanıcının ziyaret ettiği sayfalara göre
- Internet ortamında web sayfalarının içeriğinin incelenmesini gerektirir
- Bilgi erişim yönteminde problemler
  - Büyük bir belgeler kümesindeki belgeleri işaretleme
    - erişimin kolay olması için
  - Seçilen belgelerin sıralanması
  - Belgelerin sınıflandırılması: veri madenciliği yöntemleri kullanılabilir

8

Konular

- Metin madenciliği & Bilgi erişim sistemleri
- Dizinleme
- Metin ön işleme
- İçerik tabanlı sıralama
- Olasılıklı bilgi erişim sistemleri
- Gizli anlamsal inceleme
- Boyut azaltma

9

Dizin Oluşturma

- Ters dizin
  - Belgelerden oluşan veri kümesinde her sözcüğün hangi belgelerde görüldüğü işaretlenir
  - Büyük veri kümeleri için etkili
- Terim ω
  - sözcükler veya ifadeler
- Sözcük dağarcığı V
  - Terimlerden oluşan küme

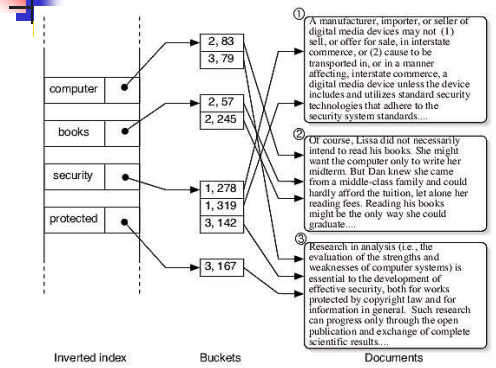
10

Ters Dizin

- Sözlük
  - her anahtar bir terim $\omega \in V$
  - anahtara ait veriler zincir (atama listesi) $b(\omega)$: ω teriminin, belgeler kümesinde her görüldüğü yeri işaret eden işaretçiler listesi
    - belge kimlik numarası (DID): belgenin küme içinde kaçıncı sırada yer aldığı
    - terimin her görüldüğü yer için ayrı bir işaretçi
      - DID
      - Terimin belge içindeki konumu

11

Ters Dizin



12

Ters Dizin Oluşturma

- Belgeler ayrıştırılır
- Terimler bulunur ω_i
  - Eğer ω_i dizinde yer almıyorsa eklenir
- Terimin bulunduğu yer zincire eklenir
- Ters dizin boyutu = $\Omega(|V|)$
- Hash tablosu kullanarak gerçeklenebilir
  - Zincirler bellekte
  - Zincirler diskte
    - Diske erişim süresinden dolayı elverişsiz
      - Özel ikincil depolama yapıları kullanılması gerekir

13

Zincirlerin Sıkıştırılması

- Zincir için gerekli saklama alanını azaltma
  - her terim için zincir DID'ye göre sıralanır
  - DID'ler arasındaki fark saklanır
- Bellek kullanımı önemli ölçüde azalır
  - Belgeler kümesinde sık yer alan terimlerin DID'leri arasında fark da azdır
  - Küçük sayılar kodlanarak bellekte daha az yer kaplarlar
- Örnek
  - DID listesi: (14, 22, 38, 42, 66, 122, 131, 226)
  - DID'ler arasındaki fark listesi: (14, 8, 16, 4, 24, 56, 9, 95)

14

Ters Dizin ile Arama

- Bir belgeler kümesi için oluşturulmuş ters dizinde bir terimi ω bulmak için
  - ters dizinde ω terimine ait zincir $b(\omega)$ bulunur
  - zincir taranarak terimin bulunduğu yerlerin listesi elde edilir
- Bir belgeler kümesi için oluşturulmuş ters dizinde k adet terim bulmak için
  - k adet liste oluşturulur
  - küme işlemleri ile listeler birleştirilir

15

Konular

- Metin madenciliği & Bilgi erişim sistemleri
- Dizinleme
- Metin ön işleme
- İçerik tabanlı sıralama
- Olasılıklı bilgi erişim sistemleri
- Gizli anlamsal inceleme
- Boyut azaltma

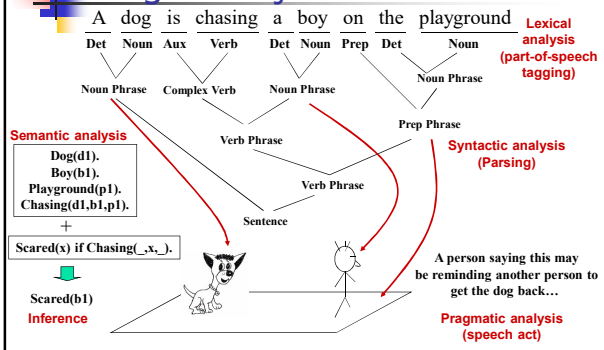
16

Metin Ön İşleme

- Belgeler için dizin oluşturmadan önce ön işleme işlemleri
  - İşaretleme
    - Metin içindeki terimleri ayıklama
    - HTML etiketlerinden arındırma
  - Farklı terimleri belirleme
    - Kök bulma: Aynı kökten gelen farklı ek almış sözcüklerin köklerini bulma
    - Sık geçen sözcükleri ayıklama: bağlaçlar, edatlar
    - Terim sayısında %20-30 oranında azalma

17

Doğal Dil İşleme

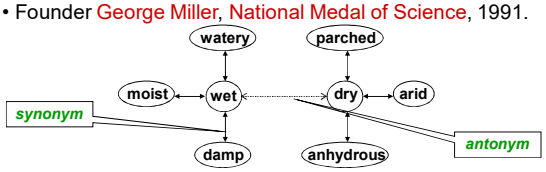


18

Kavramsal Sözlük: WordNet

An extensive lexical network for the English language

- Contains over 138,838 words.
- Several graphs, one for each part-of-speech.
- *Synsets* (synonym sets), each defining a semantic sense.
- Relationship information (antonym, hyponym, meronym …)
- Downloadable for free (UNIX, Windows)
- Expanding to other languages (Global WordNet Association)
- Funded >$3 million, mainly government (translation interest)
- Founder George Miller, National Medal of Science, 1991.



19

Kök Bulma

- Sözcüklerin biçimbirimsel çözümlemesini yaparak terimleri elde etmek
  - Örnek: İçinde *balıkçılık* sözcüğü geçen bir sorgulama için, içinde *balık* ve *balıkçı* geçen belgelerin bulunması
- İngilizce için kök bulma: Porter Stemming Algorithm
  - http://www.tartarus.org/~martin/PorterStemmer/
- Türkçe için kök bulma: Zemberek projesi
  - ITU Türkçe Doğal Dil İşleme Yazılım Zinciri
  - https://zemberek.dev.java.net/

20

Konular

- Metin madenciliği & Bilgi erişim sistemleri
- Dizinleme
- Metin ön işleme
- İçerik tabanlı sıralama
- Olasılıklı bilgi erişim sistemleri
- Gizli anlamsal inceleme
- Boyut azaltma

21

Sorgulama Sonuçlarını Sıralama

- Sorgulama içinde geçen terimlerin yer aldığı belgelerin sayısı çok fazla
  - *data mining* için Google arama motorunda dönen sonuç: 514.000.000
- kullanıcı ancak küçük bir kısmını inceleyebilir
  - sorgulama sonuçlarını sıralamak gerekir
    - sorgulamayla daha ilgili olan sonuçların başlarda yer alması

22

Vektör Uzayı Modeli

- Belgeler çok boyutlu vektör uzayında temsil edilir
- Belgeler terim vektörleri biçiminde
  $d = (\omega(1), \omega(2), \omega(3), \ldots, \omega(|d|))$
- Belgeler kümesindeki ayrık terim sayısı vektör uzayının boyutunu belirler

23

Örnek

belge	metin	terimler
d_1	*web web graph*	*web graph*
d_2	*graph web net graph net*	*graph web net*
d_3	*page web complex*	*page web complex*

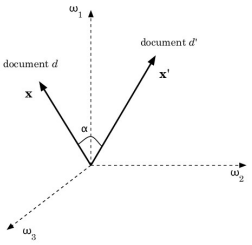
- Belgelerin boolean modeline göre temsil edilmesi:

V = [web, graph, net, page, complex]
V1 = [1 1 0 0 0]
V2 = [1 1 1 0 0]
V3 = [1 0 0 1 1]

24

Vektör Uzayı Modeli

- x, x': belge vektörleri
- ω_1, ω_2 , ω_3 :terimler
- Vektör uzayında belgelerin gösterilimi seyrek: |V|>>|d|



25

Terim Sıklığı (TF)

- Bir belge içinde, diğer terimlere göre daha sık yer alan bir terimin önemi de daha fazladır
- n_{ij} : ω_j teriminin d_i belgesinde yer alma sayısı
- Terim sıklığı (term frequency):

$$TF_{ij} = \frac{n_{ij}}{|d_i|}$$

26

Devrik Belge Sıklığı (IDF)

- Belgeler kümesinde daha az sayıda belgede görülen bir terimin ayırt edici özelliği daha fazladır
- n_j : belge sıklığı (document frequency: df) - ω_j teriminin geçtiği belge sayısı
- n : belgeler kümesindeki belge sayısı
- ω_j teriminin devrik belge sıklığı (Inverse Document Frequency):

$$IDF_j = \log\frac{n}{n_j}$$

- Belge sıklığı (df) arttıkça, devrik belge sıklığı (idf) azalır

27

Tam Ağırlıklandırma (TF-IDF)

- Bir belgede çok bulunan ancak diğer belgelerde daha az görülen bir terimin ağırlığı daha fazla
- ω_j teriminin d_i belgesindeki TF-IDF ağırlığı:

$$x_{ij} = TF_{ij} \times IDF_j$$

28

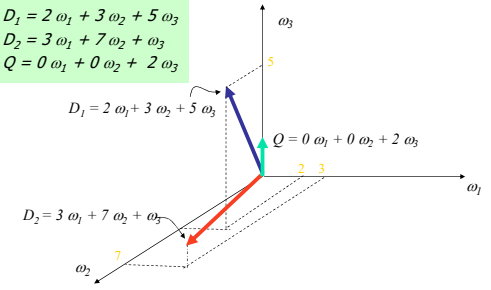
Belgeler Arası Benzerlik

- Sorgulama ile herbir belge arasındaki benzerlik hesaplanıp, benzerlik sonucuna göre sıralanır
- Herhangi iki belge arasındaki benzerlik $s(d, d') \in R$
- Vektör uzayı modelinde kosinüs benzerliği belgeler arası benzerliği hesaplamak için kullanılabilir

29

Grafik Gösterilim

- Örnek:
  $D_1 = 2\,\omega_1 + 3\,\omega_2 + 5\,\omega_3$
  $D_2 = 3\,\omega_1 + 7\,\omega_2 + \omega_3$
  $Q = 0\,\omega_1 + 0\,\omega_2 + 2\,\omega_3$



30

Kosinüs Benzerliği

- İki vektör arasındaki açının kosinüsü
  $\cos(d_1,d_2)=d_1 \bullet d_2 / ||d_1|| \, ||d_2||$
  $d_i \bullet d_j$: iki dokümanın vektör çarpımı
  $||d_i||$: d_i dokümanının uzunluğu
- **Örnek**
  d_1 = 3 2 0 5 0 0 0 2 0 0
  d_2 = 1 0 0 0 0 0 0 1 0 2

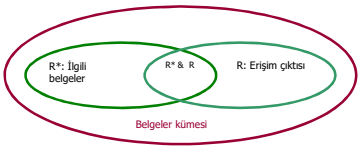
$d_1 \bullet d_2$= 3*1 + 2*0 + 0*0 + 5*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2 = 5
$||d_1||$ = (3*3+2*2+0*0+5*5+0*0+0*0+0*0+2*2+0*0+0*0)$^{0.5}$ = (42) $^{0.5}$ = 6.481
$||d_2||$ = (1*1+0*0+0*0+0*0+0*0+0*0+0*0+1*1+0*0+2*2)$^{0.5}$ = (6) $^{0.5}$ = 2.245

cos(d_1, d_2) = .3150

Sonuçların Değerlendirilmesi

- Belgeler kümesindeki belgelerin vektör uzayı modeli bulunur
- Kullanıcı sorgusu *Q* için vektör uzayı modeli bulunur
- Belgeler kümesindeki her belge için sorgulamayla olan benzerliği hesaplanır *s(d$_i$,Q), i=1,2,...,n*
- En fazla benzerlik değerine sahip olan belgeler kümesi *R* erişim çıktısı olarak belirlenir
- Belgeler kümesinde sorgulamayla ilgili belgeler kümesi *R** ve *R* karşılaştırılır.

Sonuçların Değerlendirilmesi



- Kesinlik (Precision): Erişim çıktısındaki ilgili belge sayısının erişim çıktısındaki belge sayısına oranı

$$precision = \frac{|R^* \cap R|}{R}$$

- Duyarlılık (Recall): Erişim çıktısındaki ilgili belge sayısının belgeler kümesinde ilgili belgeler sayısına oranı

$$recall = \frac{|R^* \cap R|}{|R^*|}$$

Konular

- Metin madenciliği & Bilgi erişim sistemleri
- Dizinleme
- Metin ön işleme
- İçerik tabanlı sıralama
- Olasılıklı bilgi erişim sistemleri
- Gizli anlamsal inceleme
- Terim sayısını azaltma

Olasılıklı Bilgi Erişim Sistemleri

- Temel varsayım: Kullanıcının sorgulamasına göre sadece ilgili belgelerden oluşan bir belgeler kümesi var (ideal durum)
- Probabilistic Ranking Principle (PRP) (Robertson, 1977)
  - Belgelerin kullanıcının sorgulamasına ilgili olma olasılığına göre sıralanması
  - Olasılıklar eldeki veriye göre mümkün olan en doğru şekilde hesaplanır
  - Eldeki veri ile gerçeklenebilecek en iyi sistem

Olasılıklı Bilgi Erişim Sistemleri

- Sorgu terimlerinin bir belgede bulunabilme olasılığı *P(R | d,q)*
- Belgeler olasılıklara azalacak şekilde sıralanır

$$P(R \mid d,q) \geq P(R \mid d',q)$$

d' : erişim çıktısında yer almayan belge

Konular

- Metin madenciliği & Bilgi erişim sistemleri
- Dizinleme
- Metin ön işleme
- İçerik tabanlı sıralama
- Olasılıklı bilgi erişim sistemleri
- Gizli anlamsal inceleme
- Boyut azaltma

37

Gizli Anlamsal İnceleme

- Latent Semantic Analysis
- Vektör uzayı modeli sorgulama içinde geçen terimler belge içinde de yer alıyorsa iyi sonuç veriyor
- Doğal dilin zenginliği nedeniyle sorunlar
  - Kullanıcı sorgulamalarında genelde kavramlar yer alıyor
  - eş anlamlı sözcükler: hediye – armağan
    - duyarlılık değerini etkiliyor
  - eş sesli sözcükler: çay
    - kesinlik değerini etkiliyor

38

Latent Semantic Indexing (LSI)

- Lineer cebirdeki tekil değer ayrışımı (singular value decomposition, SVD) yöntemi kullanılır
  - Veri içindeki gizli yapıyı bulmayı hedefler
  - Sözcükler ve kavramlar arasındaki önemli ilişkileri bulmayı hedefler
- ***D***: terim – belge matrisi $\boldsymbol{D} = [\boldsymbol{d}_1 \ldots \boldsymbol{d}_n]^T$
  - her satır belgelerin vektör uzayı modelindeki gösterilimi
  - her kolon terimin belgede yer alma sayısı
- ***D*** matrisinin tekil değer ayrışım matrisi **Σ** hesaplanır

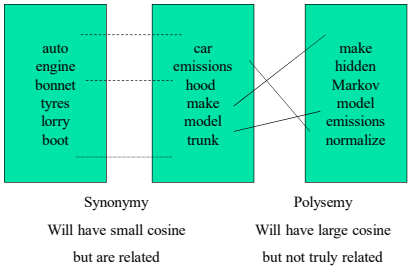
$$D = U\Sigma V^T$$

- Tekil değer ayrışım matrisindeki en büyük K değer dışındakiler sıfırlanır $\hat{\Sigma}$
- **D** terim belge matrisi yeniden oluşturulur $\hat{D} = U\hat{\Sigma}V^T$

39

The Problem

- Example: Vector Space Model
  - (from Lillian Lee)



The Problem

- Latent Semantic Indexing was proposed to address these two problems with the vector space model for Information Retrieval

Some History

- Latent Semantic Indexing was developed at Bellcore (now Telcordia) in the late 1980s (1988). It was patented in 1989.
- http://lsi.argreenhouse.com/lsi/LSI.html

Some History

- The first papers about LSI:
  - Dumais, S. T., Furnas, G. W., Landauer, T. K. and Deerwester, S. (1988), "Using latent semantic analysis to improve information retrieval." In Proceedings of CHI'88: Conference on Human Factors in Computing, New York: ACM, 281-285.
  - Deerwester, S., Dumais, S. T., Landauer, T. K., Furnas, G. W. and Harshman, R.A. (1990) "Indexing by latent semantic analysis." Journal of the Society for Information Science, 41(6), 391-407.
  - Foltz, P. W. (1990) "Using Latent Semantic Indexing for Information Filtering". In R. B. Allen (Ed.) Proceedings of the Conference on Office Information Systems, Cambridge, MA, 40-47.

LSA

- But first:
- What is the difference between LSI and LSA???
  - LSI refers to using it for indexing or information retrieval.
  - LSA refers to everything else.

LSA

- Idea (Deerwester et al):

"We would like a representation in which a set of terms, which by itself is incomplete and unreliable evidence of the relevance of a given document, is replaced by some other set of entities which are more reliable indicants. We take advantage of the implicit higher-order (or latent) structure in the association of terms and documents to reveal such relationships."

LSA

- Implementation: four basic steps
  - term by document matrix (more generally term by context) tend to be sparce
  - convert matrix entries to weights, typically:
    - L(i,j) * G(i): local and global
    - a_ij -> log(freq(a_ij)) divided by entropy for row (-sum (p logp), over p: entries in the row)
      - weight directly by estimated importance in passage
      - weight inversely by degree to which knowing word occurred provides information about the passage it appeared in

LSA

- Four basic steps
  - Rank-reduced Singular Value Decomposition (SVD) performed on matrix
    - all but the k highest singular values are set to 0
    - produces k-dimensional approximation of the original matrix (in least-squares sense)
    - this is the "semantic space"
  - Compute similarities between entities in semantic space (usually with cosine)

LSA

- SVD
  - unique mathematical decomposition of a matrix into the product of three matrices:
    - two with orthonormal columns
    - one with singular values on the diagonal
  - tool for dimension reduction
  - similarity measure based on co-occurrence
  - finds optimal projection into low-dimensional space

LSA

- SVD
  - can be viewed as a method for rotating the axes in n-dimensional space, so that the first axis runs along the direction of the largest variation among the documents
    - the second dimension runs along the direction with the second largest variation
    - and so on
  - generalized least-squares method

A Small Example

- To see how this works let's look at a small example
- This example is taken from: Deerwester, S.,Dumais, S.T., Landauer, T.K.,Furnas, G.W. and Harshman, R.A. (1990). "Indexing by latent semantic analysis." *Journal of the Society for Information Science, 41(6)*, 391-407.
- Slides are from a presentation by Tom Landauer and Peter Foltz

A Small Example

Technical Memo Titles

c1: *Human* machine *interface* for ABC *computer* applications
c2: A *survey* of *user* opinion of *computer system response time*
c3: The *EPS user interface* management *system*
c4: *System* and *human system* engineering testing of *EPS*
c5: Relation of *user* perceived *response time* to error measurement

m1: The generation of random, binary, ordered *trees*
m2: The intersection *graph* of paths in *trees*
m3: *Graph minors* IV: Widths of *trees* and well-quasi-ordering
m4: *Graph minors*: A *survey*

A Small Example – 2

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

sim (human.user) = 0 sim(user.minors) = 0

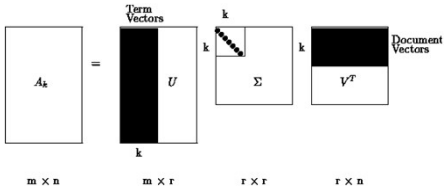
A Small Example – 3

- Singular Value Decomposition

$$\{A\}=\{U\}\{S\}\{V\}^T$$

- Dimension Reduction

$$\{\sim A\}\sim=\{\sim U\}\{\sim S\}\{\sim V\}^T$$



A Small Example – 4

- {U} =

0.22	-0.11	0.29	-0.41	-0.11	-0.34	0.52	-0.06	-0.41
0.20	-0.07	0.14	-0.55	0.28	0.50	-0.07	-0.01	-0.11
0.24	0.04	-0.16	-0.59	-0.11	-0.25	-0.30	0.06	0.49
0.40	0.06	-0.34	0.10	0.33	0.38	0.00	0.00	0.01
0.64	-0.17	0.36	0.33	-0.16	-0.21	-0.17	0.03	0.27
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.30	-0.14	0.33	0.19	0.11	0.27	0.03	-0.02	-0.17
0.21	0.27	-0.18	-0.03	-0.54	0.08	-0.47	-0.04	-0.58
0.01	0.49	0.23	0.03	0.59	-0.39	-0.29	0.25	-0.23
0.04	0.62	0.22	0.00	-0.07	0.11	0.16	-0.68	0.23
0.03	0.45	0.14	-0.01	-0.30	0.28	0.34	0.68	0.18

A Small Example – 5

- {S} =

3.34								
	2.54							
		2.35						
			1.64					
				1.50				
					1.31			
						0.85		
							0.56	
								0.36

A Small Example – 6

- {V} =

0.20	0.61	0.46	0.54	0.28	0.00	0.01	0.02	0.08
-0.06	0.17	-0.13	-0.23	0.11	0.19	0.44	0.62	0.53
0.11	-0.50	0.21	0.57	-0.51	0.10	0.19	0.25	0.08
-0.95	-0.03	0.04	0.27	0.15	0.02	0.02	0.01	-0.03
0.05	-0.21	0.38	-0.21	0.33	0.39	0.35	0.15	-0.60
-0.08	-0.26	0.72	-0.37	0.03	-0.30	-0.21	0.00	0.36
0.18	-0.43	-0.24	0.26	0.67	-0.34	-0.15	0.25	0.04
-0.01	0.05	0.01	-0.02	-0.06	0.45	-0.76	0.45	-0.07
-0.06	0.24	0.02	-0.08	-0.26	-0.62	0.02	0.52	-0.45

A Small Example – 7

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	0.16	0.40	0.38	0.47	0.18	-0.05	-0.12	-0.16	-0.09
interface	0.14	0.37	0.33	0.40	0.16	-0.03	-0.07	-0.10	-0.04
computer	0.15	0.51	0.36	0.41	0.24	0.02	0.06	0.09	0.12
user	0.26	0.84	0.61	0.70	0.39	0.03	0.08	0.12	0.19
system	0.45	1.23	1.05	1.27	0.56	-0.07	-0.15	-0.21	-0.05
response	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
time	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
EPS	0.22	0.55	0.51	0.63	0.24	-0.07	-0.14	-0.20	-0.11
survey	0.10	0.53	0.23	0.21	0.27	0.14	0.31	0.44	0.42
trees	-0.06	0.23	-0.14	-0.27	0.14	0.24	0.55	0.77	0.66
graph	-0.06	0.34	-0.15	-0.30	0.20	0.31	0.69	0.98	0.85
minors	-0.04	0.25	-0.10	-0.21	0.15	0.22	0.50	0.71	0.62

sim(human.user) = .89 sim (user.minors) = -.27

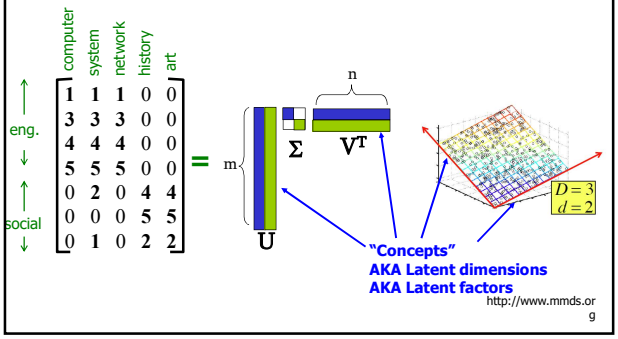
Konular

- Metin madenciliği & Bilgi erişim sistemleri
- Dizinleme
- Metin ön işleme
- İçerik tabanlı sıralama
- Olasılıklı bilgi erişim sistemleri
- Gizli anlamsal inceleme
- Boyut azaltma

58

SVD – Example: Documents-to-Terms

- **A = U Σ V^T - example: Documents to Terms**



http://www.mmds.org

SVD – Example: Documents-to-Terms

- **A = U Σ V^T - example: Documents to Terms**

	computer	system	network	history	art
eng.	1	1	1	0	0
eng.	3	3	3	0	0
eng.	4	4	4	0	0
social	5	5	5	0	0
social	0	2	0	4	4
social	0	0	0	5	5
social	0	1	0	2	2

=

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**

x

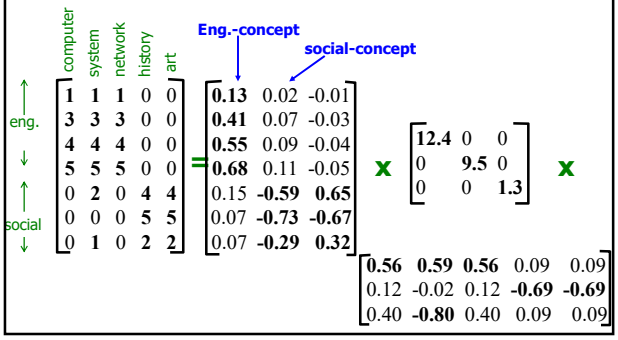
12.4	0	0
0	**9.5**	0
0	0	**1.3**

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

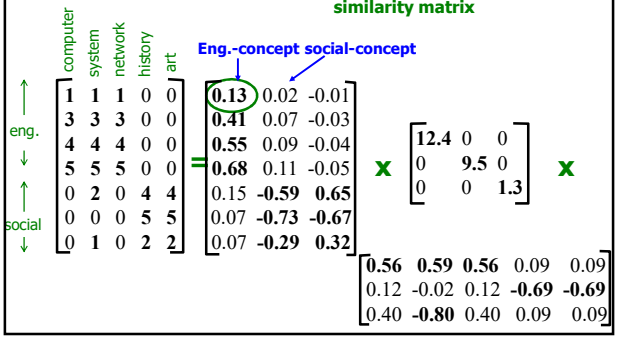
SVD – Example: Documents-to-Terms

- **A = U Σ V^T - example: Documents to Terms**



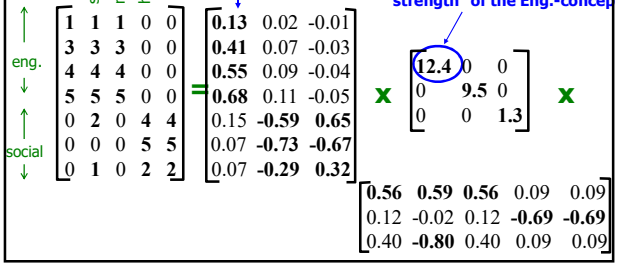
SVD – Example: Documents-to-Terms

- **A = U Σ V^T - example:** ***U*** **is "document-to-concept" similarity matrix**



SVD – Example: Documents-to-Terms

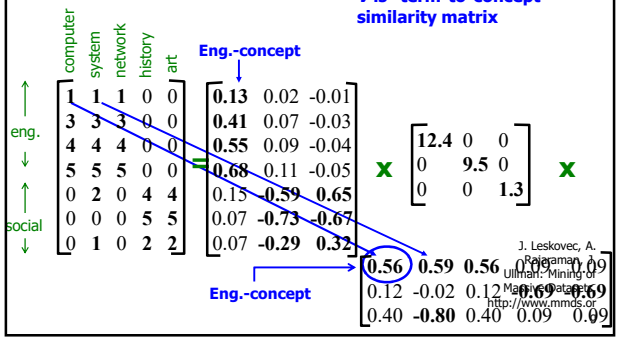
- **A = U Σ V^T - example:**



SVD – Example: Documents-to-Terms

- **A = U Σ V^T - example:**

V **is "term-to-concept" similarity matrix**



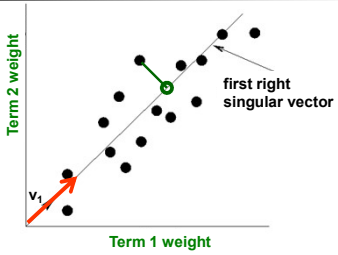
J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets, http://www.mmds.org

SVD - Interpretation #1

'**terms**', '**documents**' and '**concepts**':

- ***U***: document-to-concept similarity matrix
- ***V***: term-to-concept similarity matrix
- **Σ**: its diagonal elements:
  'strength' of each concept

SVD – Dimensionality Reduction



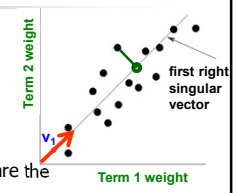
- Instead of using two coordinates (x, y) to describe point locations, let's use only one coordinate (z)
- Point's position is its location along vector v_1
- **How to choose v_1? Minimize reconstruction error**

SVD – Dimensionality Reduction

- **Goal: Minimize the sum of reconstruction errors:**

$$\sum_{i=1}^{N}\sum_{j=1}^{D}\left\|x_{ij}-z_{ij}\right\|^2$$

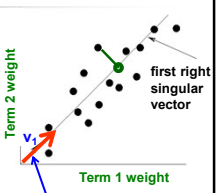
  - where x_{ij} are the "old" and z_{ij} are the "new" coordinates



- **SVD gives 'best' axis to project on:**
  - **'best'** = minimizing the reconstruction errors
- **In other words, minimum reconstruction error**

SVD - Interpretation #2

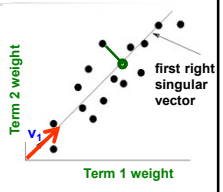
- **A = U Σ V^T - example:**
  - **V**: "term-to-concept" matrix
  - **U**: "document-to-concept" matrix



$$\begin{bmatrix}1&1&1&0&0\\3&3&3&0&0\\4&4&4&0&0\\5&5&5&0&0\\0&2&0&4&4\\0&0&0&5&5\\0&1&0&2&2\end{bmatrix}=\begin{bmatrix}\mathbf{0.13}&0.02&-0.01\\\mathbf{0.41}&0.07&-0.03\\\mathbf{0.55}&0.09&-0.04\\\mathbf{0.68}&0.11&-0.05\\0.15&\mathbf{-0.59}&\mathbf{0.65}\\0.07&\mathbf{-0.73}&\mathbf{-0.67}\\0.07&\mathbf{-0.29}&\mathbf{0.32}\end{bmatrix}\times\begin{bmatrix}\mathbf{12.4}&0&0\\0&\mathbf{9.5}&0\\0&0&\mathbf{1.3}\end{bmatrix}\times\begin{bmatrix}\mathbf{0.56}&\mathbf{0.59}&\mathbf{0.56}&0.09&0.09\\0.12&-0.02&0.12&\mathbf{-0.69}&\mathbf{-0.69}\\0.40&\mathbf{-0.80}&0.40&0.09&0.09\end{bmatrix}$$

SVD - Interpretation #2

- **A = U Σ V^T - example:**

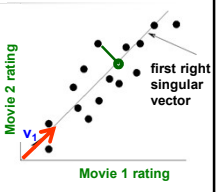


variance ('spread') on the v_1 axis

$$\begin{bmatrix}1&1&1&0&0\\3&3&3&0&0\\4&4&4&0&0\\5&5&5&0&0\\0&2&0&4&4\\0&0&0&5&5\\0&1&0&2&2\end{bmatrix}=\begin{bmatrix}\mathbf{0.13}&0.02&-0.01\\\mathbf{0.41}&0.07&-0.03\\\mathbf{0.55}&0.09&-0.04\\\mathbf{0.68}&0.11&-0.05\\0.15&\mathbf{-0.59}&\mathbf{0.65}\\0.07&\mathbf{-0.73}&\mathbf{-0.67}\\0.07&\mathbf{-0.29}&\mathbf{0.32}\end{bmatrix}\times\begin{bmatrix}\mathbf{12.4}&0&0\\0&\mathbf{9.5}&0\\0&0&\mathbf{1.3}\end{bmatrix}\times\begin{bmatrix}\mathbf{0.56}&\mathbf{0.59}&\mathbf{0.56}&0.09&0.09\\0.12&-0.02&0.12&\mathbf{-0.69}&\mathbf{-0.69}\\0.40&\mathbf{-0.80}&0.40&0.09&0.09\end{bmatrix}$$

SVD - Interpretation #2

- **A = U Σ V^T - example:**
  - **U Σ**: Gives the coordinates of the points in the projection axis



Projection of documents on the "Eng" axis (U Σ)^T:

$$\begin{bmatrix}1&1&1&0&0\\3&3&3&0&0\\4&4&4&0&0\\5&5&5&0&0\\0&2&0&4&4\\0&0&0&5&5\\0&1&0&2&2\end{bmatrix}\qquad\begin{bmatrix}1.61&0.19&-0.01\\5.08&0.66&-0.03\\6.82&0.85&-0.05\\8.43&1.04&-0.06\\1.86&-5.60&0.84\\0.86&-6.93&-0.87\\0.86&-2.75&0.41\end{bmatrix}$$

SVD - Interpretation #2

More details

- **How exactly is dim. reduction done?**

$$\begin{bmatrix}1&1&1&0&0\\3&3&3&0&0\\4&4&4&0&0\\5&5&5&0&0\\0&2&0&4&4\\0&0&0&5&5\\0&1&0&2&2\end{bmatrix}=\begin{bmatrix}\mathbf{0.13}&0.02&-0.01\\\mathbf{0.41}&0.07&-0.03\\\mathbf{0.55}&0.09&-0.04\\\mathbf{0.68}&0.11&-0.05\\0.15&\mathbf{-0.59}&\mathbf{0.65}\\0.07&\mathbf{-0.73}&\mathbf{-0.67}\\0.07&\mathbf{-0.29}&\mathbf{0.32}\end{bmatrix}\times\begin{bmatrix}\mathbf{12.4}&0&0\\0&\mathbf{9.5}&0\\0&0&\mathbf{1.3}\end{bmatrix}\times\begin{bmatrix}\mathbf{0.56}&\mathbf{0.59}&\mathbf{0.56}&0.09&0.09\\0.12&-0.02&0.12&\mathbf{-0.69}&\mathbf{-0.69}\\0.40&\mathbf{-0.80}&0.40&0.09&0.09\end{bmatrix}$$

SVD - Interpretation #2

More details

- **How exactly is dim. reduction done?**
- **A: Set smallest singular values to zero**

$$\begin{bmatrix}1&1&1&0&0\\3&3&3&0&0\\4&4&4&0&0\\5&5&5&0&0\\0&2&0&4&4\\0&0&0&5&5\\0&1&0&2&2\end{bmatrix}=\begin{bmatrix}\mathbf{0.13}&0.02&-0.01\\\mathbf{0.41}&0.07&-0.03\\\mathbf{0.55}&0.09&-0.04\\\mathbf{0.68}&0.11&-0.05\\0.15&\mathbf{-0.59}&\mathbf{0.65}\\0.07&\mathbf{-0.73}&\mathbf{-0.67}\\0.07&\mathbf{-0.29}&\mathbf{0.32}\end{bmatrix}\times\begin{bmatrix}\mathbf{12.4}&0&0\\0&\mathbf{9.5}&0\\0&0&\mathbf{1.3}\end{bmatrix}\times\begin{bmatrix}\mathbf{0.56}&\mathbf{0.59}&\mathbf{0.56}&0.09&0.09\\0.12&-0.02&0.12&\mathbf{-0.69}&\mathbf{-0.69}\\0.40&\mathbf{-0.80}&0.40&0.09&0.09\end{bmatrix}$$

SVD - Interpretation #2

More details

- How exactly is dim. reduction done?
- A: Set smallest singular values to zero

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 2 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 1 & 0 & 2 & 2 \end{bmatrix} \approx \begin{bmatrix} \mathbf{0.13} & 0.02 & -0.01 \\ \mathbf{0.41} & 0.07 & -0.03 \\ \mathbf{0.55} & 0.09 & -0.04 \\ \mathbf{0.68} & 0.11 & -0.05 \\ 0.15 & \mathbf{-0.59} & \mathbf{0.65} \\ 0.07 & \mathbf{-0.73} & \mathbf{-0.67} \\ 0.07 & \mathbf{-0.29} & \mathbf{0.32} \end{bmatrix} \times \begin{bmatrix} \mathbf{12.4} & 0 & 0 \\ 0 & \mathbf{9.5} & 0 \\ 0 & 0 & \mathbf{1.3} \end{bmatrix} \times \begin{bmatrix} \mathbf{0.56} & \mathbf{0.59} & \mathbf{0.56} & 0.09 & 0.09 \\ 0.12 & -0.02 & 0.12 & \mathbf{-0.69} & \mathbf{-0.69} \\ 0.40 & \mathbf{-0.80} & 0.40 & 0.09 & 0.09 \end{bmatrix}$$

SVD - Interpretation #2

More details

- How exactly is dim. reduction done?
- A: Set smallest singular values to zero

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 2 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 1 & 0 & 2 & 2 \end{bmatrix} \approx \begin{bmatrix} \mathbf{0.13} & 0.02 & -0.01 \\ \mathbf{0.41} & 0.07 & -0.03 \\ \mathbf{0.55} & 0.09 & -0.04 \\ \mathbf{0.68} & 0.11 & -0.05 \\ 0.15 & \mathbf{-0.59} & \mathbf{0.65} \\ 0.07 & \mathbf{-0.73} & \mathbf{-0.67} \\ 0.07 & \mathbf{-0.29} & \mathbf{0.32} \end{bmatrix} \times \begin{bmatrix} \mathbf{12.4} & 0 & 0 \\ 0 & \mathbf{9.5} & 0 \\ 0 & 0 & \mathbf{1.3} \end{bmatrix} \times \begin{bmatrix} \mathbf{0.56} & \mathbf{0.59} & \mathbf{0.56} & 0.09 & 0.09 \\ 0.12 & -0.02 & 0.12 & \mathbf{-0.69} & \mathbf{-0.69} \\ 0.40 & \mathbf{-0.80} & 0.40 & 0.09 & 0.09 \end{bmatrix}$$

SVD - Interpretation #2

More details

- How exactly is dim. reduction done?
- A: Set smallest singular values to zero

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 2 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 1 & 0 & 2 & 2 \end{bmatrix} \approx \begin{bmatrix} \mathbf{0.13} & 0.02 \\ \mathbf{0.41} & 0.07 \\ \mathbf{0.55} & 0.09 \\ \mathbf{0.68} & 0.11 \\ 0.15 & \mathbf{-0.59} \\ 0.07 & \mathbf{-0.73} \\ 0.07 & \mathbf{-0.29} \end{bmatrix} \times \begin{bmatrix} \mathbf{12.4} & 0 \\ 0 & \mathbf{9.5} \end{bmatrix} \times \begin{bmatrix} \mathbf{0.56} & \mathbf{0.59} & \mathbf{0.56} & 0.09 & 0.09 \\ 0.12 & -0.02 & 0.12 & \mathbf{-0.69} & \mathbf{-0.69} \end{bmatrix}$$

SVD - Interpretation #2

More details

- How exactly is dim. reduction done?
- A: Set smallest singular values to zero

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 2 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 1 & 0 & 2 & 2 \end{bmatrix} \approx \begin{bmatrix} \mathbf{0.92} & \mathbf{0.95} & \mathbf{0.92} & 0.01 & 0.01 \\ \mathbf{2.91} & \mathbf{3.01} & \mathbf{2.91} & -0.01 & -0.01 \\ \mathbf{3.90} & \mathbf{4.04} & \mathbf{3.90} & 0.01 & 0.01 \\ \mathbf{4.82} & \mathbf{5.00} & \mathbf{4.82} & 0.03 & 0.03 \\ 0.70 & \mathbf{0.53} & 0.70 & \mathbf{4.11} & \mathbf{4.11} \\ -0.69 & 1.34 & -0.69 & \mathbf{4.78} & \mathbf{4.78} \\ 0.32 & \mathbf{0.23} & 0.32 & \mathbf{2.01} & \mathbf{2.01} \end{bmatrix}$$

Frobenius norm:

$\|M\|_F = \sqrt{\Sigma_{ij} M_{ij}^2}$

$\|A-B\|_F = \sqrt{\Sigma_{ij} (A_{ij}-B_{ij})^2}$ is "small"

- LSA ile ilgili yansıların hazırlanmasında Melanie Martin'in yansılarından yararlanılmıştır.
- SVD ile Boyut azaltma yansılarının hazırlanmasında Mining of Massive Datasets yansılarından yaralanılmıştır.