Büyük Veri Analitiği
(Big Data Analytics)

M. Ali Akcayol
Gazi Üniversitesi
Bilgisayar Mühendisliği Bölümü

Bu dersin sunumları, "Mining of Massive Datasets, Jure Leskovec, Anand Rajaraman, Jeffrey David Ullman, Stanford University, 2011." kitabı kullanılarak hazırlanmıştır.

Konular

- Uzaklık Ölçütleri
  - Öklit uzaklıkları
  - Jaccard uzaklığı
  - Cosine uzaklığı
  - Edit uzaklığı
  - Hamming uzaklığı
- Locality-Sensitive Fonksiyonların Teorisi
  - Locality-sensitive fonksiyon kümesi
  - Locality-sensitive fonksiyonları
  - Jaccard uzaklığı için locality-sensitive fonksiyon kümesi
  - Hamming uzaklığı için locality-sensitive fonksiyon kümesi
  - Locality-sensitive kümesini iyileştirme
- Locality-Sensitive Hashing - Uygulama

Uzaklık Ölçütleri

- **Jaccard benzerliği,** uzaklık ölçütü olmamasına rağmen **kümelerin ne kadar yakın olduğunu gösterir.**
- **(1 - Jaccard benzerliği) bir uzaklık ölçütüdür.**
- Noktalardan oluşan bir uzayda, x ve y noktası için uzaklık $d(x, y)$ ile gösterilir ve aşağıdaki önermeleri sağlamalıdır:
  - $d(x, y) \geq 0$ (negatif olmaz)
  - $d(x, y) = 0$ (eğer $x = y$ ise)
  - $d(x, y) = d(y, x)$ (uzaklık simetriktir)
  - $d(x, y) \leq d(x, z) + d(z, y)$ (üçgen eşitsizliği)

3

Konular

- Uzaklık Ölçütleri
  - Öklit uzaklıkları
  - Jaccard uzaklığı
  - Cosine uzaklığı
  - Edit uzaklığı
  - Hamming uzaklığı
- Locality-Sensitive Fonksiyonların Teorisi
  - Locality-sensitive fonksiyon kümesi
  - Locality-sensitive fonksiyonları
  - Jaccard uzaklığı için locality-sensitive fonksiyon kümesi
  - Hamming uzaklığı için locality-sensitive fonksiyon kümesi
  - Locality-sensitive kümesini iyileştirme
- Locality-Sensitive Hashing - Uygulama

4

Öklit uzaklıkları

- Öklit uzaklıkları, **en yaygın kullanılan uzaklık ölçütüdür.**
- **N boyutlu öklit uzayında bir nokta** reel sayılardan oluşan **n elemanlı bir vektördür.**
- Bu uzaydaki **L_2-norm** uzaklık aşağıdaki gibidir:

$$d([x_1, x_2, \ldots, x_n],\ [y_1, y_2, \ldots, y_n]) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

- Öklit uzaklıkları **negatif olamaz, 0 ise $X = Y$ 'dir, simetriktir** $(x - y)^2=(y - x)^2$ ve **üçgen eşitsizliğini sağlar.**
- **L_r-norm** uzaklık aşağıdaki gibidir:

$$d([x_1, x_2, \ldots, x_n],\ [y_1, y_2, \ldots, y_n]) = (\sum_{i=1}^{n}|x_i - y_i|^r)^{1/r}$$

Öklit uzaklıkları

- **L_1-norm** uzaklık **Manhattan uzaklığı** olarak adlandırılır.

$$d([x_1, x_2,\ldots x_n],[y_1, y_2,\ldots y_n]) = \left(\sum_{i=1}^{n}|x_i - y_i|\right)$$

- **L_∞-norm** ise r sonsuza giderken limiti gösterir.

$$d([x_1, x_2,\ldots x_n],[y_1, y_2,\ldots y_n]) = \lim_{r\to\infty}\left(\sum_{i=1}^{n}|x_i - y_i|^r\right)^{1/r} = \max_{i=1}^{n}(|x_i - y_i|)$$

- **Örnek:** x=(2, 7) ve y=(6, 4) noktaları için aşağıdaki uzaklıklar hesaplanır.

$$L_1 - norm = |2-6| + |7-4| = 7$$

$$L_2 - norm = \sqrt{(2-6)^2 + (7-4)^2} = 5$$

$$L_\infty - norm = \max(|2-6| + |7-4|) = 4$$

Konular

- Uzaklık Ölçütleri
  - Öklit uzaklıkları
  - Jaccard uzaklığı
  - Cosine uzaklığı
  - Edit uzaklığı
  - Hamming uzaklığı
- Locality-Sensitive Fonksiyonların Teorisi
  - Locality-sensitive fonksiyon kümesi
  - Locality-sensitive fonksiyonları
  - Jaccard uzaklığı için locality-sensitive fonksiyon kümesi
  - Hamming uzaklığı için locality-sensitive fonksiyon kümesi
  - Locality-sensitive kümesini iyileştirme
- Locality-Sensitive Hashing - Uygulama

7

Jaccard uzaklığı

- Jaccard uzaklığı, **Jaccard benzerliği ile hesaplanır.**

$$d(x, y) = 1 - \mathrm{SIM}(x, y)$$

- **Jaccard uzaklığı negatif olamaz.**

  Kesişim kümesi, 0'dan küçük olamaz ve birleşim kümesinden büyük olamaz.

- **$d(x, y) = 0$ ise, $x = y$ 'dir.** ($\mathrm{SIM}(x, y) = 1$).

- **$d(x, y) = d(y, x)$ 'dir.** Kesişim ve birleşim kümeleri simetriktir.

  $x \cup y = y \cup x$ ve $x \cap y = y \cap x$ dir.

- **$d(x, y) \leq d(x, z) + d(z, y)$ sağlanır.** Jaccard uzaklığı $d(x, y)$, minhash fonksiyonunun x ve y için aynı değeri (bucket) döndürmeme **($\mathrm{SIM}(x, y)$ döndürme)** olasılığıdır.

  $h(x) \neq h(y)$ olasılığı; $h(x) \neq h(z)$ olasılığı ile $h(z) \neq h(y)$ olasılığının toplamından büyük olamaz.

  $h(x) \neq h(y)$ ise, $h(x)$ ve $h(y)$'den en az bir tanesi $h(z)$ 'den farklı olmak zorundadır.

8

Konular

- Uzaklık Ölçütleri
  - Öklit uzaklıkları
  - Jaccard uzaklığı
  - Cosine uzaklığı
  - Edit uzaklığı
  - Hamming uzaklığı
- Locality-Sensitive Fonksiyonların Teorisi
  - Locality-sensitive fonksiyon kümesi
  - Locality-sensitive fonksiyonları
  - Jaccard uzaklığı için locality-sensitive fonksiyon kümesi
  - Hamming uzaklığı için locality-sensitive fonksiyon kümesi
  - Locality-sensitive kümesini iyileştirme
- Locality-Sensitive Hashing - Uygulama

Cosine uzaklığı

- Cosine uzaklığı, **vektör elemanlarını integer veya boolean değerler olarak alır.**
- **N boyutlu uzayda noktalar bir yönü gösterir.**
- İki nokta arasındaki **cosine uzaklığı vektörler arasındaki açıyı** (0-180° arasında) **ifade eder.**

$$\cos([x_1, x_2, \ldots x_n], [y_1, y_2, \ldots y_n]) = \frac{X.Y}{\|X\|\|Y\|} = \frac{\sum_{i=1}^{n} x_i.y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

- **Örnek:** x=[1, 2, -1] ve y=[2, 1, 1]

$$\cos(x, y) = \frac{1.2 + 2.1 + (-1).1}{\sqrt{1^2 + 2^2 + (-1)^2}\sqrt{2^2 + 1^2 + 1^2}} = \frac{3}{\sqrt{6}\sqrt{6}} = 0{,}5$$

$$\cos(60°) = 0{,}5$$

Konular

- Uzaklık Ölçütleri
  - Öklit uzaklıkları
  - Jaccard uzaklığı
  - Cosine uzaklığı
  - Edit uzaklığı
  - Hamming uzaklığı
- Locality-Sensitive Fonksiyonların Teorisi
  - Locality-sensitive fonksiyon kümesi
  - Locality-sensitive fonksiyonları
  - Jaccard uzaklığı için locality-sensitive fonksiyon kümesi
  - Hamming uzaklığı için locality-sensitive fonksiyon kümesi
  - Locality-sensitive kümesini iyileştirme
- Locality-Sensitive Hashing - Uygulama

11

Edit uzaklığı

- Edit uzaklığı, **vektörleri string olarak alır.**
- $x = x_1x_2...x_n$ ile $y = y_1y_2...y_m$ noktaları için edit uzaklığı, x 'in y 'ye dönüştürülmesi için minimum insert ve delete (tek karakter) işlem sayısını gösterir.
- **Örnek:** x=abcde ve y=acfdeg
  - Delete b
  - Insert f, c den sonra
  - Insert g, e den sonra

  $d(x, y) = 3$ 'tür.

12

Edit uzaklığı

- Edit uzaklığı, **LCS (Longest Common Subsequence)** ile hesaplanabilir.
- İki string için LCS, **en uzun ortak subsequence**'dir.
- İki string'ten karakter silinerek elde edilir ve karakter sırası iki string'te de aynı olmak zorundadır.

$$d(x, y) = \text{length}(x) + \text{length}(y) - 2*\text{length}(LCS)$$

- **Örnek:** x=abcde ve y=acfdeg
  - LCS(x, y) = acde
  - $d(x, y)$ = length(x) + length(y) – 2*length(LCS) = 5 + 6 – 8 = 3
- x=aba ve y=bab
  - LCS(x, y) = ab veya ba
  - $d(x, y)$ = 3 + 3 – 4 = 2

13

Konular

- Uzaklık Ölçütleri
  - Öklit uzaklıkları
  - Jaccard uzaklığı
  - Cosine uzaklığı
  - Edit uzaklığı
  - Hamming uzaklığı
- Locality-Sensitive Fonksiyonların Teorisi
  - Locality-sensitive fonksiyon kümesi
  - Locality-sensitive fonksiyonları
  - Jaccard uzaklığı için locality-sensitive fonksiyon kümesi
  - Hamming uzaklığı için locality-sensitive fonksiyon kümesi
  - Locality-sensitive kümesini iyileştirme
- Locality-Sensitive Hashing - Uygulama

14

Hamming uzaklığı

- Hamming uzaklığı, **iki vektör için aynı konumdaki farklı eleman sayısıdır.**
- Hamming uzaklığı genellikle **iki vektör boolean değerlere sahipse kullanılır.**
- **Örnek:** x=10101 ve y=11110

  $d(x, y) = 3$

15

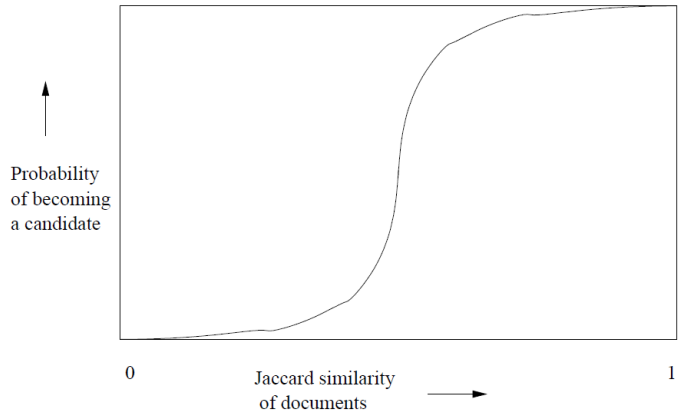
Konular

- Uzaklık Ölçütleri
  - Öklit uzaklıkları
  - Jaccard uzaklığı
  - Cosine uzaklığı
  - Edit uzaklığı
  - Hamming uzaklığı
- Locality-Sensitive Fonksiyonların Teorisi
  - Locality-sensitive fonksiyon kümesi
  - Locality-sensitive fonksiyonları
  - Jaccard uzaklığı için locality-sensitive fonksiyon kümesi
  - Hamming uzaklığı için locality-sensitive fonksiyon kümesi
  - Locality-sensitive kümesini iyileştirme
- Locality-Sensitive Hashing - Uygulama

16

Locality-Sensitive Fonksiyonların Teorisi

- **LS fonksiyonları** (örn. minhash fonksiyonları), **uzaklık değeri küçük olan çiftleri kuvvetli aday çift olarak belirleyebilmektedir.**
- S-eğrisindeki diklik arttıkça, false positive ve false negative çiftlerin sayısı azalır.



17

Konular

- Uzaklık Ölçütleri
  - Öklit uzaklıkları
  - Jaccard uzaklığı
  - Cosine uzaklığı
  - Edit uzaklığı
  - Hamming uzaklığı
- Locality-Sensitive Fonksiyonların Teorisi
  - Locality-sensitive fonksiyon kümesi
  - Locality-sensitive fonksiyonları
  - Jaccard uzaklığı için locality-sensitive fonksiyon kümesi
  - Hamming uzaklığı için locality-sensitive fonksiyon kümesi
  - Locality-sensitive kümesini iyileştirme
- Locality-Sensitive Hashing - Uygulama

18

Locality-sensitive fonksiyon kümesi

- **LS fonksiyonları,** Jaccard uzaklığı, Hamming uzaklığı veya diğer uzaklıklara uygulanabilir.
- LS fonksiyon kümeleri aşağıdaki şartları sağlamalıdır:
  - Birbirine **yakın çiftleri** uzak çiftlere göre **daha çok aday çift olarak belirleyebilmelidirler.**
  - Fonksiyonlar birbirinden **bağımsız olmalıdırlar** ve bağımsız olaylar için **cevap olasılıkları tahmin edilebilmelidir.**
  - **Aday çiftleri**, tüm çiftlere (tüm verilerine) bakma süresine göre **çok daha kısa sürede belirleyebilmelidirler.**
  - Birbirleriyle **birleştirilebilir olmalıdırlar.** Böylelikle **daha düşük false positive** ve **false negative elde edilebilir.**

19

Konular

- Uzaklık Ölçütleri
  - Öklit uzaklıkları
  - Jaccard uzaklığı
  - Cosine uzaklığı
  - Edit uzaklığı
  - Hamming uzaklığı
- Locality-Sensitive Fonksiyonların Teorisi
  - Locality-sensitive fonksiyon kümesi
  - Locality-sensitive fonksiyonları
  - Jaccard uzaklığı için locality-sensitive fonksiyon kümesi
  - Hamming uzaklığı için locality-sensitive fonksiyon kümesi
  - Locality-sensitive kümesini iyileştirme
- Locality-Sensitive Hashing - Uygulama

20

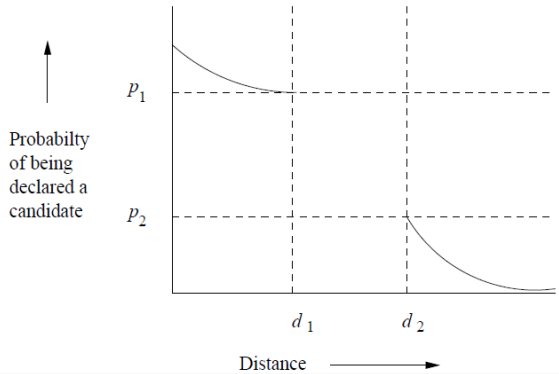
Locality-sensitive fonksiyonları

- **LS fonksiyonları iki çifti (dokümanı) giriş olarak alır ve aday çift olup olmadıklarına karar verir.**
- LS fonksiyonları, **girişlerin hash değerini hesaplar** ve **sonucun eşit olup olmadığına göre bir karar verir.**
- En kolay yöntemde, $f(x) = f(y)$ ise x ve y **aday çifttir**, $f(x) \neq f(y)$ ise x ve y **aday çift değildir.**
- **Bu** şekilde oluşturulan **fonksiyonlar LS fonksiyon kümesi olarak adlandırılır.**
- Karakteristik matris için oluşturulan **minhash fonksiyonları, LS fonksiyon kümesini oluşturur.**

21

Locality-sensitive fonksiyonları

- Bir uzaklık ölçütü d **için,** $d_1 < d_2$ **olmak üzere iki uzaklık** olsun.
- Eğer bir $\mathbf{F}$ fonksiyon kümesindeki tüm f **fonksiyonları aşağıdaki şartları sağlıyorsa** (d_1, d_2, p_1, p_2)***-sensitive*** olarak adlandırılır:
  - **Eğer** $d(x, y) \leq d_1$ **ise,** $f(x) = f(y)$ olma olasılığı **en az** p_1 **kadardır.**
  - **Eğer** $d(x, y) \geq d_2$ **ise,** $f(x) = f(y)$ olma olasılığı **en çok** p_2 **kadardır.**



22

Konular

- Uzaklık Ölçütleri
  - Öklit uzaklıkları
  - Jaccard uzaklığı
  - Cosine uzaklığı
  - Edit uzaklığı
  - Hamming uzaklığı
- Locality-Sensitive Fonksiyonların Teorisi
  - Locality-sensitive fonksiyon kümesi
  - Locality-sensitive fonksiyonları
  - Jaccard uzaklığı için locality-sensitive fonksiyon kümesi
  - Hamming uzaklığı için locality-sensitive fonksiyon kümesi
  - Locality-sensitive kümesini iyileştirme
- Locality-Sensitive Hashing - Uygulama

23

Jaccard uzaklığı için locality-sensitive fonksiyon kümesi

- **Jaccard uzaklığı için F fonksiyon kümesi**, herhangi iki d_1 ve d_2 uzaklıkları için $(d_1, d_2, 1 - d_1, 1 - d_2)$-***sensitive*** kümesi şeklinde tanımlanır.
- Burada, $0 \leq d_1 \leq d_2 \leq 1$ dir.
- Eğer Jaccard uzaklığı $d(x, y) \leq d_1$ **ise,**
  $1 - SIM(x, y) \leq d_1 \Rightarrow SIM(x, y) \geq 1 - d_1$ 'dir.
- Eğer Jaccard uzaklığı $d(x, y) \geq d_2$ **ise,**
  $1 - SIM(x, y) \geq d_2 \Rightarrow SIM(x, y) \leq 1 - d_2$ 'dir.

24

Jaccard uzaklığı için locality-sensitive fonksiyon kümesi

Örnek:

- $d_1 = 0.3$ ve $d_2 = 0.6$ olsun.
- Minhash fonksiyon kümesi **(0.3, 0.6, 0.7, 0.4)-*sensitive*** olarak tanımlanabilir.
- Jaccard uzaklığı **en çok 0.3** olan x ve y için, minhash fonksiyonu **en az 0.7 ($SIM(x, y) \geq 0.7$) olasılıkla aynı değeri (aynı bucket'a eşleştirir) üretir.**
- Jaccard uzaklığı **en az 0.6** olan x ve y için, minhash fonksiyonu **en çok 0.4 ($SIM(x, y) \leq 0.4$) olasılıkla aynı değeri (aynı bucket'a eşleştirir) üretir.**

25

Konular

- Uzaklık Ölçütleri
  - Öklit uzaklıkları
  - Jaccard uzaklığı
  - Cosine uzaklığı
  - Edit uzaklığı
  - Hamming uzaklığı
- Locality-Sensitive Fonksiyonların Teorisi
  - Locality-sensitive fonksiyon kümesi
  - Locality-sensitive fonksiyonları
  - Jaccard uzaklığı için locality-sensitive fonksiyon kümesi
  - Hamming uzaklığı için locality-sensitive fonksiyon kümesi
  - Locality-sensitive kümesini iyileştirme
- Locality-Sensitive Hashing - Uygulama

26

Hamming uzaklığı için locality-sensitive fonksiyon kümesi

- ***d*-boyutlu** x ve y **vektörleri için** $h(x, y)$ **hamming uzaklığını göstersin.**
- Herhangi bir ***i*.konumda** x **ve** y **eşit ise,** $f_i(x) = f_i(y)$ **olsun.**
- **Rastgele seçilen** herhangi bir i **için,** $f_i(x) = f_i(y)$ **olma olasılığı** $1 - (h(x, y)/d)$ şeklinde ifade edilir.
- **Hamming uzaklığı için F fonksiyon kümesi,** herhangi iki d_1 ve d_2 için $(d_1, d_2, 1 - d_1/d, 1 - d_2/d)$-*sensitive* kümesi şeklinde tanımlanır.
- Burada, $d_1 < d_2$'dir.
- Jaccard uzaklığı **0-1** arasındadır, Hamming uzaklığı **0-*d*** arasındadır. Bu yüzden d ile bölünerek ölçeklendirilmesi gerekir.
- Minhash fonksiyonları için $\mathbf{F}$ kümesinde sınırsız fonksiyon olabilir, ancak Hamming uzaklığı için d tanedir.

27

Konular

- Uzaklık Ölçütleri
  - Öklit uzaklıkları
  - Jaccard uzaklığı
  - Cosine uzaklığı
  - Edit uzaklığı
  - Hamming uzaklığı
- Locality-Sensitive Fonksiyonların Teorisi
  - Locality-sensitive fonksiyon kümesi
  - Locality-sensitive fonksiyonları
  - Jaccard uzaklığı için locality-sensitive fonksiyon kümesi
  - Hamming uzaklığı için locality-sensitive fonksiyon kümesi
  - Locality-sensitive kümesini iyileştirme
- Locality-Sensitive Hashing - Uygulama

28

Locality-sensitive kümesini iyileştirme

- Bir $\mathbf{F}$ kümesi (d_1, d_2, p_1, p_2)-*sensitive* olsun.
- $\mathbf{F}$ kümesinden bir $\mathbf{F'}$ kümesi *AND-construction* ile oluşturulabilir.
- $\mathbf{F'}$ kümesindeki her fonksiyon $\mathbf{F}$ kümesinden r tane fonksiyon kullanılarak (birleştirilerek) oluşturulur **(r-way construction)**.
- Her $f \in \mathbf{F'}$ fonksiyonu, $\{f_1, f_2, \ldots, f_r\} \in \mathbf{F}$ kümesinden oluşturulur.
- Sadece ve sadece tüm $i = 1, 2, \ldots, r$ için $f_i(x) = f_i(y)$ $(f_i \in \mathbf{F})$ olması durumunda, $f(x) = f(y)$ $(f \in \mathbf{F'})$ denir.
- Elde edilen $\mathbf{F'}$ kümesi $(d_1, d_2, (p_1)^r, (p_2)^r)$-*sensitive* olur.

29

Locality-sensitive kümesini iyileştirme

- $\mathbf{F}$ kümesinden bir $\mathbf{F'}$ kümesi *OR-construction* ile oluşturulabilir.
- $\mathbf{F'}$ kümesi $\mathbf{F}$ kümesinden r tane fonksiyon kullanılarak oluşturulur.
- Her $f \in \mathbf{F'}$ fonksiyonu, $\{f_1, f_2, \ldots, f_r\} \in \mathbf{F}$ kümesinden oluşturulur.
- Sadece ve sadece bir veya daha fazla $i = 1, 2, \ldots, r$ için $f_i(x) = f_i(y)$ $(f_i \in \mathbf{F})$ olması durumunda, $f(x) = f(y)$ $(f \in \mathbf{F'})$ denir.
- Elde edilen $\mathbf{F'}$ kümesi $(d_1, d_2, 1-(1-p_1)^r, 1-(1-p_2)^r)$-*sensitive* olur.
- $\mathbf{F}$ kümesindeki bir fonksiyonun x ve y'yi aday çift yapma olasılığı p ise, aday yapmama olasılığı $1 - p$ 'dir.
- r tane fonksiyonun aday yapmama olasılığı $(1 - p)^r$ 'dir.
- En az bir tane f_i fonksiyonunun aday çift yapma olasılığı $1 - (1 - p)^r$ şeklinde hesaplanır.

30

Locality-sensitive kümesini iyileştirme

Örnek - 1

- $\mathbf{F}_1$ kümesi $\mathbf{F}$ kümesinden 4-way *AND-construction* ile üretilsin. $\mathbf{F}_2$ kümesi de $\mathbf{F}_1$ kümesinden 4-way *OR-construction* ile üretilsin.
- 4-way AND fonksiyonları, $\boldsymbol{p}$ olasılıklarını $\boldsymbol{p^4}$ yapar.
- Ardından uygulanan 4-way OR fonksiyonları, $\boldsymbol{p^4}$ olasılıklarını $\mathbf{1 - (1 - p^4)^4}$ yapar ve tablodaki yeni olasılık değerleri bulunur.
- $\mathbf{F}$ minhash fonksiyonları olsun. $\mathbf{F}$ kümesi **(0.2, 0.6, 0.8, 0.4)-sensitive** ise, $\mathbf{F}_2$ kümesi **(0.2, 0.6, 0.8785, 0.0985)-sensitive** olur.
- $\mathbf{F}$ kümesi yerine $\mathbf{F}_2$ kümesi kullanıldığında **FN (0.8785)** sayısı **azalmıştır,** **FP (0.0985)** sayısı **azalmıştır.**

p	$1-(1-p^4)^4$
0.2	0.0064
0.3	0.0320
0.4	0.0985
0.5	0.2275
0.6	0.4260
0.7	0.6666
0.8	0.8785
0.9	0.9860

31

Locality-sensitive kümesini iyileştirme

Örnek - 2

- $\mathbf{F}$ kümesine önce 4-way *OR-construction*, ardından 4-way *AND-construction* yapılsın.
- 4-way OR fonksiyonları, $\boldsymbol{p}$ olasılıklarını $\mathbf{1 - (1 - p)^4}$ yapar.
- Ardından uygulanan 4-way AND fonksiyonları, $\mathbf{1 - (1 - p)^4}$ olasılıklarını $\mathbf{(1 - (1 - p)^4)^4}$ yapar ve tablodaki yeni olasılık değerleri bulunur.
- $\mathbf{F}$ kümesi **(0.2, 0.6, 0.8, 0.4)-sensitive** ise, $\mathbf{F}_2$ kümesi **(0.2, 0.6, 0.9936, 0.5740)-sensitive** olur.
- Yüksek olasılık 1'e yaklaşmıştır. Düşük olasılık yükselmiştir.
- $\mathbf{F}$ kümesi yerine $\mathbf{F}_2$ kümesi kullanıldığında **FN (0.9936)** sayısı **azalmıştır,** **FP (0.5740)** sayısı **artmıştır.**

p	$\left(1-(1-p)^4\right)^4$
0.1	0.0140
0.2	0.1215
0.3	0.3334
0.4	0.5740
0.5	0.7725
0.6	0.9015
0.7	0.9680
0.8	0.9936

Konular

- Uzaklık Ölçütleri
  - Öklit uzaklıkları
  - Jaccard uzaklığı
  - Cosine uzaklığı
  - Edit uzaklığı
  - Hamming uzaklığı
- Locality-Sensitive Fonksiyonların Teorisi
  - Locality-sensitive fonksiyon kümesi
  - Locality-sensitive fonksiyonları
  - Jaccard uzaklığı için locality-sensitive fonksiyon kümesi
  - Hamming uzaklığı için locality-sensitive fonksiyon kümesi
  - Locality-sensitive kümesini iyileştirme
- Locality-Sensitive Hashing - Uygulama

33

Locality-Sensitive Hashing - Uygulama

- Parmak izi eşleştirmede **anormal değişimlere** (yüksekliklerin sonlanması, birleşmesi, ayrılması, …) **bakılır.**
- Parmak izindeki **değişimler bulunduğu konuma bağlı olarak bir grid ile gösterilebilir.**
- Parmak izlerini gösteren kümedeki gridlerden oluşan **elemanlar Jaccard uzaklığı veya Jaccard benzerliği ile karşılaştırılabilir.**
- Parmak izi karşılaştırmada **iki amaç olabilir:**
  - **Bir parmak izi** ile (örn. silah üzerinde bulunan) veritabanındaki tüm parmak izleri kaşılaştırılabilir **(many-one problemi).**
  - **Tüm veritabanındaki parmak izleri** içinde birbirine benzeyenler bulunabilir **(many-many problemi).**

34

Locality-Sensitive Hashing - Uygulama

- Rastgele seçilen herhangi **bir parmak izine ait grid içerisinde rastgele seçilen bir hücrede anormal değişim** bulma **olasılığı %20 olsun.**
- **Aynı parmağa** ait **iki parmak izinden birisinde değişim olan bir hücre için** diğerine ait gridin **aynı hücresinde de anormal değişim olma olasılığı %80 olsun.**
- **Locality-sensitive fonksiyonlar** kümesi $\mathbf{F}$ 'deki her bir f fonksiyonu, **üç grid hücresi** belirlenerek tanımlanabilir.
- f fonksiyonu **her iki parmak izinde de 3 grid hücresinde** anormal **değişim varsa "EVET"** üretir, **aksi durumda "HAYIR"** üretir.

35

Locality-Sensitive Hashing - Uygulama

- **Many-one probleminin çözümü için** $\mathbf{F}$ 'deki **fonksiyonlar kullanılarak** veritabanındaki parmak izlerinin ait olduğu **bucket'lar hesaplanır.**
- Girilen **yeni parmak izinin ait olduğu bucket hesaplanır** ve **bu bucket'taki tüm parmak izleriyle karşılaştırılır.**
- **Many-many problemi için** tüm **bucket'lardaki parmak izleri kendi aralarında ikili olarak karşılaştırılır.**

36

Locality-Sensitive Hashing - Uygulama

- **F** içerisindeki bir f fonksiyonu ile **farklı parmaklara ait iki parmak izinin aynı bucket'a** atanma olasılığı $(0,2)^6 = 0,000064$ 'tür (6 bağımsız olay).
- **Aynı parmağa ait** birinci parmak izinde **3 hücrenin de anormal değişime sahip olma olasılığı** $(0,2)^3$, **bu gerçekleşirse** ikinci parmak izindeki **üç hücrenin anormal değişime sahip olma olasılığı** $(0,8)^3$ **olur.**
- **F** içerisindeki bir f fonksiyonu ile **aynı parmağa ait iki parmak izinin aynı bucket'a atanma olasılığı** $(0,2)^3.(0,8)^3 = 0,008 \cdot 0,512 = 0,004096$.
- **Aynı parmağa** ait iki parmak izinin **aynı bucket**'a gelme olasılığı **yaklaşık** $\%0,41$ (TP) olur.
- **Farklı parmaklara** ait iki parmak izinin **aynı bucket**'a gelme olasılığı **yaklaşık** $\%0,0064$ (FP) olur.

37

Locality-Sensitive Hashing - Uygulama

- **F** kümesinden 1024-way OR-construction yapılsın.
- **OR-construction** ile herhangi bir f fonksiyonu için aday yapılanlar (en az bir aynı bucket) aday çift kabul edilir.
- OR-construction yapıldığında, (d_1, d_2, p_1, p_2)-*sensitive* kümesi, $(d_1, d_2, 1 - (1 - p_1)^{1024}, 1 - (1 - p_2)^{1024})$-*sensitive* kümesine dönüşür.
- **Aynı parmağa ait iki parmak izinin** en az bir (OR) **aynı bucket'ta yer alma olasılığı** $1 - (1 - 0,004096)^{1024} = 0,985$ **(%98,5)** olur.
  (FN %1,5 – tanıma hatası, kontrol edilmesi gereken ancak kontrol edilmeyen).
- **İki farklı parmağa ait iki parmak izinin aynı bucket'ta yer alma olasılığı** $1 - (1 - 0,000064)^{1024} = 0,063$ **(%6,3)** olur.
  (FP %6,3 – veritabanında gereksiz bakılan oran).

38

Ödev

- Büyük veri içerisinde birbiriyle aynı olan kümelerin (dokümanlar, Web sayfaları, …) bulunması için kullanılan yöntemlere yönelik bir araştırma ödevi hazırlayınız.