

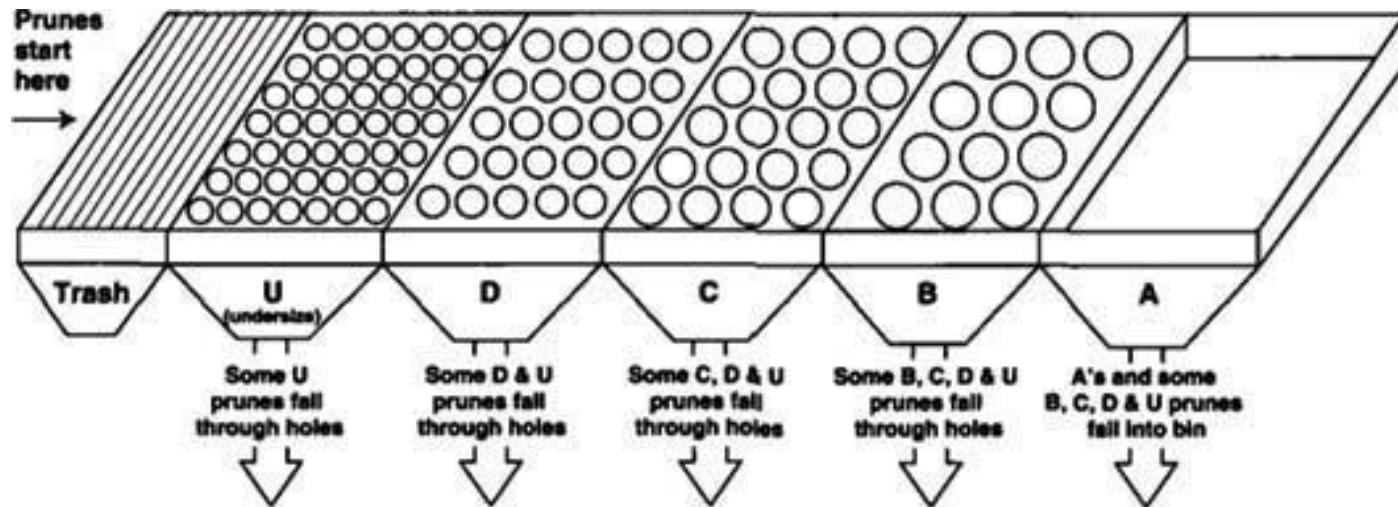
Soft kompjuting

Metrike rastojanja

K-NN klasifikator

Rastojanje – sličnost

- Mašina za sortiranje voća



- Nakon ovog procesa, u svakoj kofi se nalazi voće slične veličine
- U ovom primeru *sličnost* voća je definisana preko njegove veličine (dimenzija koje dozvoljavaju filteri U-A)

Rastojanje – sličnost

- Često je tokom obrade podataka neophodno uspostaviti klase ekvivalencije ili topologiju nad prostorom podataka
 - Klase ekvivalencije → koji objekti su (efektivno) isti
 - Topologija → možemo izmeriti razdaljinu između objekata
- Da bismo ovo uradili, moramo definisati
 - Šta znači jednakost
 - Šta znači razdaljina u prostoru podataka
- Ovo treba da uradimo na osnovu čovečijeg zapažanja, razumevanja i znanja o problemu

Rastojanje – sličnost

Pose variation



$D(x, y) = ?$

Illumination variation



Rastojanje – sličnost

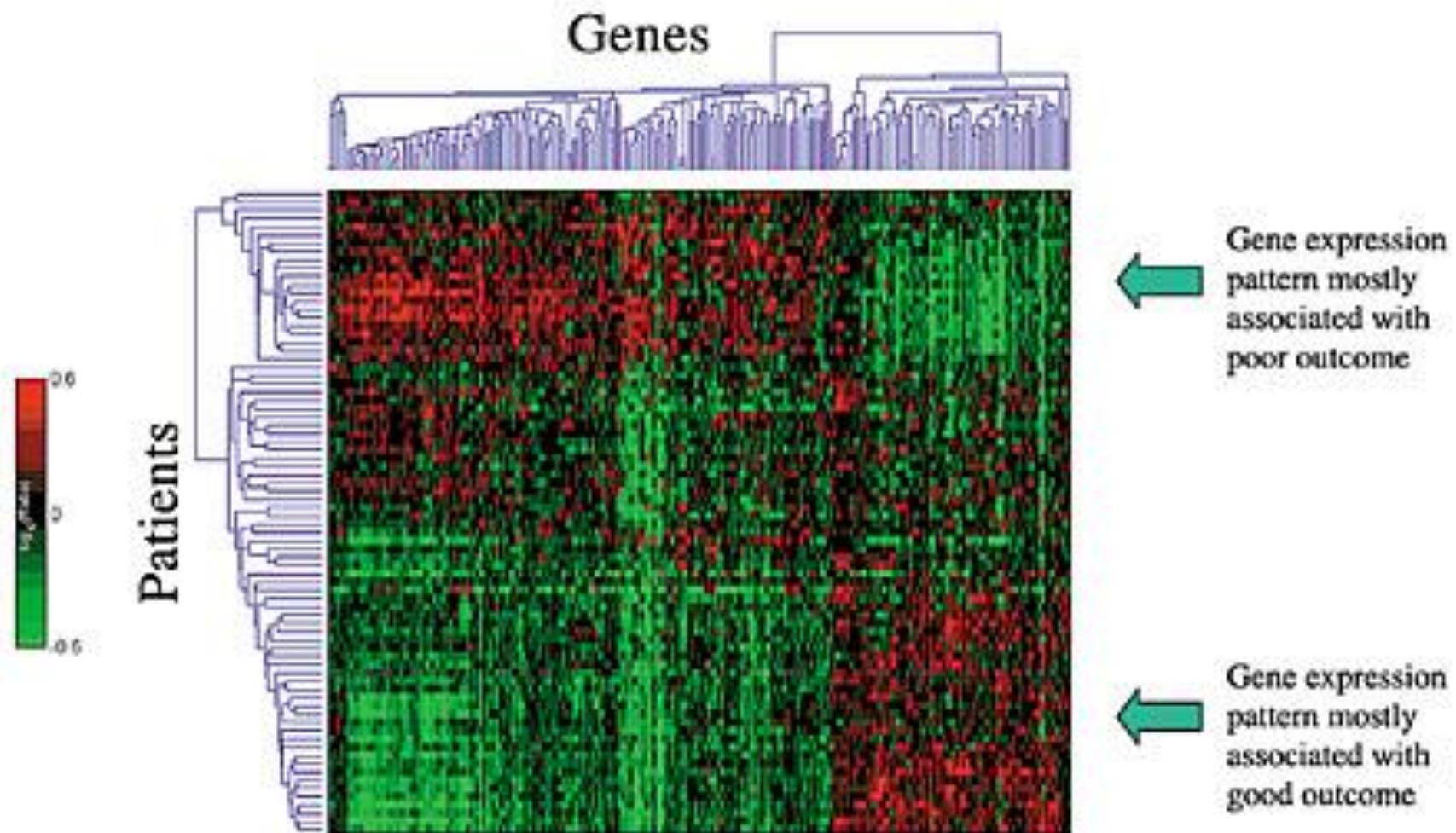


vue.ai
AI FOR FASHION

Tap into the power of visual similarity

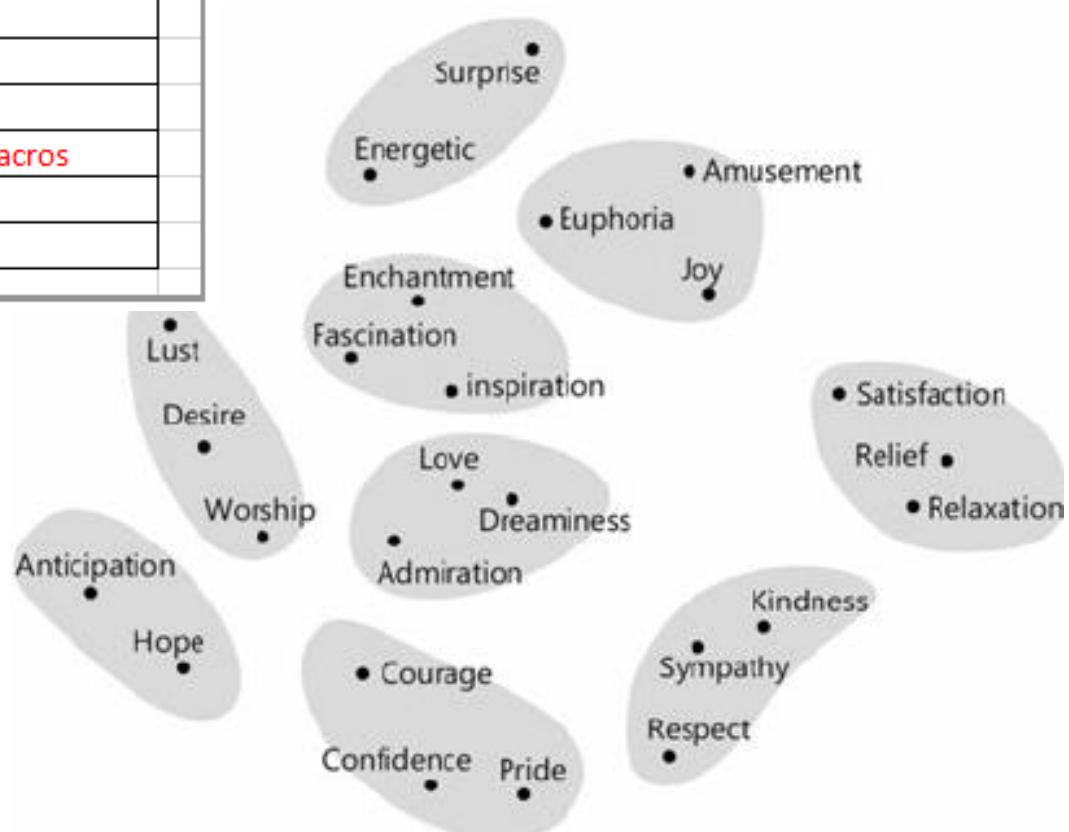
<https://www.apptha.com/blog/visual-recommendation-engine-for-e-commerce/>

Rastojanje – sličnost



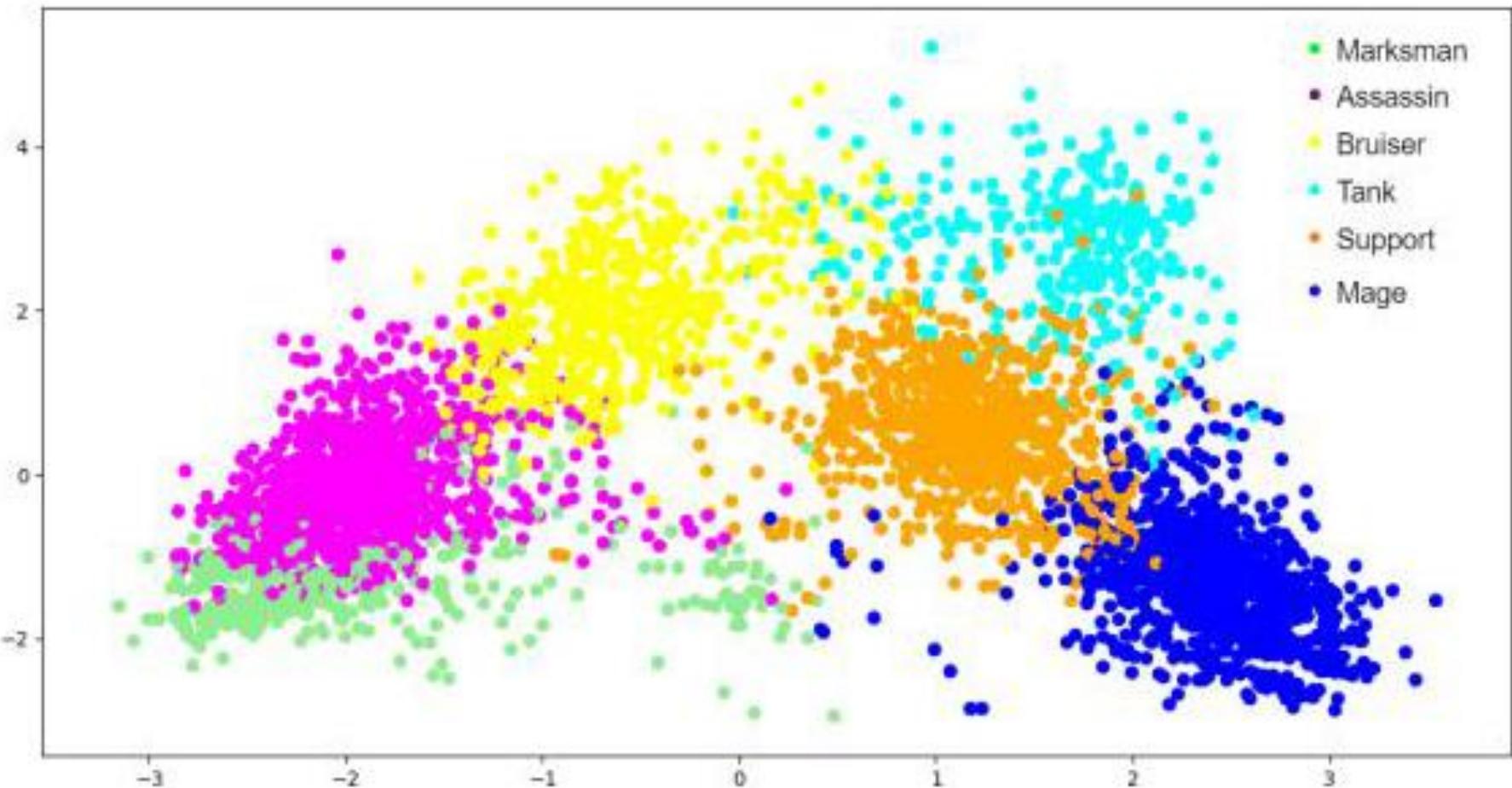
Rastojanje – sličnost

A	B	
1	Column 1	Column 2
2 something	something else	
3 Onemore	Oncemore	
4 another value	another value	
5 this one matches	this one matches	
6 but this doesn't	but this does not	
7 what about me?	what about me!	
8 well, lets try	lets try... Well	
9 but what about logic	but what about logic	
10 Well, we are using VBA	Well, we are using macros	
11 that is right	that is right	
12 last one	last one	
13		

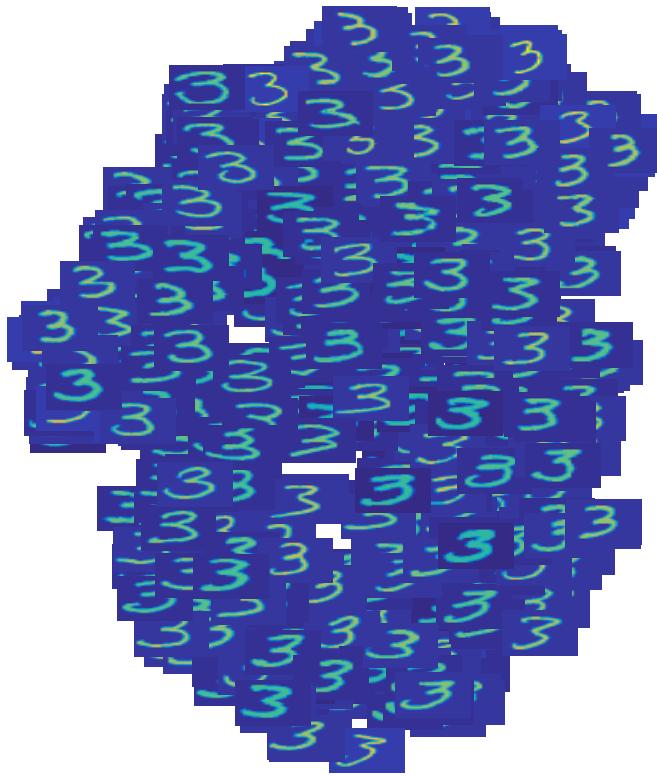


Rastojanje – sličnost

Milan Keča - Klasterovanje heroja u igri *League of Legends* (prema načinu igre)

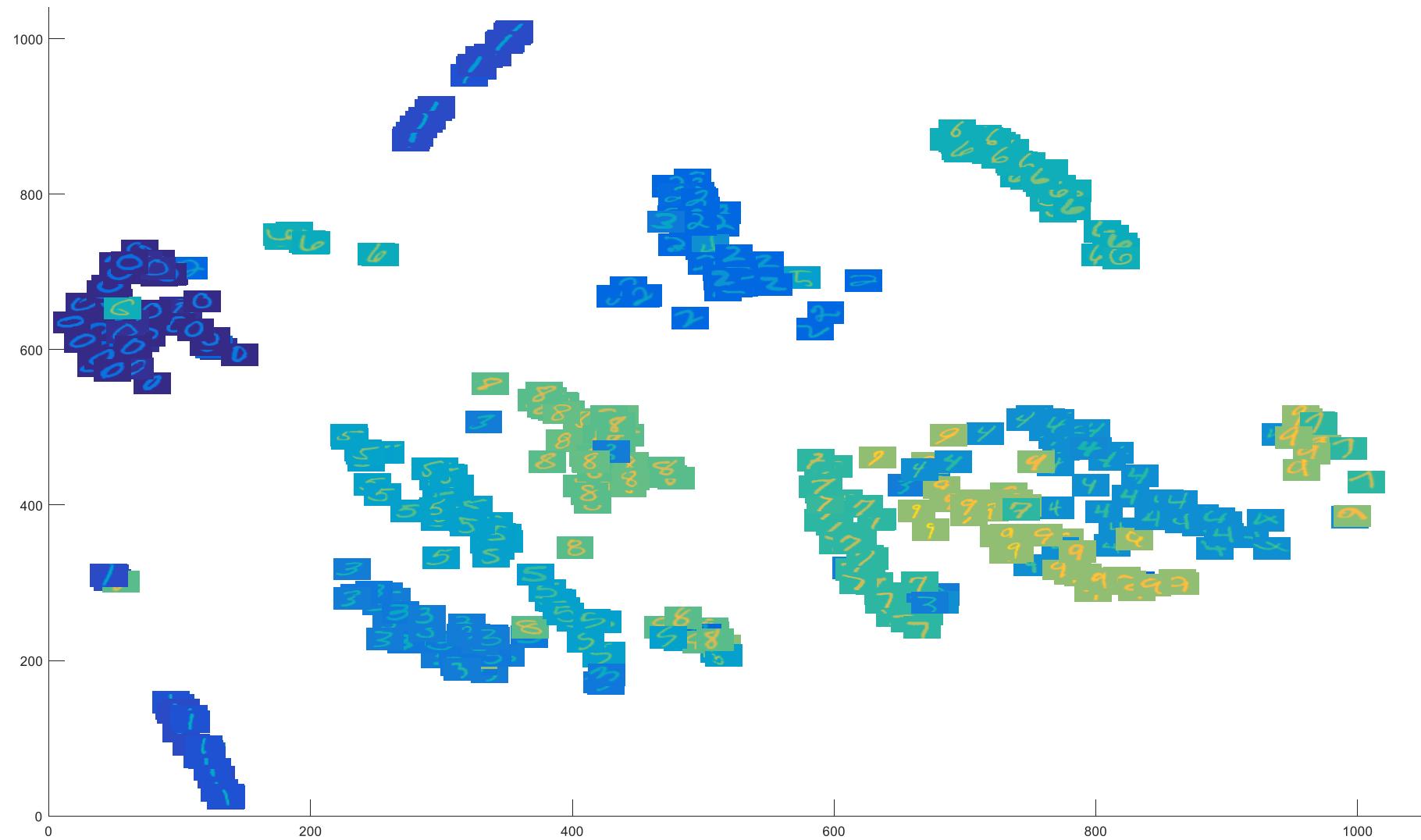


Redukcija dimenzionalnosti



t-SNE – Laurens van der Maaten, MNIST, <http://distill.pub/2016/misread-tsne/>

Redukcija dimenzionalnosti



Redukcija dimenzionalnosti



[t-SNE – Laurens van der Maaten](#), Faces in the wild

Rastojanje – sličnost

- Problem iz numerike: zaokruživanje većih tipova na manje

```
public class JavaApplication1 {                                „Hard“  
    static int i = (int) 1.1;           štampa se true samo ako važi ==  
    static float j = (float) 1.1;  
    public static void main(String [] args){  
        if(i == j){  
            System.out.println("true");}  
        else{  
            System.out.println("false");}}}
```

```
public class JavaApplication1 {                                „Soft“  
    static int i = (int) 1.1;           umesto egzaktnog poređenja koristimo  
    static float j = (float) 1.1;       poređenje sa nekom tolerancijom  
    public static void main(String [] args){  
        if( (i - j )<0.2){  
            System.out.println("true");}  
        else{  
            System.out.println("false");}}}
```

Rastojanje – sličnost

```
String a="milan ide u skolu";
String b="milan ide ";
String[] baza = { "u školu", "u obdanište", "na posao" };
```

„Hard“: samo ako je tačno

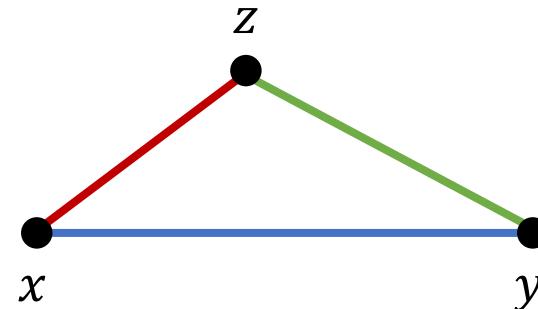
„Soft“: može i približno – gleda se razlika u broju karaktera

```
foreach (string t in baza) {
    string r = b + t;
    if (r == a) {
        Console.WriteLine(r);
        break;
    }
}
```

```
foreach (string t in baza) {
    string r = b + t;
    int d = distance(r, a);
    if (d <= 1) {
        Console.WriteLine(r);
        break;
    }
}
```

Metrika – funkcija rastojanja

- Metrika ili *funkcija rastojanja* definiše udaljenost između svih objekata iz nekog skupa
- Skup sa definisanim metrikom se zove *metrički prostor*
- Metrika d definisana nad skupom X ($d: X \times X \rightarrow \mathbb{R}$) mora zadovoljavati sledeće uslove:
 1. $d(x, y) \geq 0$
 2. $d(x, y) = 0$ akko $x = y$
 3. $d(x, y) = d(y, x)$
 4. $d(x, y) \leq d(x, z) + d(z, y)$



Metrika – funkcija rastojanja

Tip podataka	Reprezentacija	Mera rastojanja
Brojevi	Int, float	<ul style="list-style-type: none">• $(x - y)^2$• $x - y$
Vektori brojeva	Array/Matrix	<ul style="list-style-type: none">• $\sum_{i=1}^N (x_i - y_i)^2$• Kosinusna razlika i druge metode
Funkcije	Parametarski i neparametarski modeli	<ul style="list-style-type: none">• Najznačajnije je posmatranje funkcija kao distribucija verovatnoće i poređenje pomoću <i>KL</i> (<i>Kullback–Leibler</i>) divergencije
Grafovi	Serijalizacija u vektore ili matrice	<ul style="list-style-type: none">• Potrebno je izdvojiti osobine grafova koje se onda mogu numerički porebiti• Ovde je ključni problem kako uspostaviti metrički prostor nad grafovima
Objekti	Klase	<ul style="list-style-type: none">• Na koji način definišemo sličnost između objekata zavisi od toga kakve operacije poređenja možemo uvesti• Treba birati reprezentaciju koja dozvoljava luke i smislene metrike razdaljine• Za ovo je često potrebno domensko znanje o problemu

Poređenje stringova

- Hamming
- Levenshtein

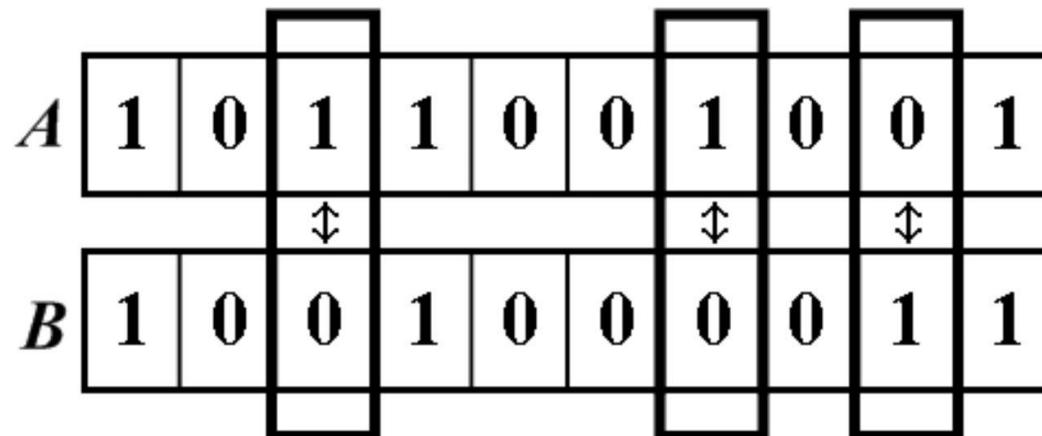
Približno poređenje stringova

- *Hamming* – broj bitova u kojima se stringovi razlikuju
- *Levenshtein (Edit distance)* – minimalan broj operacija neophodan da se jedan string pretvori u drugi
- *Needleman-Wunsch* – poravnavanje sekvenci proteina ili nukleotida
- *Smith-Waterman* – pronalaženje najvećih zajedničkih pod-stringova
- *Soundex* – indeksiranje imena po izgovoru na engleskom jeziku
- *Jaro-Winkler* – efektivno otkrivanje (sličnih) duplikata u automatski prikupljenim podacima

Hamming

- Broj različitih bitova u nizovima iste dužine

Hamming distance = 3 —



```
def hamming_distance(s1, s2):
    assert len(s1) == len(s2)
    return sum(ch1 != ch2 for ch1, ch2 in zip(s1, s2))
```

Levenshtein rastojanje

- Mera razlike između dva niza (slova, brojeva ili objekta)
- Nema ograničenje da dužine poređenih nizova moraju biti iste
- Minimalan broj *operacija* koje treba obaviti da se jedan string transformiše u drugi
- Moguće operacije:
 - Dodavanje (Tehnika → Tehnička)
 - Brisanje (Tehnička → Tehnika)
 - Zamena (Tehnička → Tehnicki)

Levenshtein rastojanje – primer

Cena sve tri operacije (dodavanje/brisanje/zamena) je 1

ubacivanje
izmena

		Z	R	E	N	J	A	N	I	N
	0	1	2	3	4	5	6	7	8	9
Z	1	0	1	2	3	4	5	6	7	8
R	2	1	0	1	2	3	4	5	6	7
E	3	2	1	0	1	2	3	4	5	6
N	4	3	2	1	0	1	2	3	4	5
A	5	4	3	2	1	1	1	2	3	4
N	6	5	4	3	2	2	2	1	2	3
I	7	6	5	4	3	3	3	2	2	3
N	8	7	6	5	4	4	4	3	3	2

Levenshtein rastojanje

```
public int LevenshteinDistance(string s, string t)
{
    int m = s.Length;    int n = t.Length;
    int[,] d = new int[m + 1, n + 1];
    for (int i = 0; i < m + 1; i++)
        for (int j = 0; j < n + 1; j++) d[i, j] = 0;
    for (int i = 1; i <= m; i++)
        d[i, 0] = i;
    for (int j = 1; j < n + 1; j++)
        d[0, j] = j;
```

Inicijalizacija matrice d
Ovo su trivijalna rešenja

Levenshtein rastojanje

```
for (int j = 1; j <= n; j++)  
{  
    for (int i = 1; i <= m; i++)  
    {  
        if (s[i - 1] == t[j - 1])  
        {  
            d[i, j] = d[i - 1, j - 1];  
        }  
        else  
        {  
            d[i, j] = Math.Min(Math.Min(  
                d[i - 1, j] + 1,  
                d[i, j - 1] + 1),  
                d[i - 1, j - 1] + 1);  
        }  
    }  
}  
return d[m, n];  
}
```

Vreme izvršavanja? $O(n*m)$

Ako je isti karakter, onda cena ostaje ista, i krećemo se dijagonalno na dole

Deo rezultata je prethodno izračunat u d

$\text{d}[i - 1, j] + 1$, $\text{d}[i, j - 1] + 1$, $\text{d}[i - 1, j - 1] + 1$;

//BRISANJE //DODAVANJE //ZAMENA

Cena operacija Može biti različita

Kako odrediti cenu operacije?

- Cena operacije može da zavisi od raznih svojstava stringova koji se porede
- Možemo je odrediti na osnovu prethodnog znanja o problemu i podacima koje treba obraditi
- Na primer,
 - Dodavanje samoglasnika: 0.5
 - Dodavanje suglasnika: 1
 - Zamena č u c: 0.2
 - Zamena slova a i p: 2 (različiti krajevi tastature)
 - Zamena slova a i s: 0.8 (bliski na tastaturi – češće se pojavljuje kao greška)
- Za specifične primene kao što je sekvenciranje amino kiselina mogu se ubaciti posebni operatori kao što je transpozicija koja menja položaj dva karaktera AC <-> CA

Levenshtein rastojanje – otvorena pitanja

- Koje operacije su potrebne i kako ih vrednovati?
- Da li je moguće statistički utvrditi najbolje cene?
 - Imamo izvorni tekst i njegovu ispravljenu verziju
 - Onda bismo mogli da optimizujemo cene operacija tako da odgovaraju izmerenim razdaljinama
 - Greškama koje se češće dešavaju bi pridodali manju cenu ispravke
 - Greškama koje su ređe u podacima bi dali veću cenu
- Ali...
 - Čak i kada imamo samo tri operacije (tri parametra – cene operacija), moguće je overfitovati podatke
- *No-free-lunch* teorema: ako je algoritam posebno pogodan za određenu primenu, onda je ekvivalentno nepogodan za neku drugu primenu
 - Što više pilagodjavamo algoritam nekom specifičnom skupu podataka, on će slabije da generalizuje na druge skupove podataka

Levenshtein rastojanje – primeri primene

- Spelling correction
 - Određivanje kandidata za korekciju pogrešno spelovanih reči
 - Biraju se oni kandidati sa minimalnom distancicom od pogrešne reči
- Traženje duplikata u registru ulica

2 oktobra	DR MLADENA STOJANOVIĆA
2. oktobar	DR BRANKA MANOJLOVIĆA
2. Oktobar	Dr Đorđa Joanovića
2. oktobar	DR ĐORĐA JOANOVIĆA
2. oktobar	DR Đorđa Jovanovića
2. oktobar	DR ĐORĐA LAZIĆA
2. Oktobara	DR IVANA RIBARA
2. Oktobara	DR JANKA BULJIKA
2. OKTOBRA	Dr Luke Mardešića
2. oktobra	Dr Mijatovića
2. Oktobra	DR MLADENA STOJANOVIĆA
2. Oktobra	Dr Mladena Stojanovića
2. OKTOBRA	Dr Ribara
2. Oktobra	Dr Ribara
2. Oktobra	DR SVETISLAVA KASAPINOVIĆA
2.Oktobra	Dr Svetislava Kasapinovića
2.Oktobra	DR SVETISLAVA KASAPINOVIĆA

Levenshtein rastojanje – primeri primene

- Praćenje tekstualnih uticaja

Example from
Horton, Olsen, Roe,
Digital Studies / Le
champ
numérique, Vol 2,
No 1 (2010)

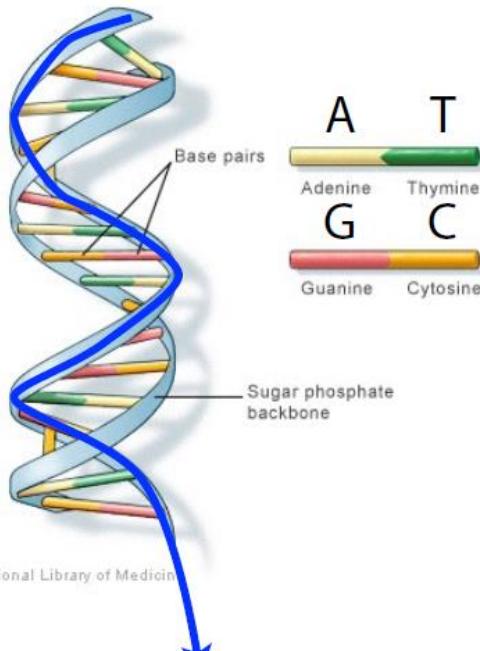
This later play
by Markham
references
Shakespeare's
poem.

Common
passages
identified by
sequence
alignment
algorithms.

She locks her lily fingers one in one. "Fondling," she saith, "since I have hemmed thee here Within the circuit of this ivory pale, I'll be a park, and thou shalt be my deer; Feed where thou wilt, on mountain or in dale: Graze on my lips; and if those hills be dry, Stray lower, where the pleasant fountains lie." Within this limit is relief enough.... (Shakespeare, Venus and Adonis [1593])

Pre. Fondling, said he, since I haue hem'd thee heere,
Within the circuit of this Iuory pale.
Dra. I pray you sir help vs to the speech of your master.
Pre. Ile be a parke, and thou shalt be my Deere: He is very busie in his study. Feed where thou wilt, in mountaine or on dale. Stay a while he will come out anon. Graze on my lips, and when those mounts are drie, Stray lower where the pleasant fountaines lie . Go thy way thou best booke in the world.
Ve. I pray you sir, what booke doe you read? (Markham, The dumbe knight. A historicall comedy... [1608])

Levenshtein rastojanje – primeri primene



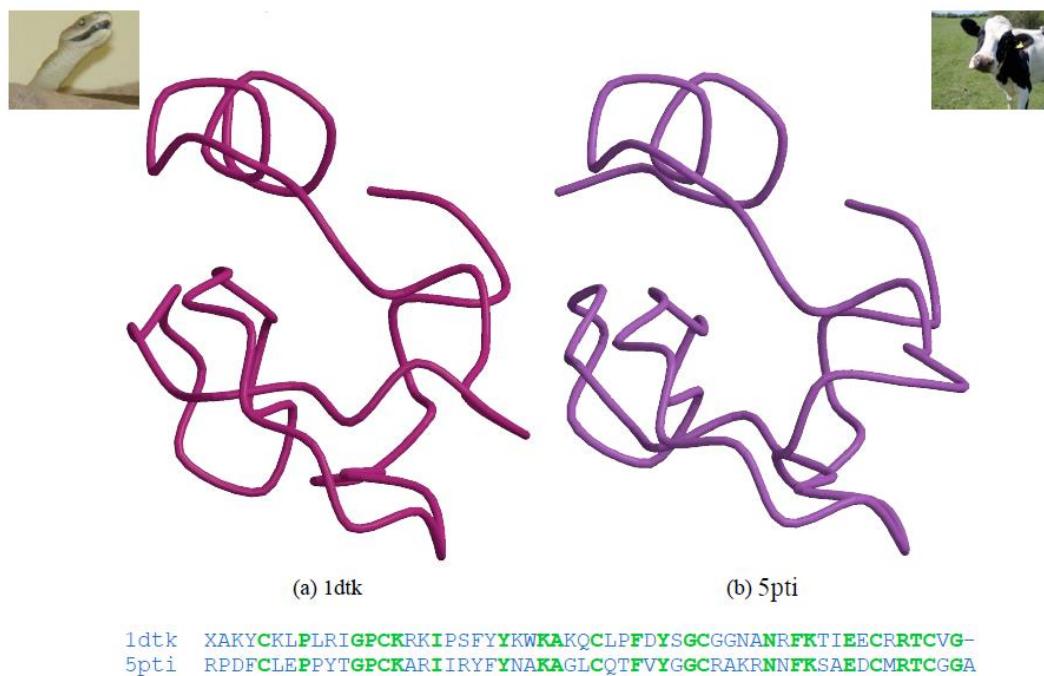
- *Protein sequencing*

- Proteine možemo predstaviti kao stringove sastavljene od slova A, C, G i T

<i>H. sapiens</i>	-EDSSDS-ENAEPDLDDNEDEEPAVEIEPEPE-----PQPVTPA
<i>P. troglodytes</i>	-EDSSDS-ENAEPDLDDNEDEEPAVEIEPEPE-----PQPVTPA
<i>C. lupus</i>	-EDSSDS-ENAEPDLDDNEDEEPAVEIEPEPE-----PQPVTPA
<i>B. taurus</i>	-EDSSDS-ENAEPDLDDNEDEEPAVEIEPEPE-----PQPVTPA
<i>M. musculus</i>	-EDSSDSEENAEAPDLDDNEEEEEPAVEIEPEPE--PQPQPPPPPQPVAPA
<i>R. norvegicus</i>	-EDSSDS-ENAEPDLDDNEEEEEPAVEIEPEPEPQPQQPQQPQPVAPA
<i>G. gallus</i>	-EDSSDSEENAEAPDLDDNEDEETAVEIEAEPE-----VSAEAPA
<i>D. rerio</i>	DDDDDDSD E HGEAPDLDDIDEEDEDDL-LDEDQMGLLDQAPPSVPIP-APA

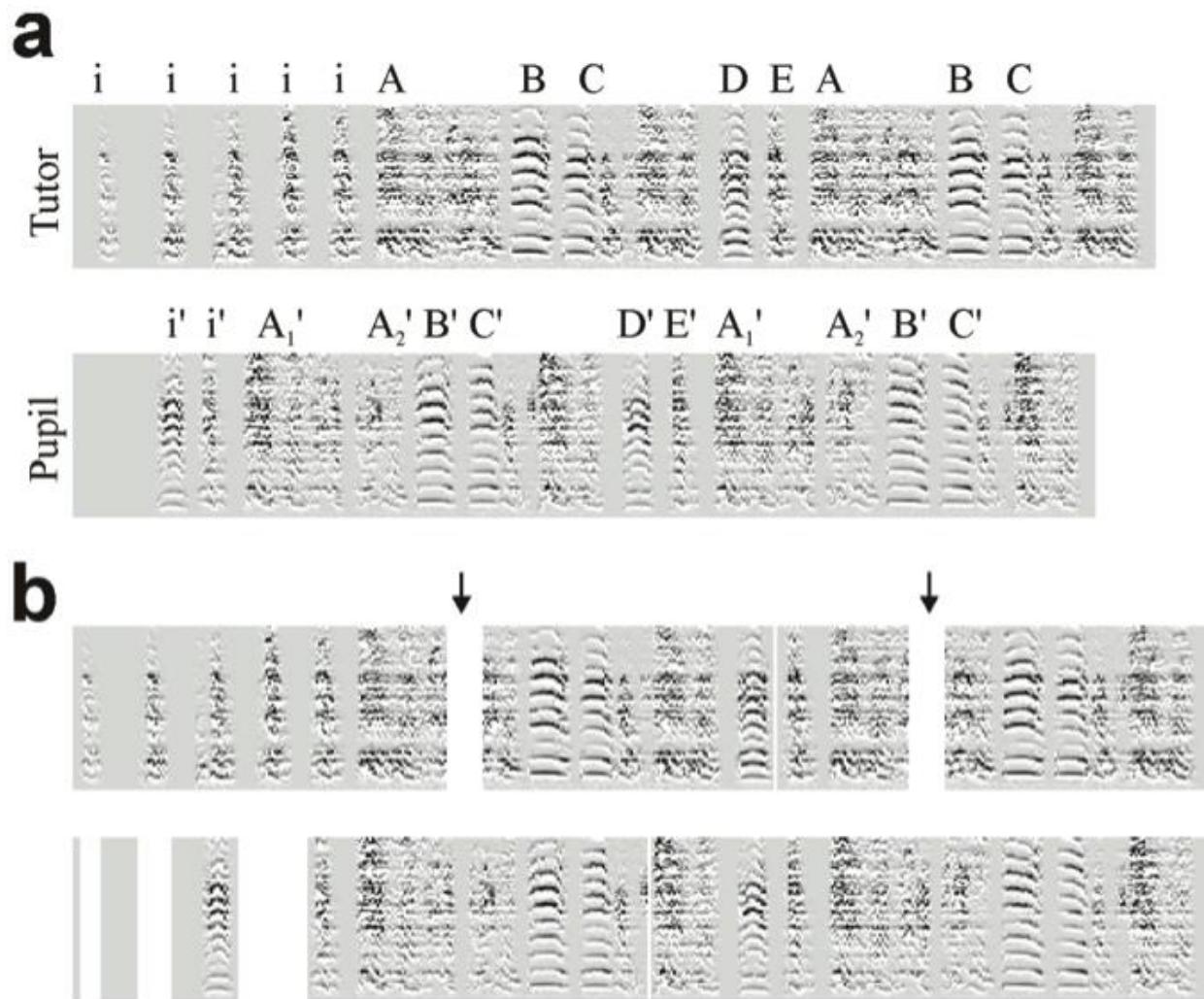
- Zašto poređiti sekvene proteina?
 - Pronaći važne sekvene pronalaženjem očuvanih regija
 - Pronaći gene slične poznatim genima
 - Razumevanje evolutivnih odnosa i udaljenosti
 - Korak u sastavljanju genoma
 - ...

Levenshtein rastojanje – primeri primene



- Primarna struktura diktira oblik proteina, a oblik indukuje njegovu funkciju
- Ako pronađemo da se delovi primarne strukture dva proteina preklapaju, onda je moguće da se i njihove funkcije preklapaju
- U istraživanju poremećaja i bolesti nekada je cilj pronaći sekvene koje se *ne preklapaju*

Poređenje pesama ptica



Metrike rastojanja – otvorena pitanja

Kako definisati metriku u nekom prostoru?

- **Euklidska udaljenost**

- U jednodimenzionom prostoru: $d(x_i, y_i) = \sqrt{(x_i - y_i)^2}$
- U višedimenzionom prostoru možemo definisati mnogo različitih metrika udaljenosti
 - Možemo sve dimenziije tretirati jednak (obavezna normalizacija!)
 - Možemo odlučiti da nekima dodelimo veću težinu, npr. poredimo dva naučna rada i neka obeležja su nam relevantnija za problem



- Kako dizajniramo/selektujemo obeležja je **veoma važno**, ali i **veoma teško** – zahteva domensko znanje

Euklidska udaljenost

Originalni dokumenti:



1	0	0	0	5	3	0	0	1	0	0	0
3	1	0	0	2	0	0	1	0	1	0	0

$$similarity = 5.1$$

Isti dokumenti, pri čemu je svakom dupliran broj reči:



2	0	0	0	10	6	0	0	2	0	0	0
6	2	0	0	4	0	0	2	0	2	0	0

$$similarity = 10.2$$

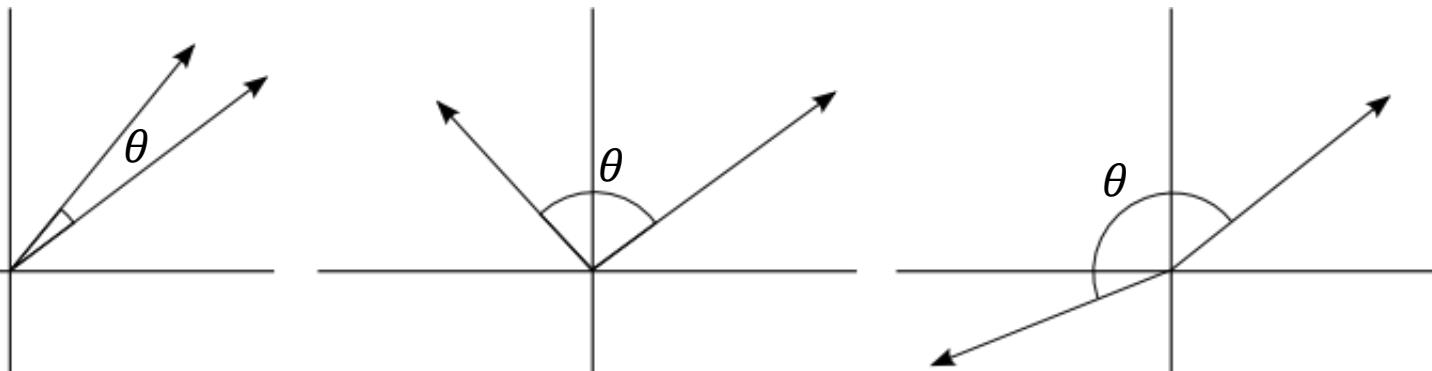
- Dokumenti su međusobno sličniji jer su duži (a ne zbog sadržaja)!
- Nekada je ovo opravdano, npr. poređenje kuća prema kvadraturi
- Ali nekada ovo ne želimo. Rešenje je da normalizujemo vektor na dužinu 1 (svaku komponentu vektora podelimo dužinom vektora)
- Npr. kosinusna sličnost je normalizovana

Kosinusna udaljenost

$$\text{similarity}(x^{(i)}, x^{(q)}) = \frac{\sum_{d=1}^D x_d^{(i)} x_d^{(q)}}{\sqrt{\sum_{d=1}^D (x^{(i)})^2} \sqrt{\sum_{d=1}^D (x^{(q)})^2}} = \frac{x^{(i)T} x^{(q)}}{\|x^{(i)}\| \|x^{(q)}\|}$$

$= \cos(\theta)$

↗
Dužina vektora
(normalizacija)



Similar scores
Score Vectors in same direction
Angle between them is near 0 deg.
Cosine of angle is near 1 i.e. 100%

Unrelated scores
Score Vectors are nearly orthogonal
Angle between them is near 90 deg.
Cosine of angle is near 0 i.e. 0%

Opposite scores
Score Vectors in opposite direction
Angle between them is near 180 deg.
Cosine of angle is near -1 i.e. -100%

Kosinusna udaljenost

Originalni dokumenti:



1	0	0	0	5	3	0	0	1	0	0	0	0
3	1	0	0	2	0	0	1	0	1	0	0	0

$$similarity = 0.54$$

Isti dokumenti, pri čemu je svakom dupliran broj reči:



2	0	0	0	10	6	0	0	2	0	0	0	0
6	2	0	0	4	0	0	2	0	2	0	0	0

$$similarity = 0.54$$

- Kosinusna sličnost je invarijantna u odnosu na dužinu dokumenta, fokusira se isključivo na sadržaj dokumenata

Normalizovati ili ne?

- Da li želimo da dokumenti budu sličniji što su duži ili nam je važan samo sadržaj?



long document



short tweet

Normalizacija može da rezultuje time da različiti objekti izgledaju mnogo sličniji



long document



long document

Čest kompromis: ograničiti maksimalnu u minimalnu dužinu dokumenta

Mere sličnosti/udaljenosti

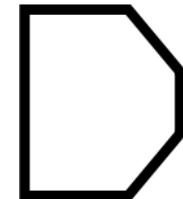
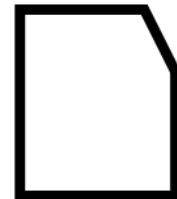
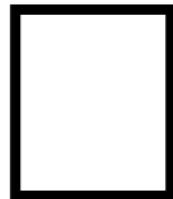
- Pored Euklidske i kosinusne mere postoje i mnoge druge mere sličnosti/udaljenosti
 - Manhattan, Jaccard, Hamming, Correlation-based, Rank-based, Mahalanobis,...
- Mere se mogu i kombinovati. Na primer, jedan dokument možemo reprezentovati tekstom i obeležjem koje nam govori koliko puta je dokument pročitan. Kada upoređujemo dva dokumenta:
 - Koristićemo kosinusnu sličnost na obeležjima vezanim za tekst
 - Koristićemo Euklidsku udaljenost za broj čitanja
- **Zaključak:** kako definisati metriku udaljenosti/sličnosti?
 - Nema jednog odgovora primenljivog na sve. Kada biramo meru, to veoma zavisi od rešavanog problema

Kako poređiti geometrijske oblike?

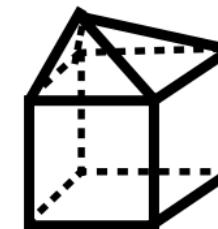
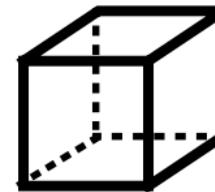
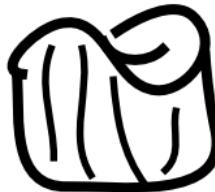
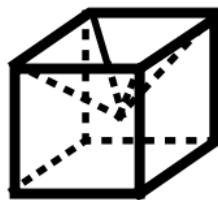
- Ljudska percepcija oblika može biti drugačija od objektivne, kao na primer optičke varke
- Da li postoje delovi slike koji su zajednički ili česti?
 - Na koji način efikasno identifikovati ovakve delove?
- Kako da iskoristimo znanje o poreklu slike u korist rezonovanja o njoj?

Kako poređiti slike – konture

- Na koji način opisati posepeni prelaz iz oblika na levoj strani u oblik na desnoj?



- Koje od ovih figura su slične i u kojoj meri?



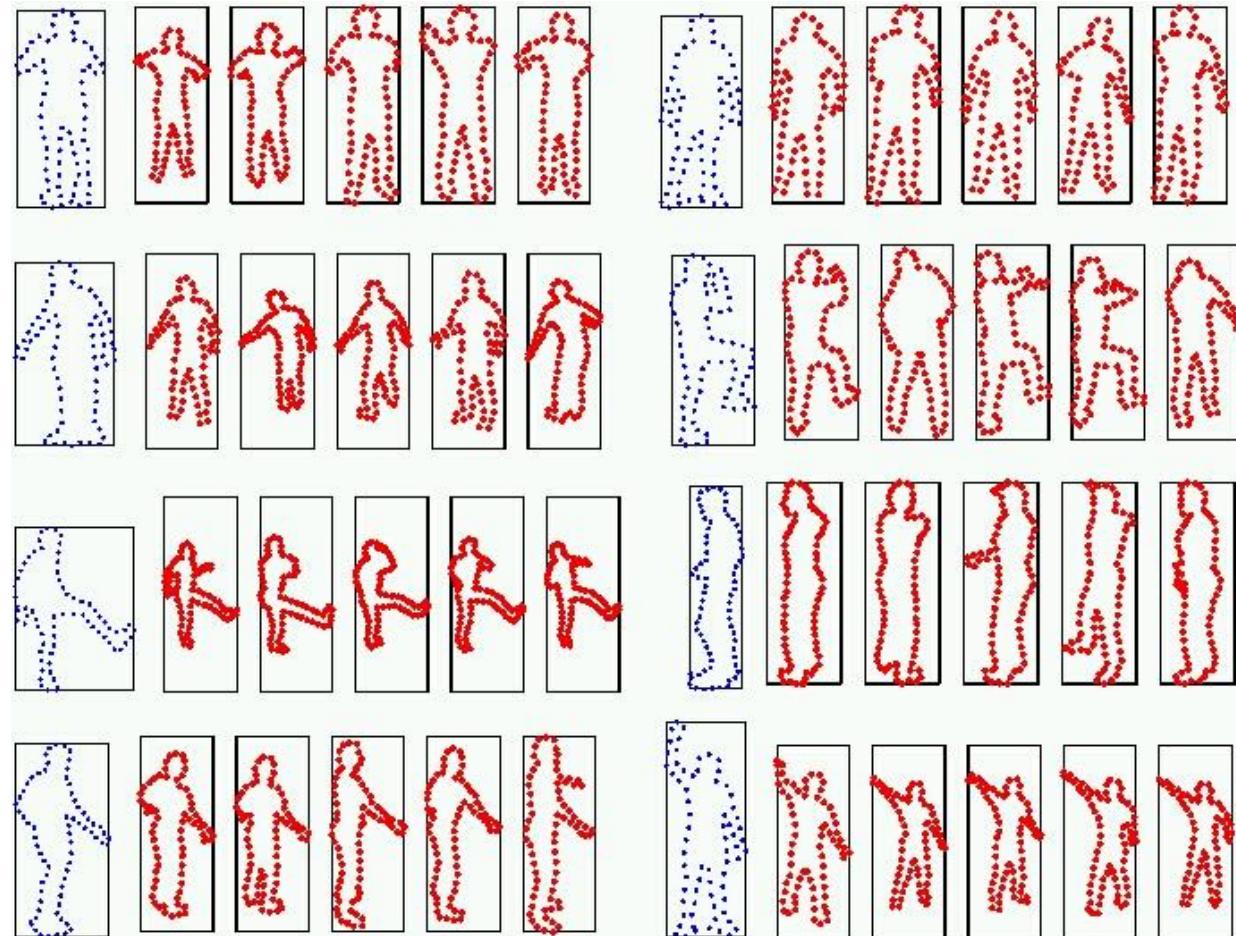
Primer: figure i delimično preklapanje

- Na koji način izmeriti i opisati sličnosti između sledeće tri figure



Siluete

- Kako iz video snimka izdvojiti siluete?
- Kako izabrati reprezentaciju za siluetu osobe, tako da je pogodna za prepoznavanje specifičnih postupaka?
- Na koji efektivan način možemo grupisati siluete?



Baze slika

- Kolekcija umetničkih dela: pronaći dela istog umetnika
- Baze medicinskih slika
 - *Image miner* – koristi se u patologiji. Omogućava stručnjacima da vide uzorke bakterija i brzo otkriju sličnosti
- *International Association of Paper Historians*
 - Stari dokumenti koji na sebi imaju neidentifikovan *watermark* se porede sa poznatim *watermarks*, što u nekim slučajevima omogućava određivanje starosti/regije
- *Photobook*
 - Policija koristi u svrhe identifikacije lica (poređenje skica ili fotografija osumnjičenih)
- Opšte baze:
 - Pronalaženje odgovarajuće slike za knjigu ili časopis
 - Pronalaženje slika određene klase objekata (konja, ptica,...) ili apstraktnih konceptata, stila,...

Image retrieval by annotation



Water lilies

Flowers in a pond

<Its biological name>

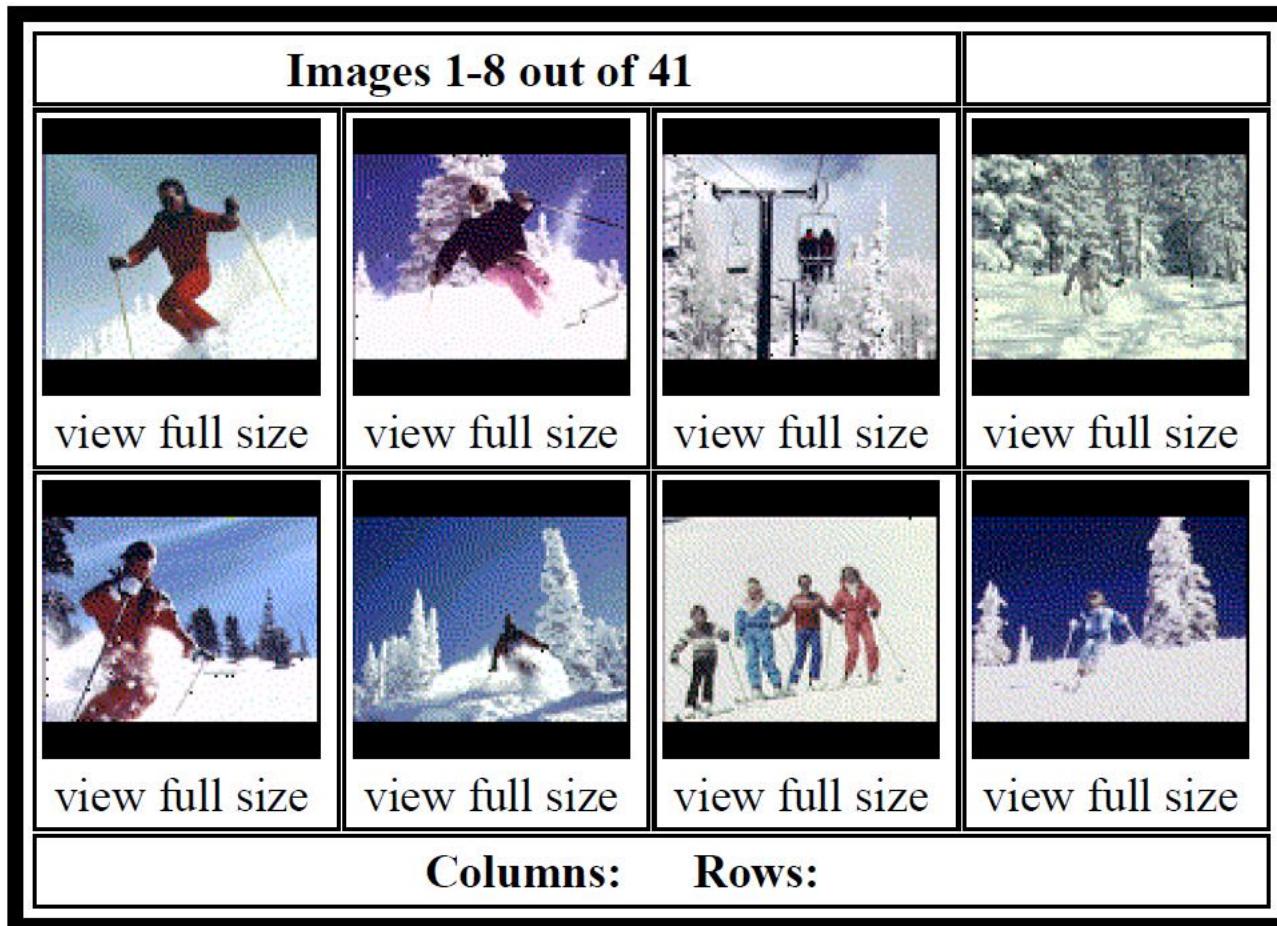
Image retrieval by annotation

- Baze slika mogu biti ogromne, a pretraga po dodeljenim ključnim rečima nezadovoljavajuća
 - Često su indeksirane prema ključnim rečima koje su određene i unete od strane ljudskog anotatora
 - Ljudske anotacije su skupe i subjektivne i veoma je verovatno da će neke korisne ključne reči biti izostavljene
 - Anotacije su na određenom jeziku – ovu prepreku ne bismo imali ako bismo gledali samu sliku
 - Prirodni jezik je po prirodi nejasan (*apple* – slika jabuke ili logo kompanije)
 - Problematično ako pretražujemo slike ne samo po sadržaju, već prema distribuciji boja, teksturi, oblicima,... i drugim vizuelnim svojstvima koje ne umemo jasno da izrazimo

Pretraga baza slika

- Rešenje: pretraga po datom primeru
 - Sistemu se kao primer da slika ili skica (a ne metapodaci)
 - Cilj je da sistem vrati slične slike
- Ovo se naziva:
 - *Content-based image retrieval (CBIR)* ili
 - *Query by image content (QBIC)*

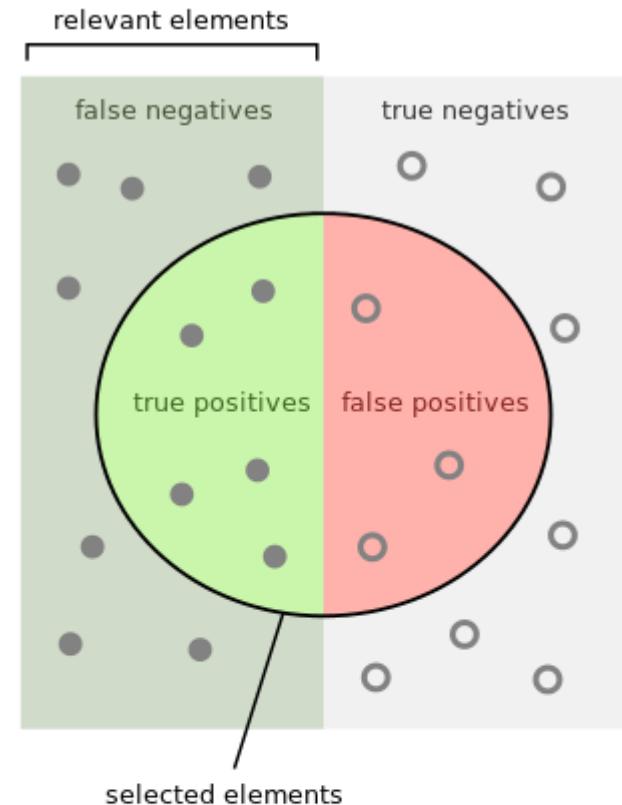
Primer



- Pretraga po raspodeli boja prema primeru gore levo

Evaluacija

- Obično se evaluira pomoću dva kriterijuma:
 - Odziv (*recall*): udeo pronađenih odgovarajućih slika u svim kandidatima
 - Preciznost (*precision*): udeo odgovarajućih slika među pronađenima
- Koji kriterijum je važniji?



How many selected items are relevant?

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

How many relevant items are selected?

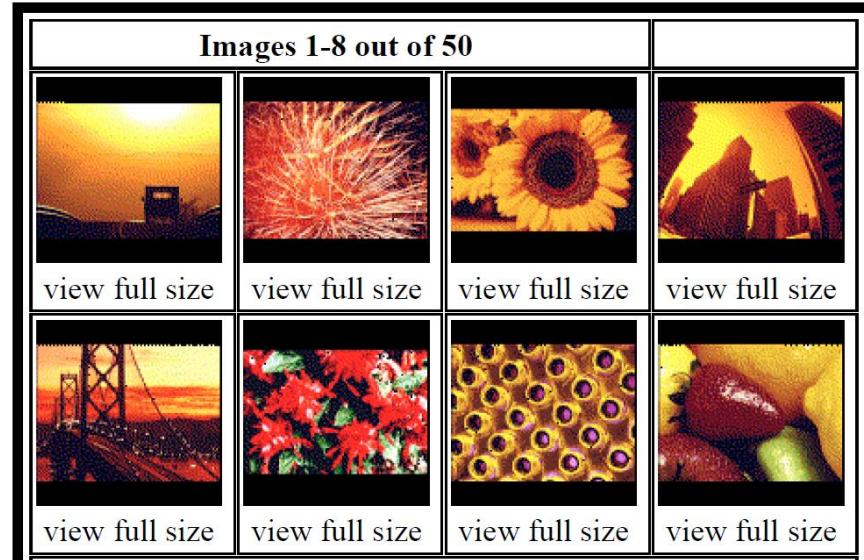
$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

Metrika sličnosti slika

- Moramo definisati rastojanje između slika
- Glavne klase sličnosti:
 - Sličnost boje
 - Sličnost tekture
 - Sličnost oblika
 - Postojanje određenih objekata (npr. lica)

Sličnost boja

- Varijante:
 - Srednja vrednost boje
 - Distribucija (histogram)
 - Relativne lokacije



- Rane (i još popularne) metrike sličnosti se baziraju na histogramu boja
 - RGB (ili drugi prostor boja) se diskretizuje u binove
 - Za svaki bin, odredi se broj piksela koji u njega upada
 - Rezultujući histogrami se porede različitim metrikama

Sličnost boja

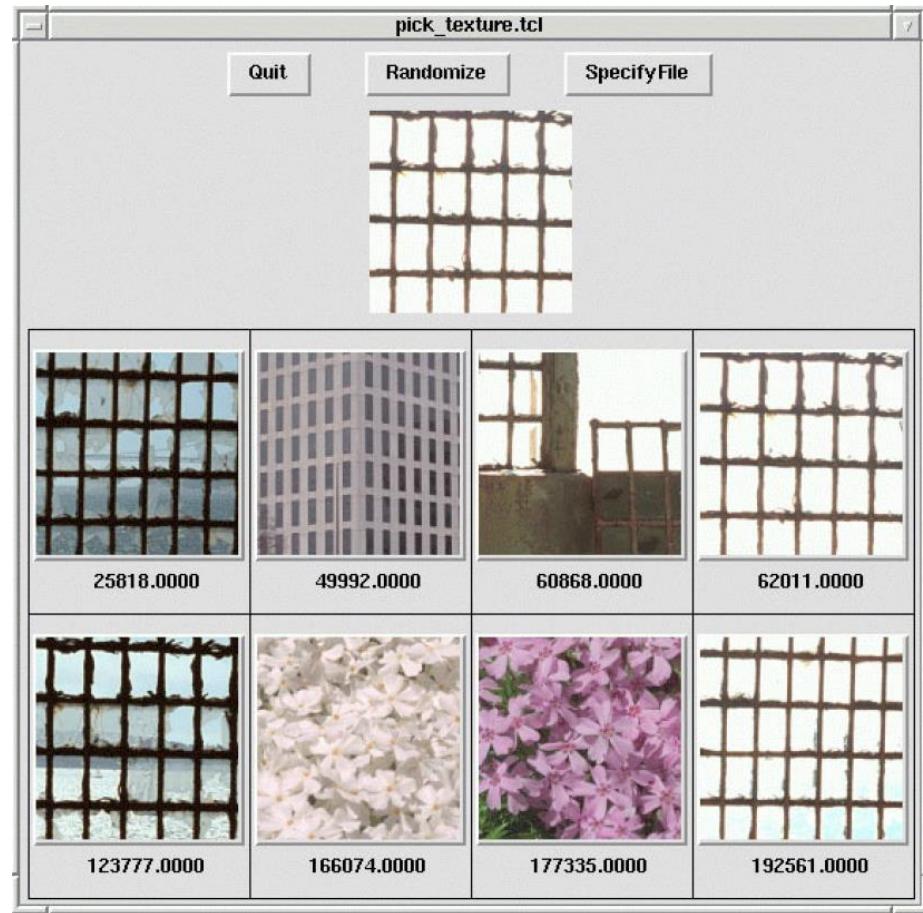
- QBIC sistem (IBM) je prvi komercijalni sistem
 - Koristi boje, teksture, oblike, lokacije i ključne reči
- Distanca bazirana na histogramu:

$$d_{hist}(I, Q) = (h(I) - h(Q))^T A (h(I) - h(Q))$$

- $h(I), h(Q)$ - histogrami od K binova
- A – $k \times k$ matrica sličnosti boja (veoma slične boje treba da imaju sličnost blisku 1, a veoma različite boje sličnost blisku 0)

Sličnost tekstura

- Tekstura je kompleksnija od boja – nema jasne definicije šta znači tekstura
 - Ali možemo gledati ponovljene šabloni piksela, njihova koncentraciju i frekvenciju
- Moramo definisati
 - Reprezentaciju strukture
 - Definiciju sličnosti u skladu sa tom reprezentacijom



Sličnost tekstura

- Najčešće korišćena reprezentacija teksture: *texture description vector*
 - Vektor brojeva koji sumarizuju teksturu na datoј regiji
 - *Haralick's five co-occurrence-based texture features*
 - *Laws' nine texture energy features*

$$d_{\text{texture}}(I, Q) = \|T(I) - T(Q)\|^2$$

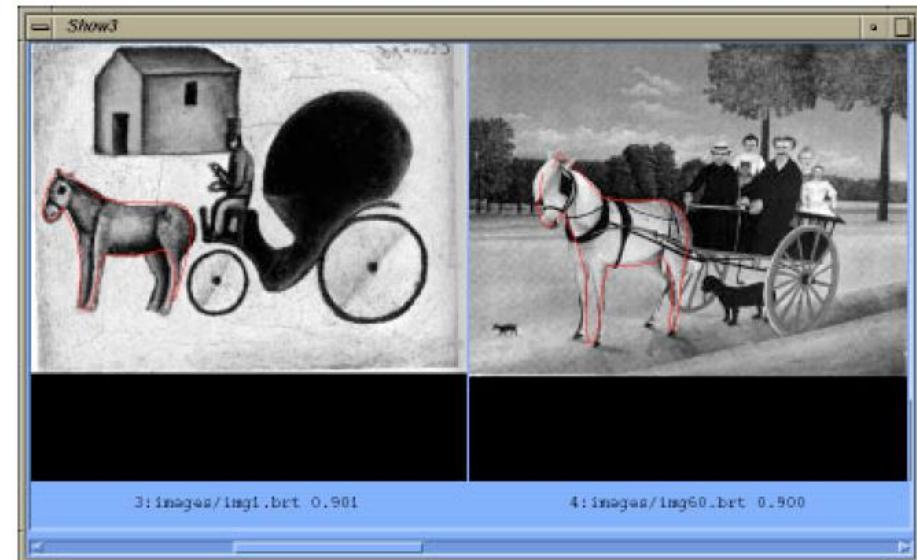
- Ovo je dobra mera za reprezentaciju slika sa jednom tipom teksture. Za celu sliku, postavimo mrežu (*grid*) preko slike i računamo vektor za svaki kvadrat

Sličnost oblika

- Na osnovu:
 - Skica
 - Segmentiranih objekata
- Globalne mere:
 - Dužina konture
 - Površina oblasti
 - Zakrivljenost konture
 - Projekcije na osu
 - Histogram uglova tangenti
 - ...

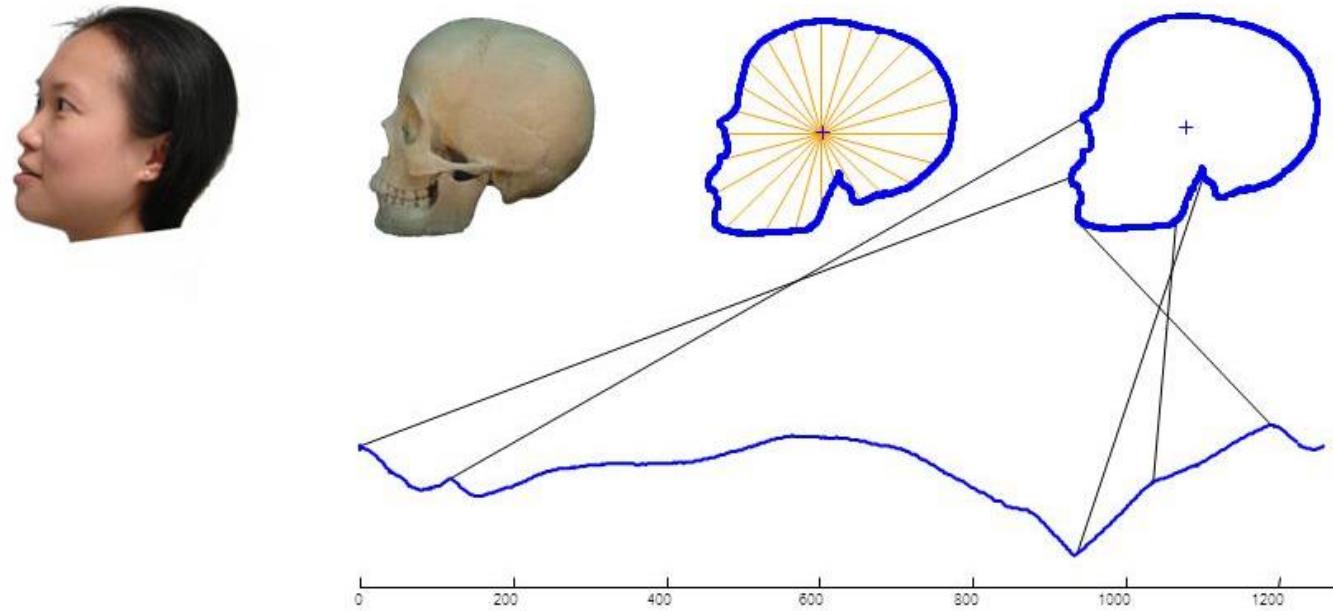


a) The user's query shape



Primer: figure i izdvajanje siluete

- Konturu neke figure iz videa je moguće pretvoriti u 1D signal.
- Ovakvi signali su pogodni za metrike i poređenja



Primer projekta

- Projekat:
 - Automatsko generisanje skupa podataka koji se sastoji od slika poznatih ličnosti
 - Za svaku poznatu ličnost obezbediti sledeće kategorije slika:
 - Sa/bez osmeha
 - Sa/bez brade
 - Sa/bez šiški
 - Različiti smerovi pogleda (gore, dole, levo, desno, pravo)
- Prvi korak:
 - Automatska kolekcija slika sa *Google Images* prema imenu osobe
 - Problem: za neka imena vraćene su slike koje ne sadrže traženu osobu već neku drugu
 - Rešenje: odrediti udaljenost lica, pronaći prosek i obrisati slike sa prevelikom udaljenošću
- Problem ostao nakon implementacije:
 - Iste slike, samo drugačije isecanje
 - Moglo bi se rešiti određivanjem sličnosti slika



K-Nearest Neighbors (K-NN)

Problem klasifikacije

- Dat je trening skup – niz objekata (primera) x kojima je pridružena klasa (obeležje) y :

$$T = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(N)}, y^{(N)})\}$$

- Kod klasifikacije y uzima *diskrete* vrednosti iz nekog predefinisanog skupa klasa, npr.
 - Klasifikujemo twitove kao pozitivne ili negativne
 - Klasifikujemo slike cifara u 10 kategorija (0,1,...,9)
- Kada stigne novi primer x^* kako dodeliti klasu y^* ?

Reprezentacija objekta

- Svaki objekat x je na neki način reprezentovan na računaru. Obično je u pitanju skup obeležja (atributa):

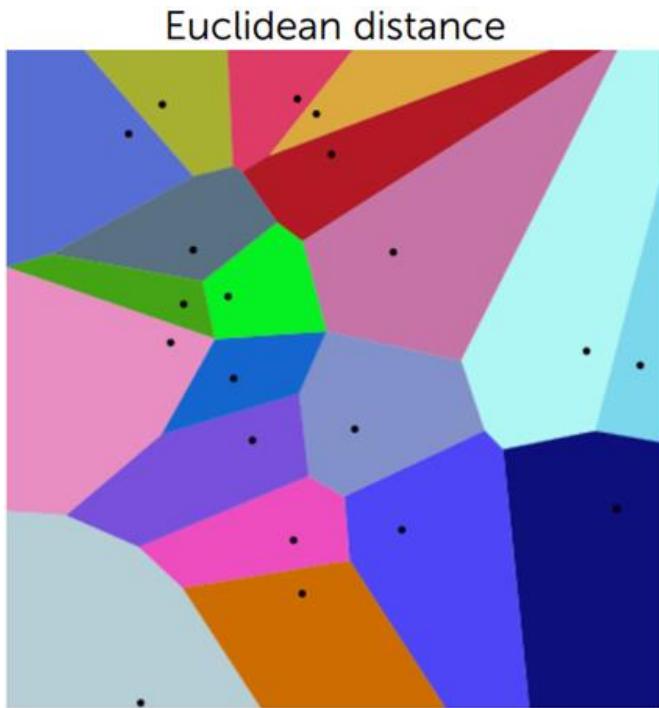
$$x^{(i)} = [x_1^{(i)}, x_2^{(i)}, \dots, x_D^{(i)}]$$

- Moguće su različite reprezentacije
 - Na primer, sliku dimenzije $m \times n$ možemo predstaviti vektorom x veličine $m \times n$ gde vrednosti vektora predstavljaju vrednost intenziteta piksela
 - Možemo smisliti i drugačiji način reprezentacije slike – npr. sliku možemo predstaviti sa dva obeležja – prosečan intenzitet piksela na slici i mera simetrije slike
 - Drugi način bi bio dobar za prepoznavanje cifara sa slika, ali verovatno ne toliko dobar za neke druge primene
 - Kako reprezentovati objekat je veoma važan problem od koga jako zavise perfotmanse obučavajućeg algoritma

Jedan najbliži sused (1-NN)

- Nađimo objekat x^+ koji je najsličniji novom primeru x^* i dodelimo novom primeru njegovu klasu $y^* = y^+$
- Ali kako se definiše *slično*?
 - Najčešće se koriste *Euklidska* ili *Gausova* razdaljina
 - Kao što smo već razmatrali, izbor udaljenosti će veoma uticati na kvalitet rešenja

Primer Euklidske razdaljine

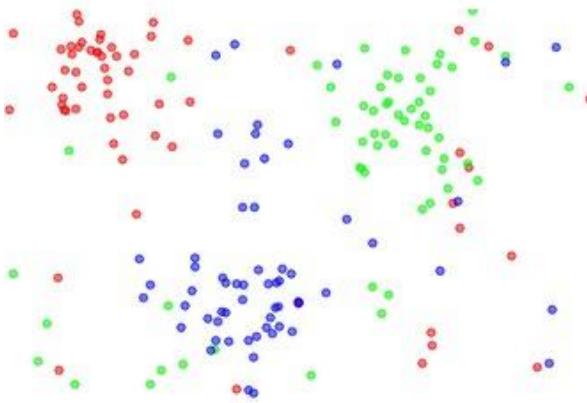


Vizuelizacija 1-NN za *Euklidsku razdaljinu* –
Voronoi-ev dijagram:

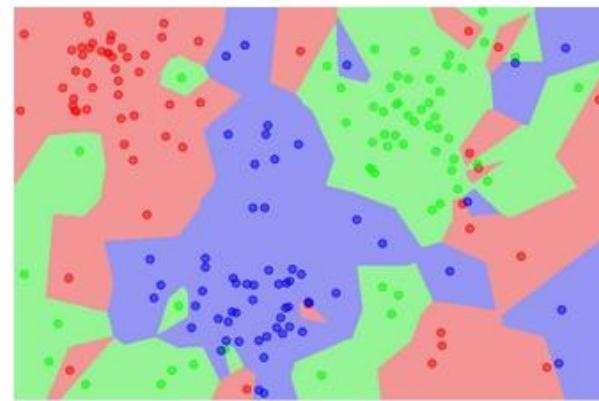
- Podeliti prostor u N regija, gde svaka regija sadrži tačno jednu tačku $x^{(i)}$ od N tačaka iz trening skupa
- Regije su definisane tako da je svakoj tački iz regije „najbliža“ tačka trening skupa $x^{(i)}$
- Ne moramo eksplicitno formirati ove regije (određivati granice), dovoljna nam je definicija metrike udaljenosti

K-NN

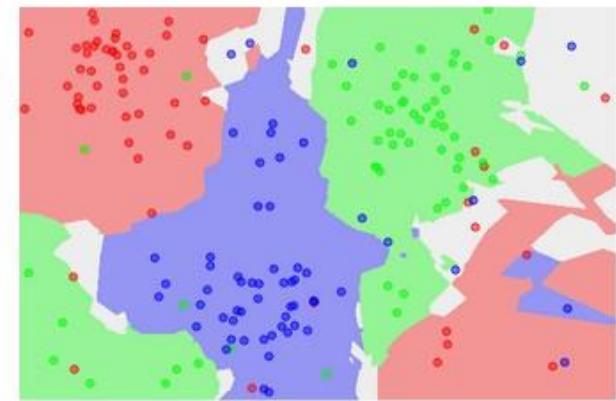
the data



NN classifier

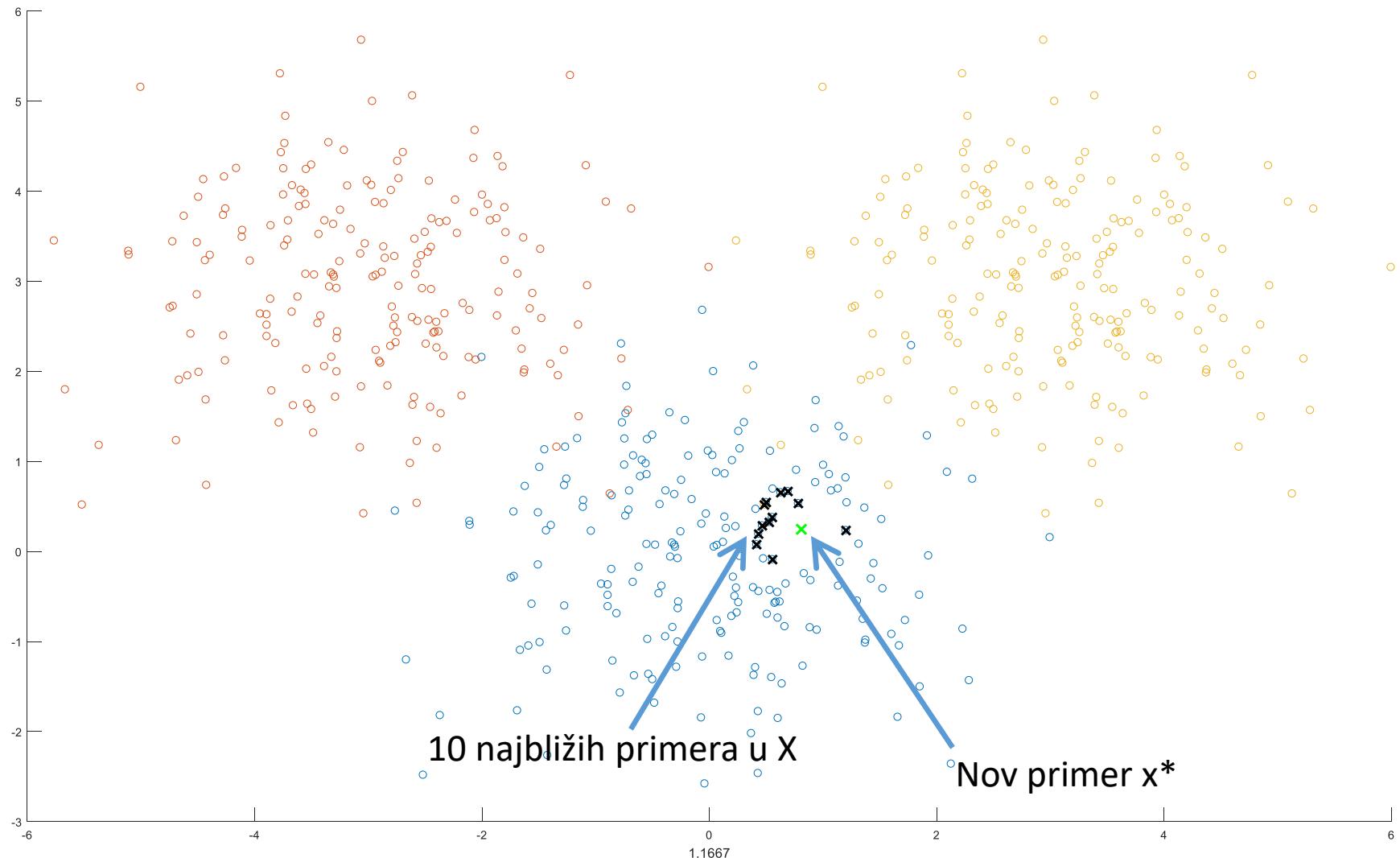


5-NN classifier



- Problem sa 1-NN su potencijalni *outlieri*
 - Primetite zelenu tačku u sred plavih – stvara malo ostrvo (verovatno) pogrešnih predikcija
- Rešenje: *K*-NN:
 - Pronaći *K* najsličnijih primera $\{x^{*1}, x^{*2}, \dots, x^{*K}\}$ i odrediti y^* kao najzastupljeniju klasu ovog skupa

K-NN primer



K-NN

- Kako odrediti K ?
 - Poseban validacioni skup ili unakrsna validacija
- Može imati veoma loše performanse kada je broj obeležja velik
 - Razlog: prokletstvo dimenzionalnosti – najbliži susedi u visokodimenzionom prostoru mogu biti veoma daleko

