ALGORITMI ANALIZE MASIVNIH PODATKOV

DOMEN MONGUS

P03 – Opisna analiza

Motivacija

- Analiza nakupov
  - Začetek Big Data analiz
- Apriori algoritem
  - Najbolj citiran članek na področju podatkovnega rudarjenja
- Temeljno vprašanje:
  - Kakšne so navade kupcev?

Motivacija

- Kaj kupuje Homer Simpson poleg plenic?

- Odgovor:
  - Če kupuješ plenice imaš doma verjetno otroka
  - Ker imaš otroka, verjetno ne hodiš dosti v ven
  - Izkaže se, da poleg plenice kupuješ pivo

Motivacija

- Kaj kupuje Homer Simpson poleg plenic?

- Posledica:
  - V trgovinah imamo skupaj pivo in plenice
  - Plenice daš v akcijo in dvigneš ceno piva
  - Lahko damo v akcijo tudi pivo?

Vsebina

- Opisna statistika
- Asociacijska pravila
- Apriori algoritem

Opisna statistika

- Iskanje opisnih (meta) podatkov

- Statistični povzetki:
  - Srednje (pričakovane) vrednosti
  - Spremenljivost (disperzija)

- Običajno izdelamo iz histograma
  - Rešujemo problem volumna

Opisna statistika

- Osnovni podatki:
  - N = Število elementov
  - *Min* n = Najmanjši element
  - *Max* n = Največji element
  - Razpon vrednosti $n \in [Min\ n, Max\ n]$

- Histogram $H[k]$ = *število elementov v košu* k
  - K = *število košev*
  - $k \in [0, K-1]$

Opisna statistika

- Kako določiti število košev K?
  - $K = \frac{Max\ n - Min\ n}{h}$, kjer je *h* velikost koša
  - $K = \sqrt{n}$
  - Normalna porazdelitev običajno

$$k = \lceil \log_2 n \rceil + 1,$$

  - Izboljšava za nenormalno porazdelitev

$$k = 1 + \log_2(n) + \log_2\left(1 + \frac{|g_1|}{\sigma_{g_1}}\right)$$

Opisna statistika

- Povprečje na osnovi histograma

$$\bar{n} = \frac{\sum_{k=0}^{K} k * H[k]}{\sum_{k=0}^{K} H[k]}$$

- Pričakovana vrednost $= arg\ max\ H[k]$

- Kako izračunamo mediano?

- Standardni odklon

$$\bar{n} = \sqrt{\frac{\sum_{k=0}^{K} k * (H[k] - \bar{n})}{\sum_{k=0}^{K} H[k]}}$$

Opisna statistika

- Spremenljivost

$$m_1 = \frac{\sum(x_i - \bar{x})}{N}$$

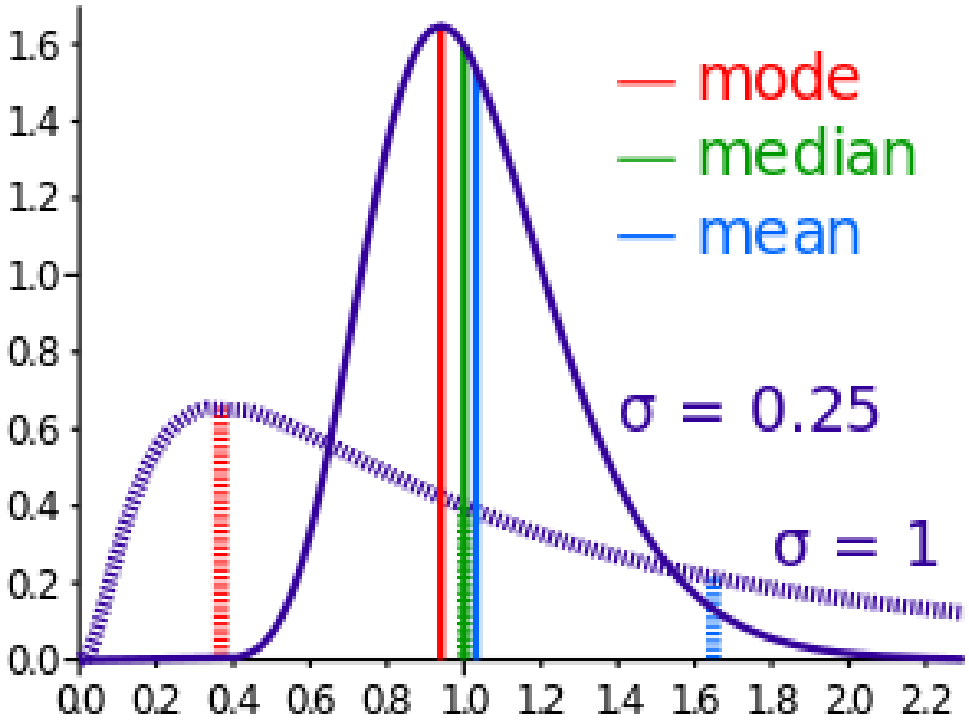
$$m_2 = \frac{\sum(x_i - \bar{x})^2}{N}$$

$$m_3 = \frac{\sum(x_i - \bar{x})^3}{N}$$

$$m_4 = \frac{\sum(x_i - \bar{x})^4}{N}$$

Asimetrija (skewness)

$$b_1 = \frac{m_3}{s^3} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^3}{\left[\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2\right]^{3/2}},$$



Opisna statistika

- Spremenljivost

$$m_1 = \frac{\sum (x_i - \bar{x})}{N}$$

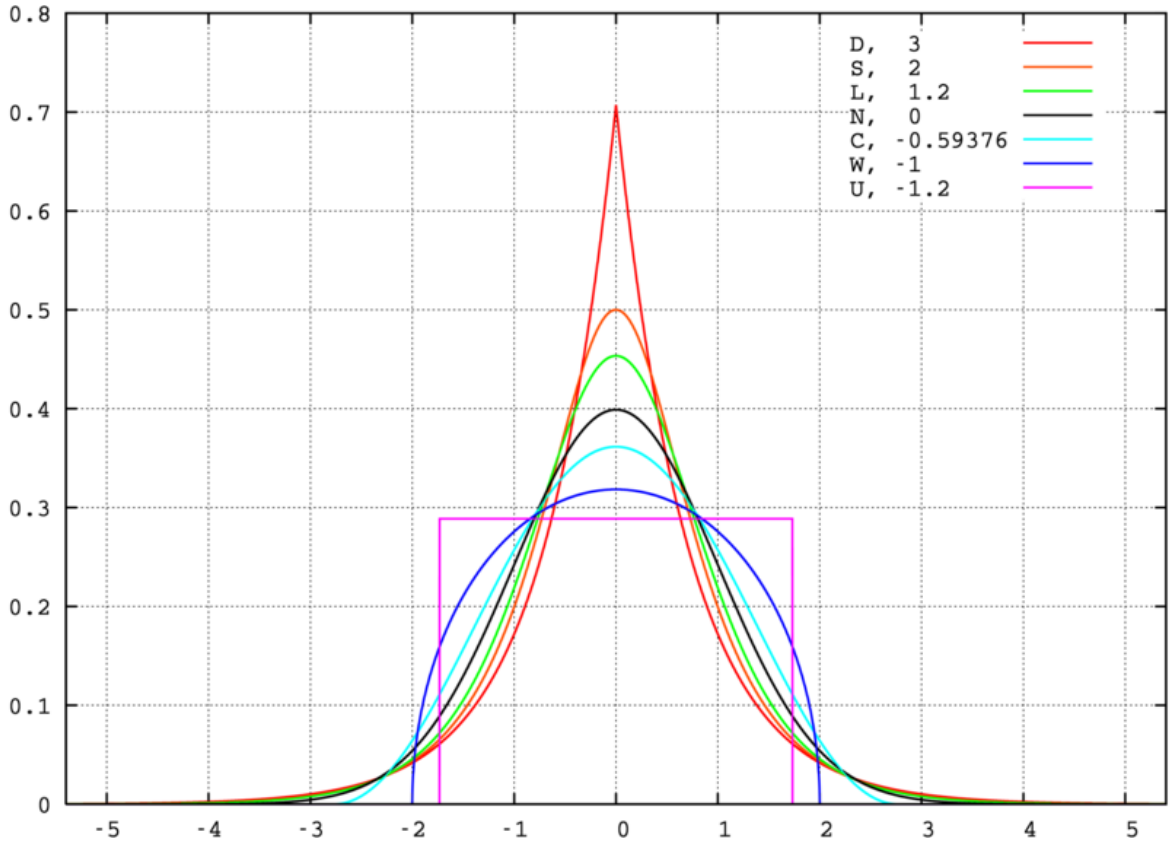
$$m_2 = \frac{\sum (x_i - \bar{x})^2}{N}$$

$$m_3 = \frac{\sum (x_i - \bar{x})^3}{N}$$

$$m_4 = \frac{\sum (x_i - \bar{x})^4}{N}$$

Sploščenost (kurtosis)

$$g_2 = \frac{m_4}{m_2^2} - 3 = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^4}{\left(\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2\right)^2} - 3$$



Model trgovine in nakupovalnega vozička

Množica pogosto kupljenih izdelkov

- Velik nabor artiklov
- Velik nabor nakupovalnih vozičkov
  - V vsakem vozičku malo artiklov
- Katere artikli so „pogosto“ kupljeni?
  - **Podorno število** množice *I*

$$sup(I) = \frac{\text{št. } vozičkov\ v\ katerih\ je\ I}{\text{št. } vseh\ vozičkov},$$

  - **Podporna pragovna vrednost** *s* določa množico pogosto kupljenih izdelkov

Primer

- Množica artiklov = {**m**arelice, **č**ešnje, **p**omaranče, **b**reskve, **j**agode}
- Podporna pragovna vrednost s = 33%, približno 3

  B_1={m,c,b} B_2={m,p,j}

  B_3={m, b} B_4={c,j}

  B_5={m,p,b} B_6={m,c,b,j}

  B_7={c,b,j} B_8={b,c}

- Kateri so pogosti artikli:
  - {m}, {c}, {b}, {j},
  - {m,b}, {b,c}, {c,j}

Formalizacija

- Iščemo „if-then“ relacije

  $\{i_1, i_2, \ldots, i_k\} \rightarrow j$

  $I \rightarrow j$, če $= \{i_1, i_2, \ldots, i_k\}$

- Zaupanje:

$$conf(I \rightarrow j) = \frac{\sup(I \cup j)}{\sup(I)} = P(j|I)$$

- Kakšno je zaupneje v pravilo {m,b} → c ?

Formalizacija

- Iščemo „if-then“ relacije

  $\{i_1 , i_2 , \ldots, i_k\} \rightarrow j$

  $I \rightarrow j$, če $= \{i_1 , i_2 , \ldots, i_k\}$

- Zaupanje:

  $$conf(I \rightarrow j) = \frac{\sup(I \cup j)}{\sup(I)} = P(j|I)$$

- Kakšno je zaupneje v pravilo {m,b} → c ?

  $conf(\{m,b\} \rightarrow c) = 50\%$

Naivni Algoritem

- **Zanimajo nas vsa asociativna pravila, ki imajo podporo večjo od *s* in zaupanje večje kot *c***
- Osvnovna ideja, če ima *I* podporo *s*, potem ima pravilo $I \rightarrow j$ podporo, ki je vsaj *cs*. Zato:
  - A vsebuje vse množice, ki imajo podporo vsaj *cs*
  - B vsebuje vse množice, ki imajo vsaj *s* (tako $A \subseteq B$)
    - $B \backslash A$ definira pravila
    - Poiščimo torej tiste množice, ki so v B in jih ni v A ter ugotovimo kateri element jim manjka
- Ne pozabimo, problem je Big Data!

Naivni Algoritem

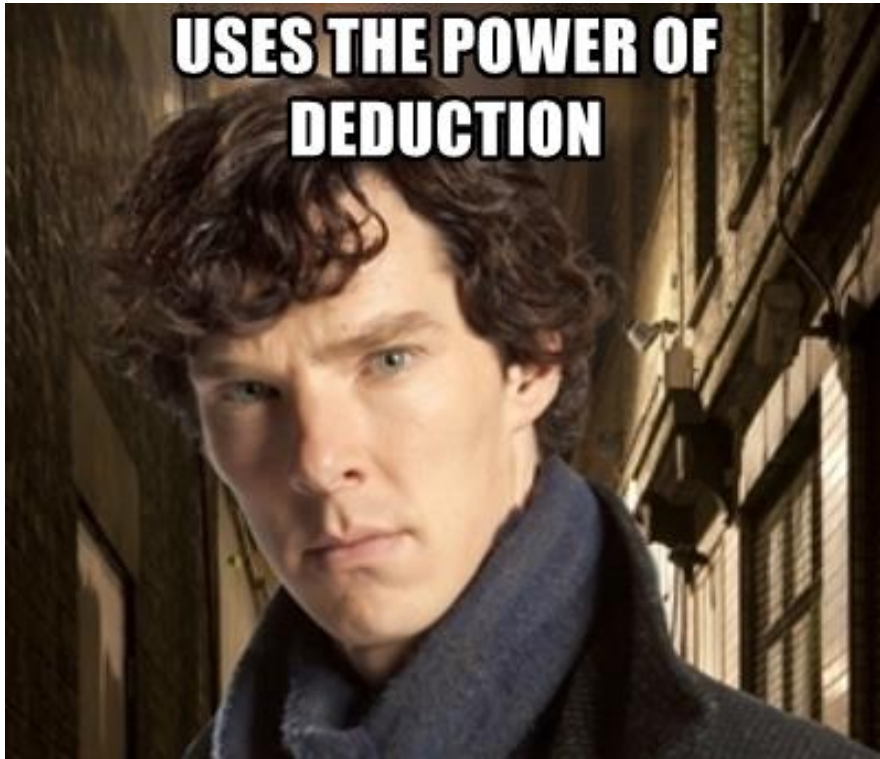
- Problem iskanja asociativnih pravil je enak problemu iskanja pogosto kupljenih artiklov
  - Zahteva uporabo DAM!
- Prešteti moremo vse množice s podporo cs
  - V bistvu histogram
- Štetje zahteva:
  - Izvedbo trikotna matrike
  - Izvedbo seznama parov

Apriorni algoritem

- Množica artiklov ne more biti pogosta, če vse njene podmnožice niso pogoste!
  - Časovna zahtevnost algoritma je enaka podpornemu številu, ki ga iščemo!
- Postopno filtriranje glede na dedukcijo!☺
  - **Monotonost (matematično):**

    $$\forall x, y : x \leq y \Rightarrow f(x) \leq f(y)$$

  - **Praktično:** če se par artiklov pojavi v s nakupovalnih vozičkih, se vsak izmed para artiklov sam pojavi vsaj s-krat.



Apriorni algoritem

- **Z drugimi besedami:** če se artikel ne pojavi v vozičku vsaj *s*-krat, potem se tudi noben izmed njegovih parov ne:
  - **Prehod 1**: Preštejemo število pojavitev vsakega artikla in brišemo vse, ki se ne pojavijo vsaj *s*-krat
    - Generiramo seznam kandidatov parov pogosto kupljenih artiklov
  - **Prehod 2**: Preštejemo kolikokrat se pojavijo pari pogosto kupljenih artiklov (vse filtrirane preskočimo) in zopet filtriramo.
    - Generiramo seznam trojic pogosto kupljenih artiklov.

Apriorni algoritem

- **Generiranje trojic pogosto kupljenih artiklov:**
  - Izberemo par, iz katerega želimo generirati trojice
  - Izberemo drugi par, ki vsebuje vsaj en element, v prvem paru
  - Dobimo tretji element, ki definira trojico.

Primer

- Množica artiklov = {**m**arelice, **č**ešnje, **p**omaranče, **b**reskve, **j**agode}
- Podporna pragovna vrednost s = 3 in c=50%

B_1={m,c,b} B_2={m,p,j}

B_3={m, b} B_4={c,j}

B_5={m,p,b} B_6={m,c,b,j}

B_7={c,b,j} B_8={b,c}

- Korak 1:
  - sup({m})=5
  - sup({c})=5
  - sup({p})=2
  - sup({b})=5
  - sup({j})=4
- Filtriranje
  - {m,c,b,j}

Primer

- Množica artiklov = {**m**arelice, **č**ešnje, **p**omaranče, **b**reskve, **j**agode}
- Podporna pragovna vrednost s = 3 in c=50%

B_1={m,c,b} B_2={m,p,j}

B_3={m, b} B_4={c,j}

B_5={m,p,b} B_6={m,c,b,j}

B_7={c,b,j} B_8={b,c}

- Filtriranje
  - {m,c,b,j}
- Generiramo pare:
  - sup({m,c})=2
  - sup({m,b})=4
  - sup({m,j})=2
  - sup({c,b})=4
  - sup({c,j})=3
  - sup({b,j})=2

Primer

- Množica artiklov = {**m**arelice, **č**ešnje, **p**omaranče, **b**reskve, **j**agode}
- Podporna pragovna vrednost s = 3 in c=50%

B_1={m,c,b} B_2={m,p,j}
B_3={m, b} B_4={c,j}
B_5={m,p,b} B_6={m,c,b,j}
B_7={c,b,j} B_8={b,c}

- Generiramo pare:
  - sup({m,c})=2
  - sup({m,b})=4
  - sup({m,j})=2
  - sup({c,b})=4
  - sup({c,j})=3
  - sup({b,j})=2
- Filtriranje
  - Število parov je 6
  - Prag = 3

Primer

- Množica artiklov = {**m**arelice, **č**ešnje, **p**omaranče, **b**reskve, **j**agode}
- Podporna pragovna vrednost s = 3 in c=50%

B_1={m,c,b} B_2={m,p,j}

B_3={m, b} B_4={c,j}

B_5={m,p,b} B_6={m,c,b,j}

B_7={c,b,j} B_8={b,c}

- Filtriranje
  - sup({m,b})=4
  - sup({c,b})=4
  - sup({c,j})=3
- Izračunamo trojice
  - sup({m,b,c})=2
  - sup({c,b,j})=2