

Novo doba

Ivan Slapničar*

Split, 12. studenog 2016.

Predavanje na otvorenju Centra izvrsnosti iz matematike Splitsko-dalmatinske županije

1 Poruke

Lako - Teško (u svemu!)

Sada - doba **pretraživanja** -> *matematika*

Dolazi - doba **preporuka** -> *matematika*

(BigData, nova tehnologija)

2 Lako - teško

vaganje: $1\text{ kg} \rightarrow 1\text{ kg} \pm 0.000000001\text{ kg}$

trčanje: $100\text{ m} \rightarrow 42,195\text{ m}$ ili $100\text{ m} < 10\text{ sek}$

zadaci iz matematike: test za CI -> državno natjecanje ([2016](#) ili [sve](#)), ili [matematička olimpijada](#)

3 Doba pretraživanja

google (i ostali)

- [50 milijardi stranica](#), [3.5 milijarde pretraživanja dnevno](#)
- **PageRank**
- povijest, kontekst - cookies, spremanje podataka (o vama), [200+ parametara](#)

4 PageRank

- Teorija grafova i linearna algebra
- [C](#), [Moler](#), [Google PageRank](#)

```
In [1]: i = vec([ 2 6 3 4 4 5 6 1 1])
        j = vec([ 1 1 2 2 3 3 3 4 6])
        G=sparse(i,j,1.0)
```

*Sveučilište u Splitu, Fakultet elektrotehnike, strojarstva i brodogradnje, R. Boškovića 32, 21000 Split, ivan.slapnicar@fesb.hr

```
Out [1]: 6×6 sparse matrix with 9 Float64 nonzero entries:
```

```
[2, 1] = 1.0
[6, 1] = 1.0
[3, 2] = 1.0
[4, 2] = 1.0
[4, 3] = 1.0
[5, 3] = 1.0
[6, 3] = 1.0
[1, 4] = 1.0
[1, 6] = 1.0
```

```
In [2]: full(G)
```

```
Out [2]: 6×6 Array{Float64,2}:
```

```
0.0  0.0  0.0  1.0  0.0  1.0
1.0  0.0  0.0  0.0  0.0  0.0
0.0  1.0  0.0  0.0  0.0  0.0
0.0  1.0  1.0  0.0  0.0  0.0
0.0  0.0  1.0  0.0  0.0  0.0
1.0  0.0  1.0  0.0  0.0  0.0
```

Na graf primijenimo slučajnu šetnju (*random walk*)- vjerojatnost da pratimo bilo koji link je jednaka:

```
In [3]: c=sum(G,1)
n=size(G,1)
for j=1:n
    if c[j]>0
        G[:,j]=G[:,j]/c[j]
    end
end
full(G)
```

```
Out [3]: 6×6 Array{Float64,2}:
```

```
0.0  0.0  0.0      1.0  0.0  1.0
0.5  0.0  0.0      0.0  0.0  0.0
0.0  0.5  0.0      0.0  0.0  0.0
0.0  0.5  0.333333  0.0  0.0  0.0
0.0  0.0  0.333333  0.0  0.0  0.0
0.5  0.0  0.333333  0.0  0.0  0.0
```

- p : vjerojatnost da pratimo neki link
- $1 - p$: je vjerojatnost da odemo na neku drugu slučajnu stranicu
- google koristi $p = 0.85$?

```
In [4]: p=0.85
z = ((1-p)*(c.!=0) + (c.==0))/n
A=p*G+ones(n)*z
```

```
Out [4]: 6×6 Array{Float64,2}:
 0.025  0.025  0.025      0.875  0.166667  0.875
 0.45   0.025  0.025      0.025  0.166667  0.025
 0.025  0.45   0.025      0.025  0.166667  0.025
 0.025  0.45   0.308333  0.025  0.166667  0.025
 0.025  0.025  0.308333  0.025  0.166667  0.025
 0.45   0.025  0.308333  0.025  0.166667  0.025
```

Krenimo u slučajnu šetnju od vektora $x_0 = \begin{bmatrix} 1/n \\ 1/n \\ \vdots \\ 1/n \end{bmatrix}$.

Sljedeći vektori u nizu su

$$x_1 = A \cdot x_0, x_2 = A \cdot x_1, x_3 = A \cdot x_2, \dots$$

Kada se algoritam stabilizira, odnosno kada je

$$A \cdot x \approx x,$$

tada je element vektora $x[i]$ **rang stranice i** .

```
In [5]: function myPageRank(G::SparseMatrixCSC{Float64,Int64}, steps::Int)
    p=0.85
    c=sum(G,1)/p
    n=size(G,1)
    for i=1:n
        G.nzval[G.colptr[i]:G.colptr[i+1]-1]./=c[i]
    end
    e=ones(n)
    x=e/n
    z = vec(((1-p)*(c.!=0) + (c.==0))/n)
    for j=1:steps
        x=G*x+(z*x)
    end
    x/norm(x,1)
end
```

```
Out [5]: myPageRank (generic function with 1 method)
```

```
In [6]: myPageRank(G,15)
```

```
Out [6]: 6-element Array{Float64,1}:
 0.321024
 0.170538
 0.106596
 0.136795
 0.0643103
 0.200737
```

4.1 Stanford web graph

Malo veći testni problem.

```
In [7]: W=readdlm("web-Stanford.txt", Int)
```

```
Out [7]: 2312497×2 Array{Int64,2}:
```

```
      1      6548
      1     15409
     6548     57031
    15409     13102
      2     17794
      2     25202
      2     53625
      2     54582
      2     64930
      2     73764
      2     84477
      2     98628
      2    100193
      ⋮
    281849    165189
    281849    177014
    281849    226290
    281849    243180
    281849    244195
    281849    247252
    281849    281568
    281865    186750
    281865    225872
    281888    114388
    281888    192969
    281888    233184
```

```
In [8]: S=sparse(W[:,2],W[:,1],1.0)
```

```
Out [8]: 281903×281903 sparse matrix with 2312497 Float64 nonzero entries:
```

```
      [6548 ,      1] = 1.0
      [15409 ,      1] = 1.0
      [17794 ,      2] = 1.0
      [25202 ,      2] = 1.0
      [53625 ,      2] = 1.0
      [54582 ,      2] = 1.0
      [64930 ,      2] = 1.0
      [73764 ,      2] = 1.0
      [84477 ,      2] = 1.0
      [98628 ,      2] = 1.0
      ⋮
    [168703, 281902] = 1.0
```

```
[180771, 281902] = 1.0
[266504, 281902] = 1.0
[275189, 281902] = 1.0
[44103 , 281903] = 1.0
[56088 , 281903] = 1.0
[90591 , 281903] = 1.0
[94440 , 281903] = 1.0
[216688, 281903] = 1.0
[256539, 281903] = 1.0
[260899, 281903] = 1.0
```

```
In [9]: @time x100=myPageRank(S,100);
```

```
1.456710 seconds (846.80 k allocations: 542.598 MB, 11.93% gc time)
```

```
In [10]: x101=myPageRank(S,101)
         maxabs((x100-x101)./x100)
```

```
Out[10]: 2.349138102570129e-7
```

```
In [11]: sortperm(x100, rev=true), sort(x101, rev=true)
```

```
Out[11]: ([89073, 226411, 241454, 262860, 134832, 234704, 136821, 68889, 105607, 69358 ...
```

5 Doba preporuka

NetFlix, Amazon Prime, PickBox, ... - on-line videoteke (filmovi i serije)

[NetFlix](#)

- [80 milijuna korisnika](#), 5.000 filmova
- [NetFlix Prize](#)

6 Matematika

Netflix Recommendation Engine se temelji na aproksimaciji (velike i rijetke) matrice

$M = \text{Korisnici} \times \text{Filmovi}$

pomoću rastava singularnih vrijednosti (SVD):

- [IncrementalSVD.jl](#)
- [A parallel recommendation engine in Julia](#)

7 Sličnost

Sličnost korisnika i i k :

$$\cos \angle(i, k) = \frac{(M[i, :], M[k, :])}{\|M[i, :]\| \cdot \|M[k, :]\|}$$

Sličnost filmova i i k :

$$\cos \angle(i, k) = \frac{(M[:, i], M[:, k])}{\|M[:, i]\| \cdot \|M[:, k]\|}$$

8 Pretraživanje

Redak $M[k, :]$ - što korisnik k misli o filmovima.

Stupac $M[:, f]$ - što o filmu f misle korisnici

Element $M[k, f]$ - što korisnik k misli o filmu f .

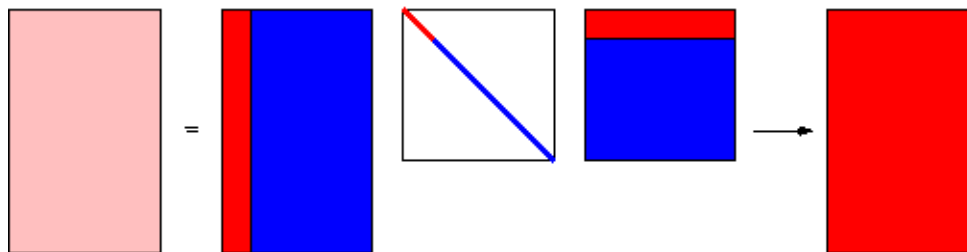
9 Problem

Matrica M je rijetka pa nemamo dovoljno informacija. Na primjer,

900188 ocjena / (6040 korisnika x 3706 filmova) = 4%

10 Aproksimacija

SVD rastav $M = U\Sigma V^T$ se aproksimira matricom jako malog ranga (npr. $m = 25$)



SVD rastav

Približna matrica je puna i daje dovoljno informacija.

Efikasna aproksimacija? -> \$ 1.000.000

```
In [12]: # Pkg.clone("git://github.com/aaw/IncrementalSVD.jl.git")
```

```
In [13]: using IncrementalSVD
```

WARNING: deprecated syntax "[a=>b for (a,b) in c]".
Use "Dict{a=>b for (a,b) in c}" instead.

```
In [14]: whos(IncrementalSVD)
```

IncrementalSVD	57 KB	Module
Rating	136 bytes	DataType
RatingSet	148 bytes	DataType
RatingsModel	160 bytes	DataType
cosine_similarity	0 bytes	IncrementalSVD.#cosine_similarity
get_predicted_rating	0 bytes	IncrementalSVD.#get_predicted_rati...
item_features	0 bytes	IncrementalSVD.#item_features
item_search	0 bytes	IncrementalSVD.#item_search
items	0 bytes	IncrementalSVD.#items

load_book_crossing_dataset	0 bytes	IncrementalSVD.#load_book_crossing...
load_large_movielens_dataset	0 bytes	IncrementalSVD.#load_large_moviele...
load_small_movielens_dataset	0 bytes	IncrementalSVD.#load_small_moviele...
rmse	0 bytes	IncrementalSVD.#rmse
similar_items	0 bytes	IncrementalSVD.#similar_items
similar_users	0 bytes	IncrementalSVD.#similar_users
split_ratings	0 bytes	IncrementalSVD.#split_ratings
train	0 bytes	IncrementalSVD.#train
truncate_model!	0 bytes	IncrementalSVD.#truncate_model!
user_features	0 bytes	IncrementalSVD.#user_features
user_ratings	0 bytes	IncrementalSVD.#user_ratings
users	0 bytes	IncrementalSVD.#users

```
In [15]: rating_set = load_small_movielens_dataset();
```

Reusing existing downloaded files...

Loading ratings 100% Time: 0:00:03

```
In [16]: fieldnames(rating_set)
```

```
Out[16]: 4-element Array{Symbol,1}:
 :training_set
 :test_set
 :user_to_index
 :item_to_index
```

```
In [17]: # Format je (korisnik, film, ocjena)
rating_set.training_set
```

```
Out[17]: 900188-element Array{IncrementalSVD.Rating,1}:
 IncrementalSVD.Rating(3718,1914,2.0)
 IncrementalSVD.Rating(2078,526,5.0)
 IncrementalSVD.Rating(2919,284,4.0)
 IncrementalSVD.Rating(3829,364,3.0)
 IncrementalSVD.Rating(4887,109,3.0)
 IncrementalSVD.Rating(4892,377,5.0)
 IncrementalSVD.Rating(4167,1208,2.0)
 IncrementalSVD.Rating(3868,559,3.0)
 IncrementalSVD.Rating(5455,920,3.0)
 IncrementalSVD.Rating(1999,295,5.0)
 IncrementalSVD.Rating(6016,490,2.0)
 IncrementalSVD.Rating(899,19,5.0)
 IncrementalSVD.Rating(4490,1242,5.0)
 ⋮
 IncrementalSVD.Rating(5776,79,4.0)
 IncrementalSVD.Rating(1425,2435,3.0)
 IncrementalSVD.Rating(2624,23,4.0)
```

```
IncrementalSVD.Rating(1146,86,2.0)
IncrementalSVD.Rating(4715,669,4.0)
IncrementalSVD.Rating(3546,1356,3.0)
IncrementalSVD.Rating(3576,1579,3.0)
IncrementalSVD.Rating(3792,651,4.0)
IncrementalSVD.Rating(1883,393,3.0)
IncrementalSVD.Rating(2353,940,2.0)
IncrementalSVD.Rating(5035,427,5.0)
IncrementalSVD.Rating(2122,2302,5.0)
```

```
In [18]: rating_set.test_set
```

```
Out[18]: 100021-element Array{IncrementalSVD.Rating,1}:
IncrementalSVD.Rating(3272,157,3.0)
IncrementalSVD.Rating(883,128,5.0)
IncrementalSVD.Rating(840,105,4.0)
IncrementalSVD.Rating(2402,1107,2.0)
IncrementalSVD.Rating(5364,1299,5.0)
IncrementalSVD.Rating(4448,774,4.0)
IncrementalSVD.Rating(678,995,4.0)
IncrementalSVD.Rating(3684,1388,3.0)
IncrementalSVD.Rating(149,406,4.0)
IncrementalSVD.Rating(4854,27,4.0)
IncrementalSVD.Rating(3821,8,2.0)
IncrementalSVD.Rating(1752,185,5.0)
IncrementalSVD.Rating(2015,237,4.0)
⋮
IncrementalSVD.Rating(3557,1364,1.0)
IncrementalSVD.Rating(1232,742,4.0)
IncrementalSVD.Rating(2860,429,3.0)
IncrementalSVD.Rating(1266,3121,3.0)
IncrementalSVD.Rating(4050,1498,4.0)
IncrementalSVD.Rating(3669,464,4.0)
IncrementalSVD.Rating(1745,1071,4.0)
IncrementalSVD.Rating(3754,2871,2.0)
IncrementalSVD.Rating(4072,1477,4.0)
IncrementalSVD.Rating(687,1086,1.0)
IncrementalSVD.Rating(934,1409,2.0)
IncrementalSVD.Rating(1150,454,3.0)
```

```
In [19]: # Korisnici i njihove šifre
rating_set.user_to_index
```

```
Out[19]: Dict{AbstractString,Int32} with 6040 entries:
"4304" => 4304
"3935" => 3935
"5422" => 5422
"5734" => 5734
```



```

"2243" => 2243
"1881" => 1881
"5425" => 5425
"4209" => 4209
"1907" => 1907
"2923" => 2923
"599" => 599
"2491" => 2491
"5944" => 5944
"228" => 228
"2590" => 2590
"3697" => 3697
"5031" => 5031
"2579" => 2579
"5551" => 5551
"1880" => 1880
"2562" => 2562
"3215" => 3215
"3991" => 3991
"4652" => 4652
"4088" => 4088
:      => :

```

```

In [20]: # Filmovi i njihove šifre
rating_set.item_to_index

```

```

Out[20]: Dict{AbstractString,Int32} with 3706 entries:

```

```

"Fried Green Tomatoes (1991)"
"Milk Money (1994)"
"From Russia with Love (1963)"
"House II: The Second Story (1987)"
"Held Up (2000)"
"Missing in Action 2: The Beginning (1985)"
"Murder, My Sweet (1944)"
"Hidden, The (1987)"
"Cable Guy, The (1996)"
"Big Kahuna, The (2000)"
"Addams Family Values (1993)"
"Farinelli: il castrato (1994)"
"Education of Little Tree, The (1997)"
"In God's Hands (1998)"
"Last Man Standing (1996)"
"Sixth Sense, The (1999)"
"Star Maps (1997)"
"Girl, Interrupted (1999)"
"Stand by Me (1986)"
"Rob Roy (1995)"
"Caligula (1980)"

```

```

=> 5
=> 1
=> 7
=> 1
=> 3
=> 2
=> 9
=> 9
=> 6
=> 8
=> 1
=> 1
=> 2
=> 2
=> 1
=> 3
=> 2
=> 3
=> 8
=> 4
=> 2

```

```

"Flirting With Disaster (1996)" => 1
"Hook (1991)" => 6
"Institute Benjamenta, or This Dream People Call Human Life (1995)" => 3
"Way We Were, The (1973)" => 7
: => :

```

```

In [21]: # Možemo posebno izvaditi naslove ...
         keys(rating_set.item_to_index)

```

```

Out[21]: Base.KeyIterator for a Dict{AbstractString,Int32} with 3706 entries. Keys:
"Fried Green Tomatoes (1991)"
"Milk Money (1994)"
"From Russia with Love (1963)"
"House II: The Second Story (1987)"
"Held Up (2000)"
"Missing in Action 2: The Beginning (1985)"
"Murder, My Sweet (1944)"
"Hidden, The (1987)"
"Cable Guy, The (1996)"
"Big Kahuna, The (2000)"
"Addams Family Values (1993)"
"Farinelli: il castrato (1994)"
"Education of Little Tree, The (1997)"
"In God's Hands (1998)"
"Last Man Standing (1996)"
"Sixth Sense, The (1999)"
"Star Maps (1997)"
"Girl, Interrupted (1999)"
"Stand by Me (1986)"
"Rob Roy (1995)"
"Caligula (1980)"
"Flirting With Disaster (1996)"
"Hook (1991)"
"Institute Benjamenta, or This Dream People Call Human Life (1995)"
:

```

```

In [22]: # i šifre
         values(rating_set.item_to_index)

```

```

Out[22]: Base.ValueIterator for a Dict{AbstractString,Int32} with 3706 entries. Val
594
1361
729
1247
3549
2177
996
981

```

```

669
893
1857
1945
2759
2814
1840
39
2670
32
85
449
2532
1290
657
3397
:

```

```

In [23]: # Koje je filmove ocijenio korisnik "3000"
         user_ratings(rating_set, "3000")

```

```

Out[23]: 96-element Array{Tuple{SubString{String},Float32},1}:
 ("American Beauty (1999)",5.0)
 ("Dances with Wolves (1990)",5.0)
 ("Brazil (1985)",5.0)
 ("Babe (1995)",5.0)
 ("Caddyshack (1980)",5.0)
 ("Gattaca (1997)",5.0)
 ("Brothers McMullen, The (1995)",5.0)
 ("When Harry Met Sally... (1989)",5.0)
 ("One Flew Over the Cuckoo's Nest (1975)",5.0)
 ("Twelve Monkeys (1995)",5.0)
 ("Time Bandits (1981)",5.0)
 ("Princess Bride, The (1987)",5.0)
 ("Chinatown (1974)",4.0)
 :
 ("2001: A Space Odyssey (1968)",2.0)
 ("Romancing the Stone (1984)",2.0)
 ("Mission: Impossible (1996)",1.0)
 ("Blind Date (1987)",1.0)
 ("Star Trek VI: The Undiscovered Country (1991)",1.0)
 ("Sphere (1998)",1.0)
 ("Sudden Death (1995)",1.0)
 ("Blue Lagoon, The (1980)",1.0)
 ("Grease 2 (1982)",1.0)
 ("Dirty Dancing (1987)",1.0)
 ("Jewel of the Nile, The (1985)",1.0)
 ("Flashdance (1983)",1.0)

```

```
In [24]: # Nađimo točan naziv i šifru za "Blade runner"
        for k in keys(rating_set.item_to_index)
            if contains(k, "Blade")
                println(k)
            end
        end
```

```
Sling Blade (1996)
Blade (1998)
Blade Runner (1982)
Some Folks Call It a Sling Blade (1993)
```

```
LoadError: UnicodeError: invalid character index
while loading In[24], in expression starting on line 2
```

```
in slow_utf8_next(::Array{UInt8,1}, ::UInt8, ::Int64) at ./strings/string.jl:12
in next at ./strings/string.jl:92 [inlined]
in next(::SubString{String}, ::Int64) at ./strings/types.jl:51
in search(::SubString{String}, ::Char, ::Int64) at ./strings/search.jl:12
in _searchindex(::SubString{String}, ::String, ::Int64) at ./strings/search.jl:30
in contains(::SubString{String}, ::String) at ./strings/search.jl:310
in macro expansion; at ./In[24]:3 [inlined]
in anonymous at ./<missing>:?
```

```
In [25]: get(rating_set.item_to_index, "Blade Runner (1982)", 0)
```

```
Out[25]: 744
```

```
In [26]: # Da li je korisnik "3000" ocijenio "Blade Runner" ?
        for k in user_ratings(rating_set, "3000")
            if contains(k[1], "Blade")
                println(k)
            end
        end
```

```
("Blade Runner (1982)", 4.0f0)
```

```
In [27]: # Da li je korisnik "3000" ocijenio "Citizen Kane" ?
        for k in user_ratings(rating_set, "3000")
            if contains(k[1], "Citizen")
                println(k)
            end
        end
```

```
In [28]: # Ovo traje dvije i po minute
        model = train(rating_set, 25);
```

Computing truncated rank 25 SVD 100% Time: 0:02:14

```
In [29]: fieldnames(model)
```

```
Out[29]: 5-element Array{Symbol,1}:
          :user_to_index
          :item_to_index
          :U
          :S
          :V
```

```
In [30]: model.U
```

```
Out[30]: 6040×25 Array{Float32,2}:
  0.0105286  0.0108881  0.0169462  0.00726165  ...  0.0165685  0.010
  0.0143127  0.0107969  0.00958497  0.0103498    0.00717566  0.01500
  0.0101161  0.00902197  0.0170195  0.00770462    0.0117291  0.01570
  0.00710551  0.00677834  0.0158305  0.00420549    0.0114087  0.01115
  0.0136108  0.0157925  0.00125129  0.0154513    0.0038251  0.00905
  0.0108928  0.0173633  0.0154919  0.0163003    ...  0.00889664  0.015
  0.00870144  0.00751549  0.0187289  0.00401489    0.0107074  0.01189
  0.0156788  0.0149871  0.00987733  0.0133478    0.0110488  0.01029
  0.0134843  0.0083181  0.010523  0.00702826    0.0111521  0.00816
  0.0188921  0.0123215  -0.0108931  0.0242425    0.00759919  0.01249
  0.01271  0.0114213  0.0110434  0.0103405    ...  0.011292  0.003
  0.00605743  0.00804548  0.0138675  0.00590076    0.0132224  0.01658
  0.0118863  0.00952506  0.0131845  0.00718291    0.00522086  0.01743
  ⋮
  0.00637816  0.0129909  0.0137177  0.0116263    0.0185623  0.01795
  0.0148243  0.0120957  0.0156166  0.0100511    0.0130567  0.00260
  0.00930082  0.0110824  0.0166959  0.0108397    ...  0.00787297  0.021
  0.0152003  0.0130907  0.0121974  0.00994505    0.00859643  0.01103
  0.0104563  0.00985341  0.0170671  0.00735145    0.0129723  0.01212
  0.00540673  0.011245  0.012312  0.00927308    0.0118498  0.01943
  0.0139161  0.007034  0.0044327  0.00543997    -0.0162419  0.01604
  0.0169726  0.0150091  -0.0224702  0.0236435    ...  -0.0177601  0.018
  0.0153233  0.0107181  -0.000431351  0.0124533    0.00457488  0.02171
  0.00511705  0.00903653  0.011796  0.00764805    0.0128089  0.02070
```

0.0145618	0.0175328	0.00831574	0.0117466	0.00398802	0.01014
0.0162319	0.0162784	-0.0097728	0.0188398	0.00443158	0.00456

In [31]: model.S

Out [31]: 25-element Array{Float32,1}:

8191.46
2145.75
823.595
1152.41
852.304
1202.16
638.137
404.692
331.272
254.959
260.26
346.955
284.287
226.332
258.695
315.226
341.985
192.93
205.88
195.595
171.799
140.685
136.543
115.004
97.6223

In [32]: model.V

Out [32]: 3706×25 Array{Float32,2}:

0.0357308	0.00829067	0.0684244	...	-0.00427573	0.00211629
0.025173	0.0150898	0.0205194		-0.00464182	0.00882804
0.0305362	0.0132749	0.0311393		-0.0013124	0.00987835
0.0320461	0.0118308	0.0538765		0.0193329	-0.02097
0.0315251	0.00753399	0.0547098		0.0118283	-0.0126033
0.0345476	0.00460288	0.0685182	...	-0.00493932	-0.00594212
0.030664	0.00816449	0.029397		0.000558795	0.006811
0.0331358	0.00385654	0.0518303		0.0315574	-0.00627985
0.0288807	0.00781605	0.0234186		0.00174335	0.0199391
0.0334986	0.00535404	0.0540243		-0.0167033	0.0369555
0.0296428	0.00542992	0.0376194	...	0.0203523	-0.0223229
0.0177424	0.0266619	0.0091863		0.00665131	0.0133215
0.025885	0.0189131	0.0132219		0.00282401	0.00724893
⋮			...		

0.00118762	0.00260339	0.00408728		0.013889	0.0148223
0.00113699	0.00239618	0.0038666	...	0.0123599	0.0132924
0.00112264	0.00224237	0.00369467		0.0102801	0.0110935
0.00114922	0.00253774	0.00402916		0.0143987	0.0154508
0.00118594	0.00257986	0.00388293		0.0104795	0.0120503
0.00124809	0.00245462	0.0034857		0.0131468	0.010036
0.00122855	0.00265491	0.00408049	...	0.0132781	0.0140098
0.00139119	0.00293329	0.00406722		0.0165901	0.0167884
0.00125033	0.00247112	0.00351265		0.0103195	0.0121719
0.00115477	0.00228725	0.00366777		0.010013	0.0110297
0.00125898	0.00277488	0.00406416		0.0130792	0.0132153
0.00128577	0.00264787	0.00386465	...	0.0126263	0.0133414

```
In [33]: similar_items(model, "Friday the 13th (1980)")
```

```
Out[33]: 10-element Array{SubString{String},1}:
"Friday the 13th (1980)"
"Amityville Horror, The (1979)"
"Jaws 2 (1978)"
"Friday the 13th Part 2 (1981)"
"Omen, The (1976)"
"Pet Sematary (1989)"
"Nightmare on Elm Street Part 2: Freddy's Revenge, A (1985)"
"Cujo (1983)"
"Candyman (1992)"
"Freddy's Dead: The Final Nightmare (1991)"
```

```
In [34]: @which similar_items(model, "Friday the 13th (1980)")
```

```
Out[34]: similar_items(model::IncrementalSVD.RatingsModel, item) at /home/slap/.julia
```

```
In [35]: similar_items(model, "Citizen Kane (1941)")
```

```
Out[35]: 10-element Array{SubString{String},1}:
"Citizen Kane (1941)"
"M*A*S*H (1970)"
"Dr. Strangelove or: How I Learned to Stop Worrying and Love the Bomb (1964)"
"Chinatown (1974)"
"Rear Window (1954)"
"Vertigo (1958)"
"Lawrence of Arabia (1962)"
"Casablanca (1942)"
"Boat, The (Das Boot) (1981)"
"Apocalypse Now (1979)"
```

```
In [36]: similar_users(model, "3000")
```

```
Out[36]: 10-element Array{SubString{String},1}:
"3000"
```

```
"6030"  
"5619"  
"5457"  
"1360"  
"1817"  
"1248"  
"5572"  
"514"  
"5368"
```

```
In [37]: # Stvarna ocjena je bila 4.0  
         get_predicted_rating(model, "3000", "Blade Runner (1982)")
```

```
Out[37]: 4.2329783f0
```

```
In [38]: IncrementalSVD.get_predicted_rating(model, "3000", "Citizen Kane (1941)")
```

```
Out[38]: 4.236797f0
```

11 Tehnologija - github

Open Source!

[github](#) je sustav za kolaborativan razvoj softvera - izlog!

12 Julia i Jupyter

[Julia](#) je novi programski jezik (evolucija) za BigData

- [JuliaBox](#)

[Jupyter](#) bilježnice:

- tekst i programi u 40 jezika (Julia, Python, R)
- HTML, knjiga (LaTeX, PDF)
- prezentacija, nbgrader

13 Hvala na pažnji

Pitanja?

```
In [ ]:
```