Matrix factorization

Nicolás Torres

IIC 3633 - Sistemas Recomendadores
Pontificia Universidad Católica de Chile

24 de septiembre de 2015

Bibliografía

Esta presentación se basa en las siguientes publicaciones

Xia Ning and George Karypis.
Sparse linear methods with side information for top-n recommendations.
In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys '12, pages 155–162. ACM, 2012.

S. Zhang, W. Wang, J. Ford, and F. Makedon.
Learning from incomplete ratings using non-negative matrix factorization.
In *Proceedings of the 6TH SIAM Conference on Data Mining*, pages 549–553, 2006.

Tabla de Contenidos

NMF

La matriz de ratings $V : m \times n$, se descompone en dos matrices de menor rango, no negativas de la forma

$$V \approx WH, \tag{1}$$

donde $W : n \times k$ y $H : k \times m$. En la práctica, el rango de factorización es escogido, tal que, $k \ll \min(m, n)$.

NMF

La matriz V se obtiene encontrando las matrices no-negativas $W(n \times k)$ y $H(k \times m)$ que optimizan

$$\min ||A - WH||_F^2 \tag{2}$$

$$\sim \max \log Pr(A|V) \tag{3}$$

Este problema puede ser resuelto utilizando el método de los multiplicadores de Lagrange.

Learning From Incomplete Ratings

Cuando la matriz de ratings está incompleta se debe optimizar el problema

$$\max \log Pr(A^{\circ}|V), \tag{4}$$

donde A° son los datos observados.

- Se proponen dos algoritmos para maximizar esta función objetivo
  - EM procedure
  - Weighted NMF

Expectation-Maximization (EM)

El algoritmo EM se usa para encontrar estimadores de máxima verosimilitud de parámetros en modelos donde los datos están incompletos.

- Expectation
- Maximization

Expectation

Calcula la expresión esperada para la máxima verosimilitud de V con respecto a datos desconocidos (A^u) dado un conjunto de datos observados (A°) y un parámetro estimado $V^{(t-1)}$, esto es

$$Q(V, V^{(t-1)}) = E[\log Pr(A^\circ, A^u|V)|A^\circ, V^{(t-1)}] \tag{5}$$

Maximization

Encuentra el parámetro $V^{(t)}$ que maximiza la esperanza calculada anteriormente $Q(V, V^{(t-1)})$, esto es

$$V^{(t)} = \text{arg}_v \max Q(V, V^{(t-1)}) \tag{6}$$

WNMF

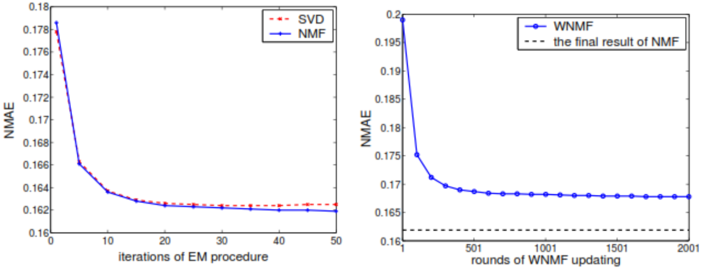
Es una variación del anterior

$$\min \sum_{ij} P_{ij}(A_{ij} - (WH)_{ij})^2, \tag{7}$$

donde P_{ij} es igual a 1 si A_{ij} es un dato observado y 0 en otro caso.

EM procedure vs. WNMF

- EM procedure converge bien empiricamente y es menos susceptible a las condiciones iniciales.
- WNMF es mucho más eficiente computacionalmente.
- Hybrid NMF puede ser más efectivo.



a EM   b WNMF

Figura 1: NMAE of the EM and WNMF approaches on MovieLens.

La Tabla 1 muestra los resultados de los algoritmos basados en NMF sobre el data set de MovieLens.

	Pearson	SVD EM	NMF EM	Hybrid NMF
NMAE	0.1707	0.1629	0.1623	0.1634
ROC-4	0.7471	0.7682	0.7723	0.7691

Tabla 1: Desempeño de los algoritmos en MovieLens

Experimentos

La Tabla 2 muestra los resultados de los algoritmos basados en NMF sobre el data set de Jester.

	Pearson	SVD EM	NMF EM	Hybrid NMF
NMAE	0.1634	0.1605	0.1599	0.1599
ROC-4	0.7539	0.7588	0.7612	0.7608

Tabla 2: Desempeño de los algoritmos en Jester

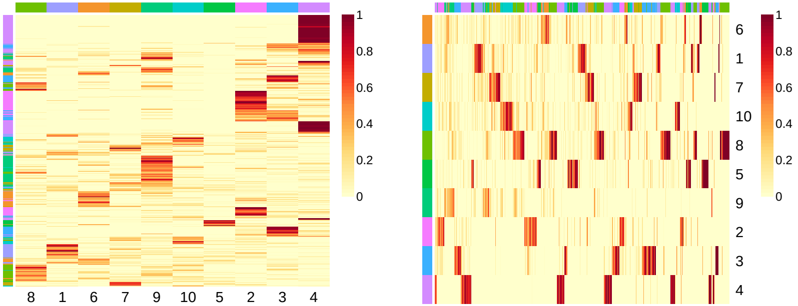
Ejemplo

Demostración de la factorización NMF en R.

```
install.packages("NMF")
library(NMF)
data(MovieLense)
NMF <- nmf(MovieLense, 10)
W <- basis(NMF)
H <- coef(NMF)
basismap(NMF)
coefmap(NMF)
```

Ejemplo

La Figura 2 exhibe las matrices W y H originadas de la descomposición NMF de rango 10, sobre el data set de MovieLens 100k.



a Matriz W

b Matriz H

Figura 2: Matrices W y H de la descomposición NMF (rango 10)

Ejemplo

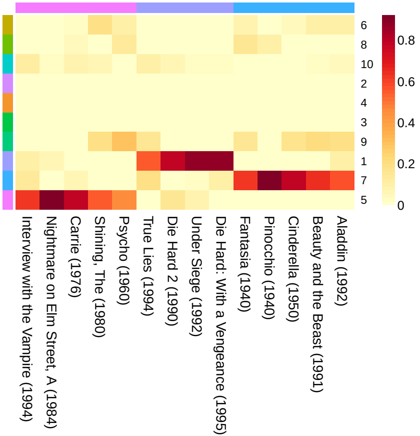


Figura 3: Items más representativos de las comunidades 1, 5 y 7

SLIM

Sparse LInear Method (SLIM) [Ning and Karypis(2012)].

- SLIM se centra en recomendación top N.
- Capaz de realizar recomendaciones de alta calidad y con gran rapidez.

SLIM

El modelo utilizado por SLIM puede ser presentado como

$$\tilde{A} = AW, \tag{8}$$

donde $A : m \times n$ es la matriz de ratings y $W : n \times n$ es una matriz sparse de coeficientes.

SLIM

La matriz W se obtiene minimizando el siguiente problema de optimización regularizado:

$$\min_{W} \frac{1}{2}||A - AW||_F^2 + \frac{\beta}{2}||W||_F^2 + \lambda||W||_1 \tag{9}$$

sujeto a $W \geq 0$, $\text{diag}(W) = 0$.

Computing W

Como las columnas de W son independientes, el problema de optimización se puede descomponer en el conjunto

$$\min_{w_j} \frac{1}{2}||a_j - Aw_j||_2^2 + \frac{\beta}{2}||w_j||_2^2 + \lambda||w_j||_1 \tag{10}$$

sujeto a $w_j \geq 0$, $w_{j,j} = 0$.

SLIM with Feature Selection

`fsSLIM` Antes de calcular w_j se pueden usar métodos de selección de características para reducir el número de variables independientes. Esto disminuye sustancialmente el tiempo de construcción del modelo.

EXPERIMENTOS

dataset	#users	#items	#trns	rsize	csize	density	ratings
ccard	42,067	18,004	308,420	7.33	17.13	0.04%	-
ctlg2	22,505	17,096	1,814,072	80.61	106.11	0.47%	-
ctlg3	58,565	37,841	453,219	7.74	11.98	0.02%	-
ecmrc	6,594	3,972	50,372	7.64	12.68	0.19%	-
BX	3,586	7,602	84,981	23.70	11.18	0.31%	1-10
ML10M	69,878	10,677	10,000,054	143.11	936.60	1.34%	1-10
Netflix	39,884	8,478	1,256,115	31.49	148.16	0.37%	1-5
Yahoo	85,325	55,371	3,973,104	46.56	71.75	0.08%	1-5

Figura 4: The Datasets Used in Evaluation.

Hit Rate (HR) y Average Reciprocal Hit-Rank (ARHR)

$$\text{HR} = \frac{\texttt{\#hits}}{\texttt{\#users}} \tag{11}$$

$$\text{ARHR} = \frac{1}{\texttt{\#users}} \sum_{i=1}^{\texttt{\#hits}} \frac{1}{p_i} \tag{12}$$

donde `#hits: TP`

Top-N Recommendation Performance

method	ccard params		HR	ARHR	mt	tt	ctlg2 params		HR	ARHR	mt	tt
itemkNN	50	-	0.195	0.145	0.54(s)	1.34(s)	10	-	0.222	0.108	33.78(s)	1.19(s)
itemprob	50	0.2	0.226	0.154	0.97(s)	1.24(s)	10	0.5	0.222	0.105	47.86(s)	0.99(s)
userkNN	150	-	0.189	0.122	0.06(s)	14.84(s)	50	-	0.204	0.106	0.37(s)	48.45(s)
PureSVD	3500	10	0.101	0.058	42.89(m)	2.65(h)	1300	10	0.196	0.099	3.95(m)	18.46(m)
WRMF	250	15	0.230	0.150	4.01(h)	9.14(m)	300	10	0.235	0.114	20.42(h)	5.49(s)
BPRMF	350	0.3	0.238	0.157	1.29(h)	6.64(m)	400	0.1	0.249	0.123	9.72(h)	3.14(m)
BPRkNN	1e-4	0.01	0.208	0.145	2.38(m)	8.15(m)	0.001	0.001	0.224	0.104	1.28(h)	22.03(m)
SLIM	5	0.5	**0.246**	0.170	17.24(m)	13.57(s)	5	2.0	**0.272**	0.140	7.24(h)	26.98(s)
fsSLIM	100	0.5	0.243	0.168	4.97(m)	4.45(s)	100	1.0	**0.282**	0.149	10.92(m)	5.00(s)
fsSLIM	50	0.5	0.244	0.169	2.40(m)	3.34(s)	10	0.5	0.262	0.138	4.21(m)	2.01(s)

method	ctlg3 params		HR	ARHR	mt	tt	ecmrc params		HR	ARHR	mt	tt
itemkNN	300	-	0.544	0.313	0.55(s)	6.66(s)	300	-	0.218	0.125	0.06(s)	0.54(s)
itemprob	400	0.3	0.558	0.322	0.87(s)	7.62(s)	30	0.2	0.245	0.138	0.09(s)	0.12(s)
userkNN	350	-	0.492	0.285	0.11(s)	19.18(s)	400	-	0.212	0.119	0.01(s)	0.78(s)
PureSVD	3000	10	0.373	0.210	1.11(h)	4.28(h)	1900	10	0.186	0.110	3.67(m)	3.22(m)
WRMF	420	20	0.543	0.308	14.42(h)	50.67(m)	270	15	0.242	0.133	3.22(h)	13.60(s)
BPRMF	300	0.5	0.541	0.283	1.49(h)	13.66(m)	350	0.1	0.249	0.128	4.00(m)	12.76(s)
BPRkNN	0.001	1e-4	0.542	0.304	6.20(m)	20.28(m)	1e-5	0.010	0.242	0.130	1.02(m)	13.53(s)
SLIM	3	0.5	**0.579**	0.347	1.02(h)	16.23(s)	5	0.5	**0.255**	0.149	11.10(s)	0.51(s)
fsSLIM	100	0.0	0.546	0.292	12.57(m)	9.62(s)	100	0.5	0.252	0.147	16.89(s)	0.32(s)
fsSLIM	400	0.5	0.570	0.339	14.27(m)	12.52(s)	30	0.5	0.252	0.147	5.41(s)	0.16(s)

Figura 5: Comparison of Top-N Recommendation Algorithms.

Top-N Recommendation Performance

method	BX						ML10M					
	params		HR	ARHR	mt	tt	params		HR	ARHR	mt	tt
itemkNN	10	-	0.085	0.044	1.34(s)	0.08(s)	20	-	0.238	0.106	1.97(m)	8.93(s)
itemprob	30	0.3	0.103	0.050	2.11(s)	0.22(s)	20	0.5	0.237	0.106	1.88(m)	7.49(s)
userkNN	100	-	0.083	0.039	0.01(s)	1.49(s)	50	-	0.303	0.146	2.26(s)	34.42(m)
PureSVD	1500	10	0.072	0.037	1.91(m)	2.57(m)	170	10	0.294	0.139	1.68(m)	1.72(m)
WRMF	400	5	0.086	0.040	12.01(h)	29.77(s)	100	2	0.306	0.139	16.27(h)	1.59(m)
BPRMF	350	0.1	0.089	0.040	8.95(m)	12.44(s)	350	0.1	0.281	0.123	4.77(h)	5.20(m)
BPRkNN	1e-4	0.010	0.082	0.035	5.16(m)	42.23(s)	0.001	1e-4	**0.327**	0.156	15.78(h)	1.08(h)
SLIM	3	0.5	**0.109**	0.055	5.51(m)	1.39(s)	1	2.0	0.311	0.153	50.98(h)	41.59(s)
fsSLIM	100	0.5	0.109	0.053	36.26(s)	0.63(s)	100	0.5	0.311	0.152	37.12(m)	17.97(s)
fsSLIM	30	1.0	0.105	0.055	16.07(s)	0.18(s)	20	1.0	0.298	0.145	14.26(m)	8.87(s)

method	Netflix						Yahoo					
	params		HR	ARHR	mt	tt	params		HR	ARHR	mt	tt
itemkNN	150	-	0.178	0.088	24.53(s)	13.17(s)	400	-	0.107	0.041	21.54(s)	2.25(m)
itemprob	10	0.5	0.177	0.083	30.36(s)	1.01(s)	350	0.5	0.107	0.041	34.23(s)	1.90(m)
userkNN	200	-	0.154	0.077	0.33(s)	1.04(m)	50	-	0.107	0.041	18.46(s)	3.26(m)
PureSVD	3500	10	0.182	0.092	29.86(m)	21.29(m)	170	10	0.074	0.027	53.05(s)	11.18(m)
WRMF	350	10	0.184	0.085	22.47(h)	2.63(m)	200	8	0.090	0.032	16.23(h)	50.05(m)
BPRMF	400	0.1	0.156	0.071	43.55(m)	3.56(m)	400	0.1	0.093	0.033	10.36(h)	47.28(m)
BPRkNN	0.01	0.01	0.188	0.092	10.91(m)	6.12(m)	0.01	0.001	0.104	0.038	2.60(h)	4.11(h)
SLIM	5	1.0	**0.200**	0.102	7.85(h)	9.84(s)	5	0.5	**0.122**	0.047	21.30(h)	5.69(m)
fsSLIM	100	0.5	**0.202**	0.104	6.43(m)	5.73(s)	100	0.5	**0.124**	0.048	1.39(m)	41.24(s)
fsSLIM	150	0.5	**0.202**	0.104	9.09(m)	7.47(s)	400	0.5	**0.123**	0.048	2.41(m)	1.72(m)

Figura 6: Comparison of Top-N Recommendation Algorithms.

SLIM for the Long-Tail Distribution

method	ML10M long tail					
	params		HR	ARHR	mt	tt
itemkNN	10	-	0.130	0.052	1.59(m)	4.62(s)
itemprob	10	0.5	0.126	0.051	1.65(m)	4.04(s)
userkNN	50	-	0.162	0.069	2.10(s)	20.43(m)
PureSVD	350	70	0.224	0.096	2.98(m)	10.45(m)
WRMF	100	2	0.232	0.097	23.15(h)	1.74(m)
BPRMF	300	0.01	0.240	0.102	22.63(h)	8.56(m)
BPRkNN	0.001	1e-4	0.239	0.098	15.72(h)	36.42(m)
SLIM	1	5.0	**0.256**	0.106	57.55(h)	47.69(s)
fsSLIM	10	5.0	0.255	0.105	25.37(m)	9.57(s)
fsSLIM	100	4.0	0.255	0.105	58.32(m)	19.32(s)

Figura 7: Performance on the Long Tail of ML10M. 1 % most popular items are eliminated from ML10M. Params have same meanings as those in Figura 6.

Recommendation for Different Top-N

dataset	N				
	5	10	15	20	25
BX	0.012	0.006	0.000	0.000	0.001
ML10M	0.000	-0.016	-0.013	-0.018	-0.021
Netflix	0.013	0.012	0.008	0.005	0.003
Yahoo	0.009	0.015	0.015	0.016	0.017

Figura 8: Performance Difference on Top-N Recommendations. Columns corresponding to N shows the performance (in terms of HR) difference between SLIM and the best of the rest methods on corresponding top-N recommendatons.