A Nonnegative Matrix factorization Approach for recommender systems

Gianna M. Del Corso   Francesco Romani

Dipartimento di Informatica, Università di Pisa, Italy

Introduction

- Retailers propose to consumers many products and choices
- Help users to find items meeting testes and needs
- Recommender systems: algorithms for recommending items of interest for users

- Content-based
- Collaborative filtering
  - Neighborhood methods: Suggest to a user items similar to those she liked in the past
  - Latent factor models: Explain the ratings by characterizing both items and users on a few factors inferred from the ratings patterns

Introduction

- Retailers propose to consumers many products and choices
- Help users to find items meeting testes and needs
- Recommender systems: algorithms for recommending items of interest for users

- Content-based
- Collaborative filtering
  - Neighborhood methods: Suggest to a user items similar to those she liked in the past
  - Latent factor models: Explain the ratings by characterizing both items and users on a few factors inferred from the ratings patterns

Introduction

- Retailers propose to consumers many products and choices
- Help users to find items meeting testes and needs
- Recommender systems: algorithms for recommending items of interest for users

- Content-based
- Collaborative filtering
  - Neighborhood methods: Suggest to a user items similar to those she liked in the past

  - Latent factor models: Explain the ratings by characterizing both items and users on a few factors inferred from the ratings patterns

Introduction

- Retailers propose to consumers many products and choices
- Help users to find items meeting testes and needs
- Recommender systems: algorithms for recommending items of interest for users

- Content-based
- Collaborative filtering
  - Neighborhood methods: Suggest to a user items similar to those she liked in the past
  - Latent factor models: Explain the ratings by characterizing both items and users on a few factors inferred from the ratings patterns

Latent factor models

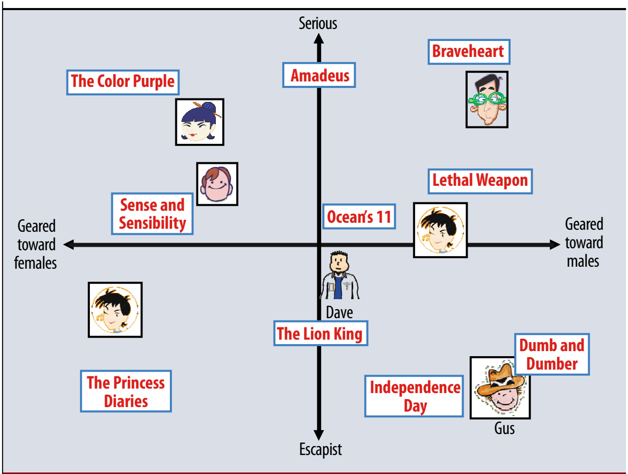


Figure 2. A simplified illustration of the latent factor approach, which characterizes both users and movies using two axes—male versus female and serious versus escapist.

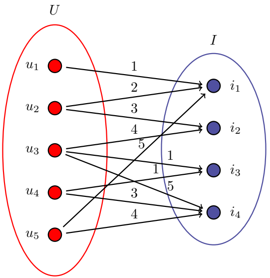
The model

A dataset of recommendation can be viewed as a weighted bipartite graph:

$$G = (U \cup I, \Omega),$$

- U set of users, I set of items, $\Omega \subseteq U \times I$.
- $V = \{1, 2, \ldots, v\}$ set of possible votes
- $V_0 = V \cup ?$

The model



Associate the Utility matrix

$$A = \begin{bmatrix} 1 & ? & ? & ? \\ 2 & 3 & ? & ? \\ ? & 4 & 1 & 5 \\ ? & ? & 1 & 3 \\ 5 & ? & ? & 4 \end{bmatrix}$$

The goal of a recommender system is to predict some of the ? to make personalized recommendations

The latent factor models

- We assume the expressed ratings as characterized by a low number of latent factors.
- For example in movies: genres, actors, directors
- Low-rank approximation of the utility matrix
- The model is trained using the available ratings and used to predict new ratings.

Nonnegative Matrix Factorization approach

- Define the projector $\mathcal{P}_\Omega$ as

$$\mathcal{P}_\Omega(M) = \begin{cases} m_{ij} \text{ if } (i,j) \in \Omega \\ 0 \quad \text{otherwise} \end{cases}$$

- Given $k \ll \min(n, m)$, find W and H such that

$$\min_{W,H} \|\mathcal{P}_\Omega(A - WH^T)\|_F^2, \quad W \in \mathbb{R}_+^{n\times k}, H \in \mathbb{R}_+^{m\times k}, WH^T \leq v.$$

- User u is represented by row vector $\mathbf{w}_u$, while $\mathbf{h}_i$ represents item i.
- k represents the number of "latent factors" of our data.

Nonnegative Matrix Factorization approach

$$\min_{W,H} \frac{1}{2}\|A - WH^T\|_F^2, \quad W \in \mathbb{R}_+^{n\times k}, H \in \mathbb{R}_+^{m\times k}$$

- This problem is non convex $\rightarrow$ many local minima
- The problem is convex in either one of the two matrices.

$$\min_{W\in\mathbb{R}_+^{n\times k}} \frac{1}{2}\|A - WH^T\|_F^2, \quad \min_{H\in\mathbb{R}_+^{m\times k}} \frac{1}{2}\|A - WH^T\|_F^2$$

Nonnegative Matrix Factorization approach

$$H_{i+1} = \text{argmin}_{X \geq 0} \|A - W_i X\|_F^2$$

$$W_{i+1} = \text{argmin}_{Y \geq 0} \|A - Y H_{i+1}^T\|_F^2,$$

- The Alternating Nonnegative Least Square is a very successful class of algorithms.
- Every limit point generated from the ANLS framework is a stationary point for the non-convex original problem.
- Many methods to solve the least square problems in ANLS: Active set, projected gradient, projected quasi-Newton, greedy coordinate descent.

Recommender system with NMF

- In our case A is not completely known (only a few ratings)
- This problem is not convex too!
- We can proceed similarly to the NMF
- Many attempts in this direction: regularization techniques to avoid overfitting, addition of a prior factor to avoid overgrow of W and H, etc

Our Approach

- Adaptively we update the Utility matrix on the ? with the current values of $W_i H_i^T$
- Regularization is performed with a cut to v of the entries of $W_i H_i^T$
- Stopping criteria to avoid stagnation.

Unfortunately no theoretical results on the convergence!

Our Approach

- Adaptively we update the Utility matrix on the ? with the current values of $W_i H_i^T$
- Regularization is performed with a cut to v of the entries of $W_i H_i^T$
- Stopping criteria to avoid stagnation.

Unfortunately no theoretical results on the convergence!

Our Approach

- Adaptively we update the Utility matrix on the ? with the current values of $W_i H_i^T$
- Regularization is performed with a cut to v of the entries of $W_i H_i^T$
- Stopping criteria to avoid stagnation.

Unfortunately no theoretical results on the convergence!

The algorithm

$C_0 = \mathcal{P}_\Omega(A)$, $W_0 = \mathrm{rand}(m,k)$, $H_0 = 0$

flag=TRUE, mFE $= 0$, $j = 0$

while($j < j_{\max}$ & flag)

$j++$;

$\mathrm{mFE}_{\mathrm{old}} \leftarrow \mathrm{mFE}$

$H_j = \mathrm{argmin}_{X \geq 0} \|C_{j-1} - W_{j-1}X^T\|_F^2$

$W_j = \mathrm{argmin}_{Y \geq 0} \|C_{j-1} - YH_j^T\|_F^2$

$R \leftarrow \mathrm{cut}_v([W_j H_j^T])$

$C_j \leftarrow \mathcal{P}_\Omega(A) + \mathcal{P}_{\bar{\Omega}}(R)$

$\mathrm{MIE} = \max_\Omega |A - R|$ Maximum Integer error

$\mathrm{mFE} = \frac{\|\mathcal{P}_\Omega(A - W_j H_j^T)\|_F}{|\Omega|}$ Mean Frobenius error

$\mathrm{flag} = \left(\mathrm{MIE} \neq 0 \,\&\, \frac{|\mathrm{mFE}_{\mathrm{old}} - \mathrm{mFE}|}{|\mathrm{mFE}|} > \mathrm{tol}\right)$

endwhile

Output: R

The algorithm

$C_0 = \mathcal{P}_\Omega(A), \;\; W_0 = \text{rand}(m,k), H_0 = 0$

flag=TRUE, mFE $= 0$, $j = 0$

while($j < j_{\text{max}}$ & flag)

$j++;$

$\text{mFE}_{\text{old}} \leftarrow \text{mFE}$

$H_j = \text{argmin}_{X \geq 0} \|C_{j-1} - W_{j-1} X^T\|_F^2$

$W_j = \text{argmin}_{Y \geq 0} \|C_{j-1} - Y H_j^T\|_F^2$

$R \leftarrow \text{cut}_v([W_j H_j^T])$

$C_j \leftarrow \mathcal{P}_\Omega(A) + \mathcal{P}_{\bar{\Omega}}(R)$

$\text{MIE} = \max_\Omega |A - R|$ Maximum Integer error

$\text{mFE} = \frac{\|\mathcal{P}_\Omega(A - W_j H_j^T)\|_F}{|\Omega|}$ Mean Frobenius error

$\text{flag} = \left(\text{MIE} \neq 0 \,\&\, \frac{|\text{mFE}_{\text{old}} - \text{mFE}|}{|\text{mFE}|} > \text{tol}\right)$

endwhile

Output: R

The algorithm

$C_0 = \mathcal{P}_\Omega(A), \;\; W_0 = \text{rand}(m,k), H_0 = 0$
flag=TRUE, mFE $= 0, j = 0$
while$(j < j_{\text{max}} \,\&\, \text{flag})$
  $j++;$
  $\text{mFE}_{\text{old}} \leftarrow \text{mFE}$
  $H_j = \text{argmin}_{X \geq 0} \|C_{j-1} - W_{j-1} X^T\|_F^2$
  $W_j = \text{argmin}_{Y \geq 0} \|C_{j-1} - Y H_j^T\|_F^2$
  $R \leftarrow \text{cut}_v([W_j H_j^T])$
  $C_j \leftarrow \mathcal{P}_\Omega(A) + \mathcal{P}_{\bar{\Omega}}(R)$
  $\text{MIE} = \max_\Omega |A - R|$ Maximum Integer error
  $\text{mFE} = \frac{\|\mathcal{P}_\Omega(A - W_j H_j^T)\|_F}{|\Omega|}$ Mean Frobenius error
  $\text{flag} = \left(\text{MIE} \neq 0 \,\&\, \frac{|\text{mFE}_{\text{old}} - \text{mFE}|}{|\text{mFE}|} > \text{tol}\right)$
endwhile
Output: R

The algorithm

$C_0 = \mathcal{P}_\Omega(A), \;\; W_0 = \mathrm{rand}(m, k), H_0 = 0$
flag=TRUE, $\mathrm{mFE} = 0, j = 0$
while($j < j_{\max}$ & flag)
$j++;$
$\mathrm{mFE}_{\mathrm{old}} \leftarrow \mathrm{mFE}$
$H_j = \mathrm{argmin}_{X \geq 0} \|C_{j-1} - W_{j-1} X^T\|_F^2$
$W_j = \mathrm{argmin}_{Y \geq 0} \|C_{j-1} - Y H_j^T\|_F^2$
$R \leftarrow \mathrm{cut}_v([W_j H_j^T])$
$C_j \leftarrow \mathcal{P}_\Omega(A) + \mathcal{P}_{\bar{\Omega}}(R)$
$\mathrm{MIE} = \max_\Omega |A - R|$ Maximum Integer error
$\mathrm{mFE} = \frac{\|\mathcal{P}_\Omega(A - W_j H_j^T)\|_F}{|\Omega|}$ Mean Frobenius error
$\mathrm{flag} = \left(\mathrm{MIE} \neq 0 \; \& \; \dfrac{|\mathrm{mFE}_{\mathrm{old}} - \mathrm{mFE}|}{|\mathrm{mFE}|} > \mathrm{tol}\right)$
endwhile
Output: R

The algorithm

$C_0 = \mathcal{P}_\Omega(A), \;\; W_0 = \mathrm{rand}(m,k), H_0 = 0$
flag=TRUE, mFE $= 0, j = 0$
while($j < j_{\max}$ & flag)

$j++;$
$\mathrm{mFE}_{\mathrm{old}} \leftarrow \mathrm{mFE}$
$H_j = \mathrm{argmin}_{X \geq 0} \|C_{j-1} - W_{j-1}X^T\|_F^2$
$W_j = \mathrm{argmin}_{Y \geq 0} \|C_{j-1} - YH_j^T\|_F^2$
$R \leftarrow \mathrm{cut}_v([W_j H_j^T])$
$C_j \leftarrow \mathcal{P}_\Omega(A) + \mathcal{P}_{\bar{\Omega}}(R)$
$\mathrm{MIE} = \max_\Omega |A - R|$ Maximum Integer error
$\mathrm{mFE} = \frac{\|\mathcal{P}_\Omega(A - W_j H_j^T)\|_F}{|\Omega|}$ Mean Frobenius error
$\mathrm{flag} = (\mathrm{MIE} \neq 0 \,\&\, \dfrac{|\mathrm{mFE}_{\mathrm{old}} - \mathrm{mFE}|}{|\mathrm{mFE}|} > \mathrm{tol})$

endwhile
Output: R

Experimental results

Experiments to address

- Convergence of the iterative schema
- Accuracy of predictions

Data:

- MovieLens 100K, 1M, 10M ratings on a scale 1-5.
- Synthetic matrix generated as product of two random matrices of rank k.

Evaluation metrics

Let $\Theta \subseteq \Omega$, we define

- $\mathrm{MAE}_\Theta = \frac{\sum_{(u,i)\in\Theta} |a_{ui} - r_{ui}|}{|\Theta|}$ Mean Absolute Error

Evaluation metrics

- $\text{CMAE} = \text{MAE}_{\Delta} = \frac{\sum_{(u,i)\in\Delta} |a_{ui} - r_{ui}|}{|\Delta|}$, Constrained MAE
  $\Delta = \{(u,i) \in \Omega | \, r_{ui} \geq 4 \text{ or } a_{ui} \geq 4\}$

- $\text{0-1}_{\Theta} = \frac{\sum_{(u,i)\in\Theta} \text{mm}(a_{ui}, r_{ui})}{|\Theta|}$, 0-1 Loss

$$\text{mm}(a_{ui}, r_{ui}) = \begin{cases} 1 \text{ if } (a_{ui} \geq 4 \,\&\, r_{ui} < 4) | (a_{ui} < 4 \,\&\, r_{ui} \geq 4) \\ 0 \text{ otherwise.} \end{cases}$$

A low value of 0-1 Loss indicates that the algorithm returns almost always correct recommendations

Evaluation metrics

- $\text{CMAE} = \text{MAE}_\Delta = \frac{\sum_{(u,i)\in\Delta} |a_{ui} - r_{ui}|}{|\Delta|}$, Constrained MAE
  $\Delta = \{(u,i) \in \Omega |\, r_{ui} \geq 4 \text{ or } a_{ui} \geq 4\}$
- $\text{0-1}_\Theta = \frac{\sum_{(u,i)\in\Theta} \text{mm}(a_{ui}, r_{ui})}{|\Theta|}$, 0-1 Loss

$$\text{mm}(a_{ui}, r_{ui}) = \begin{cases} 1 \text{ if } (a_{ui} \geq 4 \,\&\, r_{ui} < 4) | (a_{ui} < 4 \,\&\, r_{ui} \geq 4) \\ 0 \text{ otherwise.} \end{cases}$$

A low value of 0-1 Loss indicates that the algorithm returns almost always correct recommendations

Evaluation metrics

$$\mathcal{S}_\Theta = \{(u,i) \in \Theta | a_{ui} \geq 4\}, \mathcal{R}_\Theta = \{(u,i) \in \Theta | r_{ui} \geq 4\}$$

- Precision: the fraction of items correctly recommended over the number of recommended items

$$\text{Precision}_\Theta = 100 \frac{|\mathcal{S}_\Theta| \cap |\mathcal{R}_\Theta|}{|\mathcal{R}_\Theta|},$$

- Recall: the fraction of items correctly recommended over the number of items that should be recommended

$$\text{Recall}_\Theta = 100 \frac{|\mathcal{S}_\Theta| \cap |\mathcal{R}_\Theta|}{|\mathcal{S}_\Theta|}.$$

Convergence

- Is our iterative scheme correct?
- Is the latent factor model adequate?

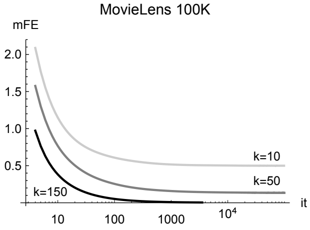
- For a sufficiently large value of k, mFE, MIE, MAE_Ω, $0\text{-}1_\Omega$ converge to zero?
- This corresponds to the correct reconstruction of all the observed ratings of the utility matrix.

Convergence

- Is our iterative scheme correct?
- Is the latent factor model adequate?

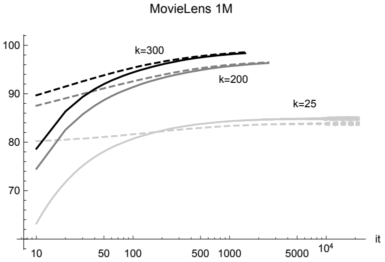
- For a sufficiently large value of k, mFE, MIE, MAE_Ω, 0-1_Ω converge to zero?
- This corresponds to the correct reconstruction of all the observed ratings of the utility matrix.

Convergence



For large values of k we have convergence to zero while for moderately small values of k the two metrics stagnate

Convergence



Trend of Precision (dashed lines) and Recall (solid) for different values of k. Since the cardinality of $\mathcal{S}_\Omega$ does not change during the iterations, the value of Precision increases in a more regular way. This is due to the increase of the set of recommendations $\mathcal{R}_\Omega$.

Convergence

Synthetic data: We construct a $1\,000 \times 5\,000$ matrix A_s with $\text{rank}(A_s) = 20$, and $|\Omega| = 500K$ (90% of the entries are ?).
Is our algorithm able to capture the structure of the rank-20 matrix?

Matrix	k	iter	MAE_Ω	0-1_Ω	Recall_Ω	Precision_Ω
	15	84,430	0.265	0.113	88.26	88.68
Synthetic	20	65,590	0.214	0.028	96.62	97.60
	25	51,480	0.199	0.022	97.41	98.09

The problem is underdetermined we expect that the larger the k the better the fit of the values in Ω. Even if A_s can be totally reconstructed using rank 20 matrix our algorithm misses to find the global minimum

Convergence

Synthetic data: We construct a $1\,000 \times 5\,000$ matrix A_s with $\text{rank}(A_s) = 20$, and $|\Omega| = 500K$ (90% of the entries are ?).
Is our algorithm able to capture the structure of the rank-20 matrix?

Matrix	k	iter	MAE_Ω	0-1_Ω	Recall_Ω	Precision_Ω
	15	84,430	0.265	0.113	88.26	88.68
Synthetic	20	65,590	0.214	0.028	96.62	97.60
	25	51,480	0.199	0.022	97.41	98.09

The problem is underdetermined we expect that the larger the k the better the fit of the values in Ω. Even if A_s can be totally reconstructed using rank 20 matrix our algorithm misses to find the global minimum

Convergence

Matrix	k	mFE	MAE_Ω	$0\text{-}1_\Omega$	$Recall_\Omega$	$Precision_\Omega$
MovieLens 100K	6	0.594	0.602	0.220	79.79	80.78
	10	0.500	0.551	0.1925	82.26	83.18
	50	0.133	0.269	0.055	94.93	95.10
	100	0.023	0.103	0.002	99.78	99.83
	150	0.004	0.040	0	100	100
MovieLens 1M	15	0.540	0.575	0.203	81.80	83.27
	25	0.470	0.536	0.181	83.81	84.86
	100	0.226	0.357	0.093	91.89	91.93
	200	0.109	0.234	0.041	96.46	96.33
	300	0.059	0.163	0.018	98.56	98.37

By increasing the value of k we get a better approximation and all the error measures decrease. The values 0 in 0-1 Loss and the 100% of Precision and Recall mean that we never have a mismatch in the reconstruction of positive ratings.

Accuracy of predictions

Split the data among two disjoint sets

- Ω_{80} Training Set: 80% of ratings uniformly sampled
- Θ_{20} Test Set: $\Theta_{20} = \Omega - \Omega_{80}$.
- The algorithm learns from Ω_{80} but the performance is evaluated on Θ_{20}.
- A large value of k is not a good option since the model tend to overfit the data
- The time is linear in k. Small k implies lower time

Accuracy of predictions

Split the data among two disjoint sets

- Ω_{80} Training Set: 80% of ratings uniformly sampled
- Θ_{20} Test Set: $\Theta_{20} = \Omega - \Omega_{80}$.
- The algorithm learns from Ω_{80} but the performance is evaluated on Θ_{20}.
- A large value of k is not a good option since the model tend to overfit the data
- The time is linear in k. Small k implies lower time

Accuracy of predictions

Split the data among two disjoint sets

- Ω_{80} Training Set: 80% of ratings uniformly sampled
- Θ_{20} Test Set: $\Theta_{20} = \Omega - \Omega_{80}$.

- The algorithm learns from Ω_{80} but the performance is evaluated on Θ_{20}.
- A large value of k is not a good option since the model tend to overfit the data
- The time is linear in k. Small k implies lower time

Accuracy of predictions

Split the data among two disjoint sets

- Ω_{80} Training Set: 80% of ratings uniformly sampled
- Θ_{20} Test Set: $\Theta_{20} = \Omega - \Omega_{80}$.

- The algorithm learns from Ω_{80} but the performance is evaluated on Θ_{20}.
- A large value of k is not a good option since the model tend to overfit the data
- The time is linear in k. Small k implies lower time

Accuracy of predictions

Matrix	k	$MAE_{\Omega_{80}}$	$0\text{-}1_{\Theta_{20}}$	$Recall_{\Theta_{20}}$	$Precision_{\Theta_{20}}$
MovieLens 1M	6	0.6253	0.2550	77.04	79.27
	10	0.5940	0.2555	77.21	78.82
	15	0.5642	0.2605	77.05	77.89
MovieLens 10M	6	0.6093	0.2630	74.56	72.69
	10	0.5937	0.2612	74.95	72.48
	15	0.5864	0.2633	75.20	71.70

Predictions of recommendations:

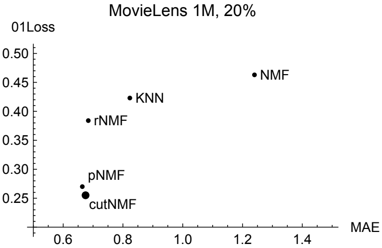
- Low values of 0-1 Loss
- High Precision and Recall

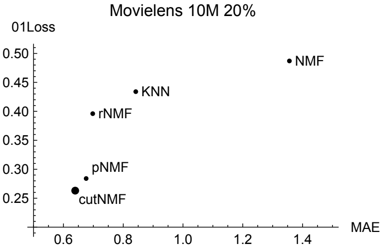
Comparison with other NMF methods

	MovieLens 1M			MovieLens 10M		
Method	$MAE_{\Theta_{20}}$	$CMAE_{\Theta_{20}}$	$0\text{-}1_{\Theta_{20}}$	$MAE_{\Theta_{20}}$	CMAE	$0\text{-}1_{\Theta_{20}}$
KNN Pearson	0.823	0.721	0.423	0.842	0.743	0.434
NMF	1.243	1.106	0.463	1.356	1.234	0.487
rNMF	0.684	0.574	0.384	0.698	0.586	0.396
pNMF	**0.664**	**0.526**	0.270	0.676	**0.542**	0.284
cutNMF	0.675	0.659	**0.255**	**0.639**	0.654	**0.263**

We see that our method performs well for the 0-1 Loss and MAE measures. In this table are reported the values obtained with $k = 6$ (the best value for pNMF).

Comparison with other methods





Conclusion and further work

- We proposed an Adaptive approach for Personalized recommendations
- Assumes the latent factor model
- Dataset-dependent
- Shown convergence and effectiveness in recommending items

What's next

- Since the goal is to recommend only a few top items to a user, use other error measure such as NDCG, AUR, etc.
- We are interested in recommending only a few items to each user, it is not really important to retrieve all the missing ratings.
- Understand better the role of k.