

Lecture 8 - Graphs and Machine Learning II

F. d'Alché-Buc, E. Le Pennec

Fall 2016

Introduction to recommendation systems

Outline

1 Introduction to recommendation systems

2 Measures of success, preprocessing

3 K-NN approach to CF

4 Matrix Factorization

- Intro
- Link with PCA
- A convex formulation
- Proximal gradient descent

5 Illustrations

- A groundbreaking theory

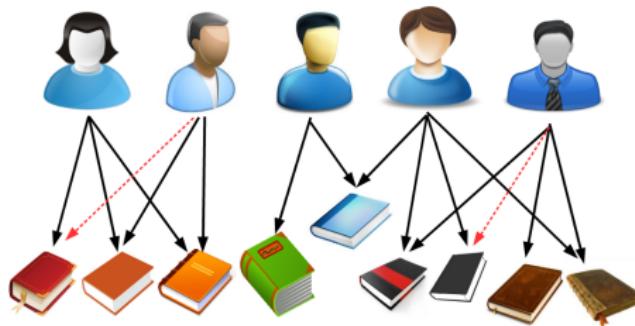
6 Conclusion and perspectives

7 References

Introduction to recommendation systems

Recommendation

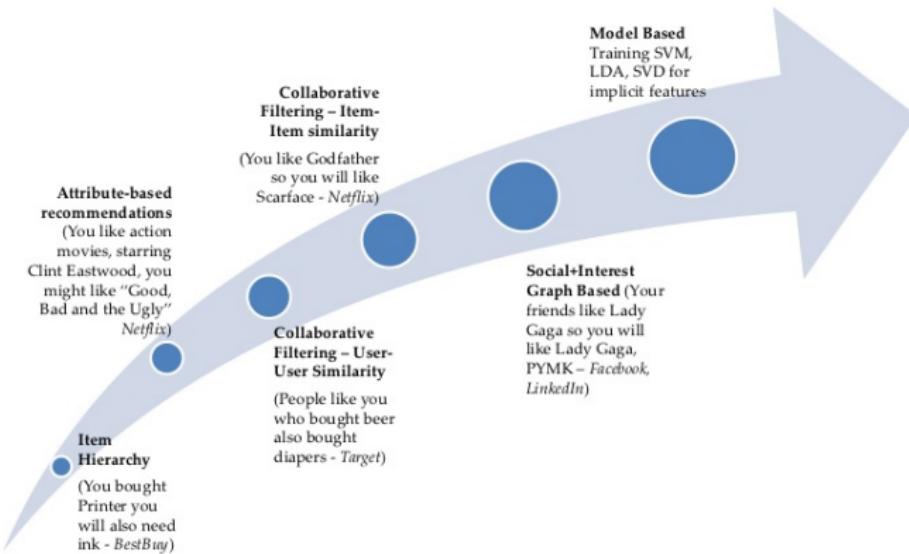
A set of users, a set of items, some users rate some items. Given a user, predict a score attached to an item.



Introduction to recommendation systems

Approaches to recommendation

Recommender Approaches



Introduction to recommendation systems

Using item content and user profile

If items could be described by feature vectors x or pairs of some metric based on content, and idem for users profile y . Then, build a function $f : (x, y) \rightarrow r$ from a training set

$$\{((x_i, y_i), r_i), i = 1, \dots, n\}$$

Any supervised regression method able to cope with an ordered pair is suitable.

Example: Movies: genre, actors, ... People: age, gender, job, ...

- Appropriate metric or

kernel: $k((x, y), (x', y')) = k_i(x, x')k_u(y, y')$: kernel approaches
OR k-nearest neighbors

- Tree-based methods on input $z^T = [x^T \ y^T]$

N.B.: it is closed to a link prediction problem in a bipartite graph

Introduction to recommendation systems

Using item content and user profile

- Pros

- You learn a function so you can predict for a new pair of item and user
- Benefit from existing regression methods

- Cons

- Requires features, maybe easy in the case of items but difficult in the case of users
- Data are no more iid

Introduction to recommendation systems

No features on items neither on users

What is collaborative filtering? a transductive problem.

Many **users**



Many **items**



- Based on many observed user-item interactions (rating, purchase, click)
- Predict new interactions

Introduction to recommendation systems

Collaborative Filtering

Amazon.com recommends products based on your purchases, browsing history

- but based on the purchases and history of other users too

Ces recommandations sont basées sur les [articles que vous possédez](#) et plus encore.

afficher: [Tous](#) | [Nouveautés](#) | [Bientôt](#)

1. [**Hypothermie : Une enquête du commissaire Erlendur Sveinsson**](#)
de Arnaldur Indriðason (19 mai 2011)
Moyenne des commentaires client :  (60)
En stock

Prix conseillé : EUR 7,90
Prix : EUR 6,94
87 neufs et d'occasion à partir de EUR 0,92

 [Ajouter au panier](#) [Ajouter à votre liste d'envies](#)

Vous l'avez déjà Vous n'êtes pas intéressé  Évaluez cet article

Recommandé parce que vous avez acheté **L'homme du lac** et plus ([Modifier](#))

Introduction to recommendation systems

Collaborative Filtering

Google News recommends news based on your browsing activity

- but on the browsing activity of other users too

À la une

Quand Patrick Buisson enregistrait Nicolas Sarkozy

Les Echos - Il y a 3 minutes

Dans son édition à paraître mercredi, « Le canard enchaîné » publie le verbatim de l'enregistrement d'une réunion à l'Elysée, réalisé en 2011 par Patrick Buisson, alors conseiller de Nicolas Sarkozy, au moyen d'un dictaphone. Patrick Buisson était le ...

Un enregistrement de Buisson à l'Elysée retranscrit dans le Canard ... TF1

Un après-midi à l'Elysée enregistré par Patrick Buisson Europe1

Citée à de nombreuses reprises : Sarkoleaks : Le Canard Enchaîné publie le script des ... Le Lab Europe 1

Autres
Patrick Buisson »
Nicolas Sarkozy »

Voir l'actu en direct

Ukraine : John Kerry à Kiev, Poutine sort de son silence

Le Figaro - Il y a 25 minutes

Le président russe Vladimir Poutine exclut pour le moment une intervention armée. A Kiev, le secrétaire d'Etat américain John Kerry dénonce un "acte d'agression". A VENIR : Début du direct : le 12/03/2012 à 10h55. EN COURS : Mis à jour il y a quelques ...

Paris : un cyber-jihadiste condamné à un an de prison ferme

Le Parisien - Il y a 41 minutes

Il traduisait les revues de propagande d'Al Qaida et se faisait appeler Abou Siyad Al-Normandy sur le site jihadiste qu'il animait. Romain Letellier, un musulman converti de 27 ans, a été condamné mardi à un an de prison ferme. Le tribunal correctionnel de ...

Le sondage non publié sur DSK n'excite pas les strauss-kahniens

L'Express - Il y a 1 heure

Le Parisien Magazine n'a pas publié les réactions des proches de DSK à un sondage BVA que l'hebdomadaire a choisi de ne pas divulguer. L'Express les a contactés. Imprimer. Zoom moins. Zoom plus. 15. Voter (1). Le sondage non publié sur DSK n'excite ...

#GrâciasPuyol : la vidéo poignante du Barça en hommage à son ...

Eurosport.com FR - Il y a 20 minutes

À l'occasion de l'annonce de son départ en fin de saison, le FC Barcelone a rendu hommage à Carles Puyol à travers une émouvante vidéo. Sur un fond sonore poignant, on y voit notamment les premiers pas du joueur avec le maillot blaugrana.

Pacte de responsabilité: le projet du patronat rejeté par les syndicats

Le Nouvel Observateur - Il y a 3 minutes

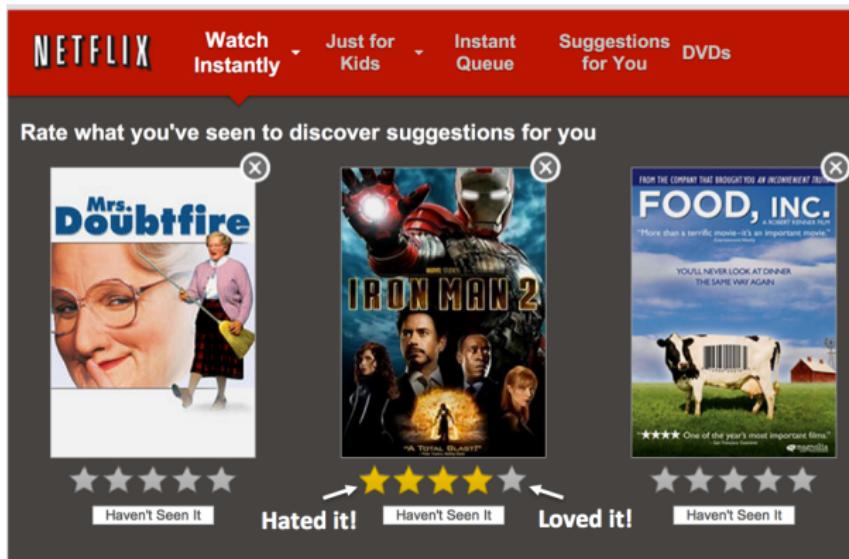
Paris (AFP) - Le patronat a présenté mardi un projet d'accord sur les contreparties du "pacte de responsabilité", essuyant un tir groupé des syndicats qui l'ont jugé totalement insuffisant, à la veille d'une deuxième séance de discussion entre les partenaires ...

Introduction to recommendation systems

Collaborative Filtering

Netflix predicts the rating you'd give to a movie

- using all the ratings given by all users



- 60% of Netflix's DVD rentals due to recommendations

Introduction to recommendation systems

Collaborative Filtering

Netflix predicts the rating you'd give to a movie

- using all the ratings given by all users

Sci-Fi & Fantasy



Inception

Because

Memen

The

Batma

Add

Not Interested

5 stars

Not Interested

Inception

2010 PG-13 148 minutes

Dom Cobb earns a tidy sum infiltrating the dreams of corporate titans to steal their most closely held secrets.

Starring: Leonardo DiCaprio, Joseph Gordon-Levitt

Director: Christopher Nolan

Genre: Sci-Fi & Fantasy

Availability: DVD and Blu-ray

★★★★★ 4.4 Our best guess for LESTER

Heart icon Recommended based on your interest in: *Batman Begins, The Matrix and Memento*

- 60% of Netflix's DVD rentals due to recommendations

Introduction to recommendation systems

Collaborative Filtering: Netflix Prize



Introduction to recommendation systems

Collaborative Filtering: Netflix Prize



Introduction to recommendation systems

Collaborative Filtering: Netflix Prize

Netflix Prize

- October 2, 2006
- Dataset: 100 million ratings , $n_U = 480K$ users, $n_I = 18K$ movies
- **only 1.1%** of the matrix is filled!
- **Goal:** create a computer code that predicts ratings
- \$1m grand prize for anyone beating **Cinematch** accuracy by 10%
- 5000 teams over 150 countries participated

Introduction to recommendation systems

Collaborative Filtering, Matrix Completion

Collaborative Filtering, Matrix completion: **fill unobserved entries** of a matrix

$$\begin{bmatrix} \times & ? & ? & ? & \times & ? \\ ? & ? & \times & \times & ? & ? \\ \times & ? & ? & \times & ? & ? \\ ? & ? & \times & ? & ? & \times \\ \times & ? & ? & ? & ? & ? \\ ? & ? & \times & \times & ? & ? \end{bmatrix}$$

- Unknown matrix M has size $n_U \times n_I$
- Observe $m \ll n_U n_I$ entries ($100m \ll 8.64M$ for Netflix)
- Seems impossible!

Introduction to recommendation systems

Collaborative Filtering

There is hope:

- Personal preferences are correlated: if Alice likes A and B and Bob likes A, B and C , then Alice is more likely to like C
- There are latent factors that describe the data in a much lower dimensional space. Groups of users, groups of movies, factors that explain the taste of users. High-dimensionality but hidden low-dimensional structure

Collaborative Filtering task:

- discover patterns (low-dimensional hidden structure, latent factors)
- use these patterns for prediction of new user-item interactions
- Do not necessarily use item or user attributes (demographic information, author, genre, cast, plot, etc.)

Introduction to recommendation systems

Collaborative Filtering

Given:

- Users $u \in \{1, \dots, n_U\}$
- Items $i \in \{1, \dots, n_I\}$
- When u has an interaction with item i , (watches the movie, clicks on a banner, buys a product), he enters a scalar rating $r_{u,i}$ (number of clicks, rating of a movie, etc.)
- Set E of pairs (u, i) of observed ratings $r_{u,i}$

Matrix completion problem

$$M = \begin{bmatrix} ? & ? & 2 & \cdots & 5 \\ 2 & ? & 1 & \cdots & ? \\ ? & 2 & ? & \cdots & 4 \end{bmatrix}$$

Measures of success, preprocessing

Outline

1 Introduction to recommendation systems

2 Measures of success, preprocessing

3 K-NN approach to CF

4 Matrix Factorization

- Intro
- Link with PCA
- A convex formulation
- Proximal gradient descent

5 Illustrations

- A groundbreaking theory

6 Conclusion and perspectives

7 References

Measures of success, preprocessing

Collaborative Filtering

Measures of success. Decompose $E = E_{\text{train}} \cup E_{\text{test}}$ into training and testing respectively. $r_{u,i}$ = ground truth and $\hat{r}_{u,i}$ = estimated rating

- Root Mean Square Error

$$\text{RMSE} = \sqrt{\frac{1}{|E_{\text{test}}|} \sum_{(u,i) \in E_{\text{test}}} (r_{u,i} - \hat{r}_{u,i})^2}$$

- Mean Absolute Error

$$\text{MAE} = \frac{1}{|E_{\text{test}}|} \sum_{(u,i) \in E_{\text{test}}} |r_{u,i} - \hat{r}_{u,i}|$$

- Ranking error: fraction of true top-5 preferences are in my predicted top 5?

Measures of success, preprocessing

Biases

- Remove biases from the ratings
- Some users give systematically higher ratings
- Some items get systematically better rates (old movies...)
- Don't forget that PCA needs centering

Remove bias

$$\tilde{r}_{u,i} = r_{u,i} - b_{u,i}$$

Let's denote

- $U(i)$ the set of users who rated item i
- $I(u)$ the set of items rated by user u

Measures of success, preprocessing

Biases

Let's compute means! We can choose $b_{u,i}$ as one of the following:

- Global mean

$$b = \frac{1}{|E|} \sum_{(u,i) \in E} r_{u,i}$$

- Item's mean rating

$$b_i = \frac{1}{|U(i)|} \sum_{u \in U(i)} r_{u,i}$$

- User's mean rating

$$b_u = \frac{1}{|I(u)|} \sum_{i \in I(u)} r_{u,i}$$

- Item's mean rating + user's mean deviation from item mean

$$b_{u,i} = b_i + \frac{1}{|I(u)|} \sum_{i' \in I(u)} (r_{u,i'} - b_{i'})$$

Measures of success, preprocessing

Interpolating means

- Some users have much more ratings than other ($1000 \times$ more!)
- Users with a small numbers of ratings are not as reliable as ones: more noisy. It's hard to trust a mean with only one rating!
- Interpolate between a global estimate and an estimate from user's data

Interpolate to have a better bias estimation:

$$\tilde{b}_u = \frac{\alpha}{\alpha + |I(u)|} b + \frac{|I(u)|}{\alpha + |I(u)|} b_u$$



where we recall that b is the global mean

$$b = \frac{1}{|E|} \sum_{(u,i) \in E} r_{u,i}$$

and b_u is the user's mean

$$b_u = \frac{1}{|I(u)|} \sum_{i \in I(u)} r_{u,i}$$

K-NN approach to CF

Outline

- 1 Introduction to recommendation systems
- 2 Measures of success, preprocessing
- 3 K-NN approach to CF
- 4 Matrix Factorization
 - Intro
 - Link with PCA
 - A convex formulation
 - Proximal gradient descent
- 5 Illustrations
 - A groundbreaking theory
- 6 Conclusion and perspectives
- 7 References

K-NN approach to CF

K-NN

K-Nearest Neighbor Method (K-NN) most widely used family of methods

- item-based or user-based
- for product recommendation: **item-based**
- represent each item as a vector of user's ratings with many missing values $r_{\bullet,i} = [2, ?, 1, ?, ?, ?, 2, 1, ?, ?, ?, 5]$

Idea: users rate similar items similarly.

In order to predict a rating $r_{u,i}$ for user u and item i :

- Compute similarity between i and every other items
- Find the K items rated by u most similar to i
- Compute weighted average of these ratings

K-NN approach to CF

K-NN: Similarity measures

How to measure similarity between items?

- Cosine similarity

$$d(r_{\bullet,i}, r_{\bullet,i'}) = \frac{\langle r_{\bullet,i}, r_{\bullet,i'} \rangle}{\|r_{\bullet,i}\| \|r_{\bullet,i'}\|}$$

- Pearson correlation coefficient

$$d(r_{\bullet,i}, r_{\bullet,i'}) = \frac{\langle r_{\bullet,i} - \bar{r}_{\bullet,i}, r_{\bullet,i'} - \bar{r}_{\bullet,i'} \rangle}{\|r_{\bullet,i} - \bar{r}_{\bullet,i}\| \|r_{\bullet,i'} - \bar{r}_{\bullet,i'}\|}$$

- Inverse Euclidean distance

$$d(r_{\bullet,i}, r_{\bullet,i'}) = \frac{1}{\|r_{\bullet,i} - r_{\bullet,i'}\|}$$

Vectors $r_{\bullet,i}$ contains many missing values: compute these similarities over subsets of users that rated both items i and i'

K-NN approach to CF

K-NN

How to choose the K -nearest neighbors?

- Select the K items with largest similarity score to item i , among the items rated by u , denoted $N(i, u)$
- Prediction given by

$$\hat{r}_{u,i} = b_{u,i} + \sum_{i' \in N(i,u)} w_{i,i'} (r_{u,i'} - b_{u,i'})$$

where $w_{i,i'}$ weights

Example of weights:

- Equal weights

$$w_{i,i'} = \frac{1}{N(u,i)}$$

- Similarity weights

$$w_{i,i'} = \frac{d(r_{\bullet,i}, r_{\bullet,i'})}{\sum_{i'' \in N(i,u)} d(r_{\bullet,i}, r_{\bullet,i''})}$$

K-NN approach to CF

K-NN

Even better: user optimized weights.

- Choose weights that best predict other known ratings of i among all users that rated i
- Corresponds to many small linear regression problems
- Needs to store **many** weights $O(n_I^2)$

K-NN approach to CF

Conclusion for K-NN methods

- Easy to implement
- No training time
- Flexible
- But need to store many parameters (all item, vectors, weights in memory)
- Don't exploit hidden low-dimensional structure

Matrix Factorization

Outline

- 1 Introduction to recommendation systems
- 2 Measures of success, preprocessing
- 3 K-NN approach to CF
- 4 Matrix Factorization
 - Intro
 - Link with PCA
 - A convex formulation
 - Proximal gradient descent
- 5 Illustrations
 - A groundbreaking theory
- 6 Conclusion and perspectives
- 7 References

- 1 Introduction to recommendation systems
- 2 Measures of success, preprocessing
- 3 K-NN approach to CF
- 4 Matrix Factorization
 - Intro
 - Link with PCA
 - A convex formulation
 - Proximal gradient descent
- 5 Illustrations
 - A groundbreaking theory
- 6 Conclusion and perspectives
- 7 References

Matrix Factorization

MF: if I know the item's features

Matrix Factorization (= MF)

- Assume that we know features about the items

$$y_i = [\text{cast}, \text{year}, \text{genre}, \dots] \in \mathbb{R}^r$$

for all $i = 1, \dots, n_I$.

- r features for each item
- We want the users parameters x_u for $u = 1, \dots, n_U$

Linear regression

$$\hat{x}_u \in \operatorname{argmin}_{x_u} \sum_{i \in I(u)} (r_{u,i} - \langle x_u, y_i \rangle)^2.$$

Even better: **ridge regression**

$$\hat{x}_u = \operatorname{argmin}_{x_u} \sum_{i \in I(u)} (r_{u,i} - \langle x_u, y_i \rangle)^2 + \lambda \|x_u\|_2^2$$

- But we don't want to construct ad-hoc features y_i for items
- Not a good idea for building recommendations

Matrix Factorization

MF: if I know the user's features

- Assume that we know user's features x_u for all $u = 1, \dots, n_U$.
- r features for each user
- We want the items features y_i for $i = 1, \dots, n_I$

Once again: **linear regression**

$$\hat{y}_i \in \operatorname{argmin}_{y_i} \sum_{u \in U(i)} (r_{u,i} - \langle x_u, y_i \rangle)^2.$$

Even better: **ridge regression**

$$\hat{y}_i \in \operatorname{argmin}_{y_i} \sum_{u \in U(i)} (r_{u,i} - \langle x_u, y_i \rangle)^2 + \lambda \|y_i\|_2^2$$

- But we can't construct ad-hoc features x_u for users
- Still not a good idea for building recommendations

Matrix Factorization

MF: put everything together

- We don't want to construct ad-hoc features y_i for items
- We can't construct ad-hoc features x_u for users

So let's learn items and users features **at the same time!**

Putting altogether

$$\hat{x}_u = \operatorname{argmin}_{x_u} \sum_{i \in I(u)} (r_{u,i} - \langle x_u, \hat{y}_i \rangle)^2 + \lambda \|x_u\|_2^2$$

$$\hat{y}_i = \operatorname{argmin}_{y_i} \sum_{u \in U(i)} (r_{u,i} - \langle \hat{x}_u, y_i \rangle)^2 + \lambda \|y_i\|_2^2$$

for all $u = 1, \dots, n_U$ and $i = 1, \dots, n_I$

Matrix Factorization

MF: put everything together

$$\hat{x}_u = \operatorname{argmin}_{x_u} \sum_{i \in I(u)} (r_{u,i} - \langle x_u, \hat{y}_i \rangle)^2 + \lambda \|x_u\|_2^2$$

$$\hat{y}_i = \operatorname{argmin}_{y_i} \sum_{u \in U(i)} (r_{u,i} - \langle \hat{x}_u, y_i \rangle)^2 + \lambda \|y_i\|_2^2$$

for all $u = 1, \dots, n_U$ and $i = 1, \dots, n_I$

- **Issue:** the \hat{x}_u 's depends on the \hat{y}_i 's that depend on the \hat{x}_u 's that depends on the \hat{y}_i that... !

Matrix Factorization

MF: put everything together

Let's rewrite this. Put

$$X^\top = \begin{bmatrix} \vdots & \vdots & \vdots \\ x_1 & \cdots & x_{n_U} \\ \vdots & \vdots & \vdots \end{bmatrix} \quad \text{and} \quad Y^\top = \begin{bmatrix} \vdots & \vdots & \vdots \\ y_1 & \cdots & y_{n_I} \\ \vdots & \vdots & \vdots \end{bmatrix}$$

Then we consider the minimization of

$$F(X, Y) = \sum_{(u,i) \in E} (r_{u,i} - \langle x_u, y_i \rangle)^2 + \lambda \sum_{u=1}^{n_U} \|x_u\|_2^2 + \lambda \sum_{i=1}^{n_I} \|y_i\|_2^2$$

over $X \in \mathbb{R}^{n_U \times r}$ and $Y \in \mathbb{R}^{n_I \times r}$ jointly.

- The penalization terms $\lambda \sum_{u=1}^{n_U} \|x_u\|_2^2 + \lambda \sum_{i=1}^{n_I} \|y_i\|_2^2$ prevent from overfitting

Matrix Factorization

MF

$$F(X, Y) = \sum_{(u,i) \in E} (r_{u,i} - \langle x_u, y_i \rangle)^2 + \lambda \sum_{u=1}^{n_U} \|x_u\|_2^2 + \lambda \sum_{i=1}^{n_I} \|y_i\|_2^2$$

over $X \in \mathbb{R}^{n_U \times r}$ and $Y \in \mathbb{R}^{n_I \times r}$ jointly.

Let's write this with matrices

$$F(X, Y) = \|\mathcal{P}_E(R - XY^\top)\|_F^2 + \lambda\|X\|_F^2 + \lambda\|Y\|_F^2$$

where

$$\|A\|_F = \text{Frobenius norm of } A = \sqrt{\sum_{j,k} A_{j,k}^2}$$

and

$$\mathcal{P}_E(A)_{u,i} = \begin{cases} A_{u,i} & \text{if } (u, i) \in E \\ 0 & \text{otherwise} \end{cases}$$

Matrix Factorization

Matrix Factorization

Put $\lambda = 0$ and $E = \{1, \dots, n_U\} \times \{1, \dots, n_I\}$

$$F(X, Y) = \|\mathcal{P}_E(R - XY^\top)\|_F^2 = \|R - XY^\top\|_F^2$$

Solution is given by the SVD (Singular Value Decomposition)

Recall that

- X of size $n_U \times r$
- Y of size $n_I \times r$

Then

$$\operatorname{argmin}_{X, Y} \|R - XY^\top\|_F^2$$

is given by thresholded SVD of R

Matrix Factorization

Matrix Factorization: SVD

SVD (Singular Value Decomposition)

Any matrix $R \in \mathbb{R}^{n_U \times n_I}$ writes

$$R = U\Sigma V^\top$$

where

- U is the matrix of **left singular vectors** (columns of U are eigenvectors of RR^\top , it satisfies $U^\top U = I$)
- V is the matrix of **right singular vectors** (eigenvectors of $R^\top R$, it satisfies $V^\top V = I$)
- $\Sigma = \text{diag}[\sigma_1, \dots, \sigma_{n_U \wedge n_I}]$ is the diagonal matrix containing the **singular values**

$$\sigma_1 \geq \dots \geq \sigma_{n_U \wedge n_I}$$

where

$$\sigma_j = \sqrt{\lambda_j(R^\top R)} = j\text{th eigenvalue of } R^\top R$$

Matrix Factorization

Matrix Factorization: SVD

$$M = U \Sigma V^*$$

Fundamental result:

$$\underset{M \in \mathbb{R}^{n_u \times n_l : \text{rank}(X)=r}}{\operatorname{argmin}} \|R - M\|_2^2 = U_r \Sigma_r V_r^\top$$

where $R = U \Sigma V^\top$ is the SVD of R and

- $\Sigma_r = \operatorname{diag}[\sigma_1, \dots, \sigma_r]$
- U_r contains the first r columns of U
- V_r contains the first r columns of V
- Don't forget that PCA = SVD of the covariance matrix

Matrix Factorization

Matrix Factorization: SVD

Hence a solution of

$$(\hat{X}, \hat{Y}) \in \operatorname{argmin}_{X, Y} \|R - XY^\top\|_F^2$$

is given by

$$\hat{X} = U_r \Sigma_r \quad \text{and} \quad \hat{Y} = V_r^\top$$

- Matrix completion can be understood as an SVD with missing entries
- With extra regularization to avoid overfitting using ridge penalization

Matrix Factorization

Matrix Factorization: algorithms

Ok. So how do I solve

$$F(X, Y) = \sum_{(u,i) \in E} (r_{u,i} - \langle x_u, y_i \rangle)^2 + \lambda \sum_{u=1}^{n_U} \|x_u\|_2^2 + \lambda \sum_{i=1}^{n_I} \|y_i\|_2^2$$

or equivalently

$$F(X, Y) = \|\mathcal{P}_E(R - XY^\top)\|_F^2 + \lambda \|X\|_F^2 + \lambda \|Y\|_F^2$$

Matrix Factorization

Algorithm 1. Alternating Least-Squares (ALS)

Idea: if we knew Y we could solve X using ridge regression and vice-versa: alternate between optimizing on X and Y with the other matrix fixed

Alternating least-squares (ALS) algorithm

Repeat until convergence:

- For each u solve the linear system:

$$x_u^{\text{new}} \leftarrow \text{solution of } \sum_{i \in I(u)} (y_i y_i^\top + \lambda I) x_u = \sum_{i \in I(u)} r_{u,i} y_i$$

- For each item i solve

$$y_i^{\text{new}} \leftarrow \text{solution of } \sum_{u \in U(i)} (x_u x_u^\top + \lambda I) y_i = \sum_{u \in U(i)} r_{u,i} x_u$$

- $x_u \leftarrow x_u^{\text{new}}$, $y_i \leftarrow y_i^{\text{new}}$

- Updates for x_u and y_i can be done in parallel
- Complexity. Space: $O(n_u r + n_I r)$ and time: $O(n_u r^3 + n_I r^3)$ per iteration. $O(r^3)$ for solving the linear systems
- No need to store the complete ratings matrix

Matrix Factorization

Algorithm 2. Gradient Descent (GD)

Idea: use standard gradient descent

- $\nabla_{x_u} F(X, Y) = \lambda x_u + \sum_{i \in I(u)} (\langle x_u, y_i \rangle - r_{u,i}) y_i$
- $\nabla_{y_i} F(X, Y) = \lambda y_i + \sum_{u \in U(i)} (\langle x_u, y_i \rangle - r_{u,i}) x_u$

Gradient Descent algorithm

Repeat until *convergence*:

- For each u update

$$x_u^{\text{new}} \leftarrow x_u - \eta \left(\lambda x_u + \sum_{i \in I(u)} (\langle x_u, y_i \rangle - r_{u,i}) y_i \right)$$

- For each i update

$$y_i^{\text{new}} \leftarrow y_i - \eta \left(\lambda y_i + \sum_{u \in U(i)} (\langle x_u, y_i \rangle - r_{u,i}) x_u \right)$$

- $x_u \leftarrow x_u^{\text{new}}$, $y_i \leftarrow y_i^{\text{new}}$

- Updates for x_u and y_i can be done in parallel
- Complexity: $O(n_u r + n_I r)$ no $O(r^3)$ overhead iteration
- No need to store the complete ratings matrix

Matrix Factorization

Algorithm 3. Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent algorithm

Repeat until *convergence*:

- Choose $(u, i) \in E$ at random

- Update x_u

$$x_u^{\text{new}} \leftarrow x_u - \eta(\lambda x_u + (\langle x_u, y_i \rangle - r_{u,i})y_i)$$

- Update y_i

$$y_i^{\text{new}} \leftarrow y_i - \eta(\lambda y_i + (\langle x_u, y_i \rangle - r_{u,i})x_u)$$

- $x_u \leftarrow x_u^{\text{new}}, y_i \leftarrow y_i^{\text{new}}$

- Complexity: $O(n_u r + n_I r)$ no $O(r^3)$ overhead iteration
- No need to store the complete ratings matrix

Matrix Factorization

Conclusion for CF using Matrix Factorization

Parameters to tune:

- step-size, or learning rate η . Must be decreasing for SGD
- Regularization parameter $\lambda > 0$ and number of latent factors r . Tuned using cross-validation

There is a big problem:

$$F(X, Y) = \sum_{(u,i) \in E} (r_{u,i} - \langle x_u, y_i \rangle)^2 + \lambda \sum_{u=1}^{n_U} \|x_u\|_2^2 + \lambda \sum_{i=1}^{n_I} \|y_i\|_2^2$$

is **not** a convex problem

- Local minimum, initialization is important
- No guarantees towards a good minimum
- Mostly heuristics

- 1 Introduction to recommendation systems
- 2 Measures of success, preprocessing
- 3 K-NN approach to CF
- 4 Matrix Factorization
 - Intro
 - Link with PCA
 - A convex formulation
 - Proximal gradient descent
- 5 Illustrations
 - A groundbreaking theory
- 6 Conclusion and perspectives
- 7 References

Matrix Factorization

A convex formulation of the matrix completion problem

- Unknown matrix R of size $n_U \times n_I$
- If R has rank r , its degrees of freedom are $r(n_U + n_I - r)$
- $E \subset \{1, \dots, n_U\} \times \{1, \dots, n_I\}$ of observed entries of R
- We need $|E| \geq r(n_U + n_I - r)$ (otherwise no hope to recover R)
- We observe only $\mathcal{P}_E(R)$

We assume that the rank of R is small. So let's penalize the rank

- Tempting to consider

$$\hat{R} \in \operatorname*{argmin}_{M \in \mathbb{R}^{n_U \times n_I}} \left\{ \frac{1}{2} \|\mathcal{P}_E(M - R)\|_F^2 + \lambda \operatorname{rank}(M) \right\}$$

- Too hard
- For Lasso we've found that a convex relaxation of ℓ_0 is ℓ_1
- Can't we do the same for the rank?

Yes!

Matrix Factorization

A convex formulation of the matrix completion problem

Convex relaxation for the rank

$$\text{rank}(M) = \sum_{k=1}^{n_U \wedge n_I} \mathbf{1}_{\sigma_j(M) > 0} = \|\sigma(M)\|_0$$

Use nuclear norm:

$$\|M\|_* = \sum_{j=1}^{n_I \wedge n_U} \sigma_j(M)$$

Hence tempting to consider

$$\hat{R} \in \operatorname{argmin}_{M \in \mathbb{R}^{n_I \times n_U}} \left\{ \frac{1}{2} \|\mathcal{P}_E(M - R)\|_2^2 + \lambda \|M\|_* \right\}$$

for a regularization parameter $\lambda > 0$. This is a **convex** problem!

Proximal gradient descent for the CF problem

Repeat until *convergence*:

- $M \leftarrow S_{\lambda\eta_k}(M - \eta_k(\mathcal{P}_E(M - R)))$

where S_λ is the spectral soft-thresholding operator: if $M = U\Sigma V^\top$ SVD of M , then

$$S_\lambda(M) = U \text{diag}[(\sigma_1 - \lambda)_+, \dots, (\sigma_{n_1 \wedge n_2} - \lambda)_+] V^\top$$

Thresholding of the singular value: the solution will be of low rank.
Many other algorithms, more memory efficient and faster

Matrix Factorization

Proximal gradient descent

- Convex problem: convex optimization and convergence guarantees to a minimum
- Bottleneck: a SVD is required at **each iteration!** Complexity of a SVD $O((n_U \vee n_I)(n_U \wedge n_I)^2)$
- Can be reduced using partial SVD (compute only k top singular values and vectors). Complexity is (best case) $O(n_1 n_2 k)$
[keyword: **Lanczos** algorithms]
- Compute an approximate solution, given some tolerance
- Remedy for large SVD is the **divide and conquer** principle

Illustrations

Outline

1 Introduction to recommendation systems

2 Measures of success, preprocessing

3 K-NN approach to CF

4 Matrix Factorization

- Intro
- Link with PCA
- A convex formulation
- Proximal gradient descent

5 Illustrations

- A groundbreaking theory

6 Conclusion and perspectives

7 References

Illustrations

Collaborative Filtering, Matrix Completion

Sketch of application: image inpainting

THE SALINAS VALLEY is in Northern California. It is a long narrow swale between two ranges of mountains, and the Salinas River winds and twists up the center until it falls at last into Monterey Bay.

I remember my childhood names for grasses and secret flowers. I remember where a toad may live and what time the birds awaken in the summer—and what trees and seasons smelled like—how people looked and walked and smelled even. The memory of odors is very rich.

I remember that the Gabilan Mountains to the east of the valley were light gay mountains full of sun and loveliness and a kind of invitation, so that you wanted to climb into their warm foothills almost as you want to climb into the lap of a beloved mother. They were beckoning mountains with a brown grass love. The Santa Lucias stood up against the sky to the west and kept the valley from the open sea, and they were dark and brooding-unfriendly and dangerous. I always found in myself a dread of west and a love of east. Where I ever got such an idea I cannot say, unless it could be that the morning came over the peaks of the Gabilans and the night drifted back from the ridges of the Santa Lucias. It may be that the birth and death of the day had some part in my feeling about the two ranges of mountains.

From both sides of the valley little streams slipped out of the hill canyons and fell into the bed of the Salinas River. In the winter of wet years the streams ran full-freshet, and they swelled the river until sometimes it raged and boiled, bank full, and then it was a destroyer. The river tore the edges of the farm lands and washed whole acres down; it toppled barns and houses into itself, to go floating and bobbing away. It trapped cows and pigs and sheep and drowned them in its muddy brown water and carried them to the sea. Then when the late spring came, the river drew in from its edges, and the sand banks appeared. And in the summer the river didn't run at all above ground. Some pools would be left in the deep swirl places under a high bank. The tulips and grasses grew back, and willows straightened up with the flood debris in their upper branches. The Salinas was only a part-time river. The summer sun drove it underground—it was not a fine river at all, but it was the only one we had and so we boasted about it how dangerous it was in a wet winter and how dry it was in a dry summer. You can boast about anything if it's all you have. Maybe the less you have, the more you are required to boast.

The floor of the Salinas Valley, between the ranges and below the foothills, is level because this valley used to be the bottom of a hundred-mile inlet from the sea. The river mouth at Moss Landing was centuries ago the entrance to this long inland water. Once, fifty miles down the valley, my father bored a well. The drill came up first with topsoil and then with gravel and then with white sea sand full of shells and even pi...



Illustrations

Collaborative Filtering, Matrix Completion

Sketch of application: image inpainting



Illustrations

Collaborative Filtering, Matrix Completion

Sketch of application: image inpainting



Illustrations

Collaborative Filtering, Matrix Completion

Sketch of application: image inpainting



Illustrations

Collaborative Filtering, Matrix Completion

Sketch of application: matrix completion



Illustrations

Collaborative Filtering, Matrix Completion

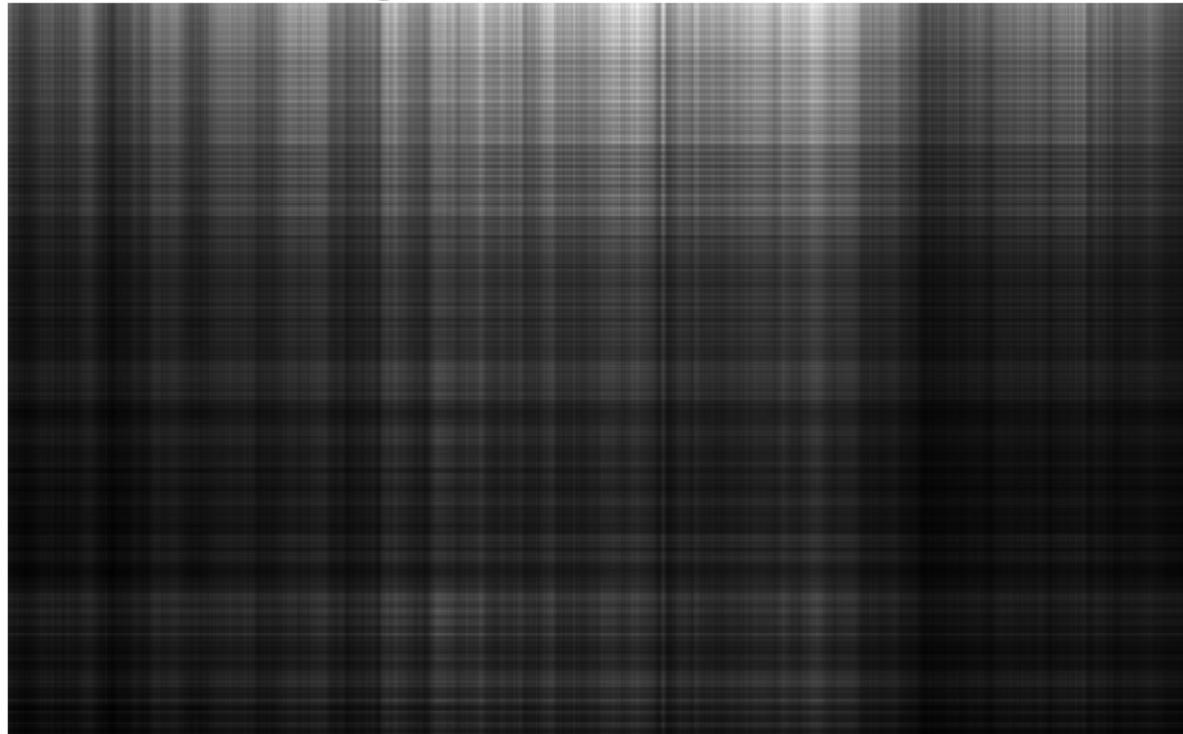
We only observe 35% of the picture



Illustrations

Matrix Completion

Solutions of increasing rank 2



Illustrations

Matrix Completion

Solutions of increasing rank 49



Illustrations

Matrix Completion

Solutions of increasing rank 200.



Illustrations

Matrix Completion: a quick overview of groundbreaking theory

Exact reconstruction (no noise)

$$\hat{R} \in \operatorname{argmin}\{\|M\|_* \text{ such that } \mathcal{P}_E(M) = \mathcal{P}_E(R)\}$$

Assume $n = n_U = n_I$ for short and put $m = |E|$. Then under some assumptions

- No method can succeed if $m \leq crn \log n$. Namely, need **at least**

$$m \geq crn \log n$$

observed entries to recover M and $r = \operatorname{rank}$

- If $m \geq crn(\log n)^2$ then reconstruction is **exact!** with a large probability
- In this setting, convex relaxation is **exact: no loss** when relaxing rank by $\|\cdot\|_*$
- Gives the exact same solution as the one constrained by rank!
- Convex programming incredibly powerful in this case
- Compressed sensing theory (See Emmanuel Candès's work)

Conclusion and perspectives

Outline

- 1 Introduction to recommendation systems
- 2 Measures of success, preprocessing
- 3 K-NN approach to CF
- 4 Matrix Factorization
 - Intro
 - Link with PCA
 - A convex formulation
 - Proximal gradient descent
- 5 Illustrations
 - A groundbreaking theory
- 6 Conclusion and perspectives
- 7 References

Conclusion and perspectives

NetflixWinner

Blend multiple variations of MF approaches and k-NN approaches through Gradient boosting decision trees including Matrix Factorization which behaves the best among base methods.

- The Belkor solution to the netflix grand prize, Yehuda Koren http://www.netflixprize.com/assets/GrandPrize2009_BPC_BellKor.pdf
- Matrix factorization techniques, Koren, Bell, Volinski, IEEE Computer 2009 <https://datajobs.com/data-science-repo/Recommender-Systems->

- 1 Introduction to recommendation systems
- 2 Measures of success, preprocessing
- 3 K-NN approach to CF
- 4 Matrix Factorization
 - Intro
 - Link with PCA
 - A convex formulation
 - Proximal gradient descent
- 5 Illustrations
 - A groundbreaking theory
- 6 Conclusion and perspectives
- 7 References

References

References

- Recommender systems: slides/tutorial Amatriain
- The Belkor solution to the netflix grand prize, Yehuda Koren http://www.netflixprize.com/assets/GrandPrize2009_BPC_BellKor.pdf
- Matrix factorization techniques, Koren, Bell, Volinski, IEEE Computer 2009 <https://datajobs.com/data-science-repo/Recommender-Systems->
- Beyond MF, Compressed sensing: <http://dsp.rice.edu/cs>

- 4 students
- A non trivial problem (no pure classification, no pure regression): quantile regression, active learning, ranking, recommendation, time series modeling, churn,...
- A short presentation 2/3 pages of the problem and state of the art
- Description/discussion of at least 2 methods , referenced in papers or of your own
- Description of evaluation metrics, model selection protocole
- toy dataset + two real datasets : analysis of performance of at least 2 methods
- Conclusion, perspectives

- before Dec 15, 2016: submit the problem to solve (on the website) - you receive a go-nogo answer.
- before Jan 6, 2017: (reportv1) short presentation of the problem and state of the art
- before Jan 31, 2017: (reportv2) Full report + code-ready-for-demo, dataset available