Recommender Systems

Dr. Fayyaz ul Amir Afsar Minhas

PIEAS Biomedical Informatics Research Lab

Department of Computer and Information Sciences

Pakistan Institute of Engineering & Applied Sciences

PO Nilore, Islamabad, Pakistan

http://faculty.pieas.edu.pk/fayyaz/

Slides adapted from Dr. Andrew Ng's videos on Recommender Systems
https://www.youtube.com/playlist?list=PL_npY1DYXHPT-3dorG7Em6d18P4JRFDvH

Recommenders

- Task
  - Predict the rating or preference a user would give an item
- Given:
  - Training data
    - Items rated by users
- Issues
  - New users
  - New items
  - Cold Start
- Other names
  - Matrix Completion Problem
  - Information Filtering Problem

https://en.wikipedia.org/wiki/Cold_start

Examples

MOVIE / USERS	Alice (1)	Bob (2)	Carol (3)	Dave (4)
Love at last	5	5	0	0
Romance Forever	5	**?**	**?**	0
Cute puppies of love	**?**	4	0	**?**
Nonstop car chases	0	0	5	4
Swords vs. Karate	0	0	5	**?**

Applications

- Youtube
- Facebook
- Amazon
- Last.fm
- Netflix
  - Competition: 2006-2009 $1, 100M movie ratings, 107 algorithms
- Pandora

https://en.wikipedia.org/wiki/Recommender_system

How would you solve this problem?

Existing Approaches

- Content Based
  - Extract Features for each item
- Collaborative (Filtering)
  - No features needed
- Hybrid Approaches

Content Based Filtering: Representation

- Extract Features for each item
  - $\boldsymbol{x}^i = [1 \quad x_1^i \quad \dots \quad x_K^i], i = 1 \dots m$
- Assign a weight (parameter) vector for each user
  - $\boldsymbol{w}^j = [w_k^j], j = 1 \dots u, k = 0 \dots K$
- The "score" of a user for an item is then given by
  - $f_{ij} = \left(\boldsymbol{w}^j\right)^T \boldsymbol{x}^i$

MOVIE / USERS	Alice (1)	Bob (2)	Carol (3)	Dave (4)	(romance) x_1	(action) x_2
Love at last	5	5	0	0	0.90	0.00
Romance Forever	5	**?**	**?**	0	1.0	0.01
Cute puppies of love	**?**	4	0	**?**	0.99	0.00
Nonstop car chases	0	0	5	4	0.10	1.00
Swords vs. Karate	0	0	5	**?**	0.00	0.90

Content Based Filtering: Evaluation

- Applying principal of structural risk minimization
- Minimize Empirical Error
  - Define Empirical Error
    - For user j, let y_{ij} be the rating of item i if the user has rated that item
      - Let's use r_{ij} as an indicator variable to indicate if the user has rated the item $\left(r_{ij} = 1\right)$ or not $\left(r_{ij} = 0\right)$
      - The empirical error term can be written as (similar to regression)

$$E = \sum_{j=1}^{u} \sum_{i=1:r_{ij}=1}^{m} \left(\left(\boldsymbol{w}^{j}\right)^{T} \boldsymbol{x}^{i} - y_{ij}\right)^{2}$$

- Regularize

$$R = \sum_{j=1}^{u} \sum_{k=1}^{K} \left(w_{k}^{j}\right)^{2}$$

Content Based Filtering: Optimization

- The complete optimization problem becomes

$$min_{\boldsymbol{w}^j, j=1\dots u} \frac{1}{2}\sum_{j=1}^{u}\sum_{i=1:r_{ij}=1}^{m}\left(\left(\boldsymbol{w}^j\right)^T\boldsymbol{x}^i - y_{ij}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{u}\sum_{k=1}^{K}\left(w_k^j\right)^2$$

- This problem can be solved using Gradient Descent

$$w_k^j := w_k^j - \alpha \nabla w_k^j$$

Write the expression for ∇w_k^j

Solution

- $\nabla w_0^j = \sum_{i=1:r_{ij}=1}^{m} \left(\left(\boldsymbol{w}^j\right)^T \boldsymbol{x}^i - y_{ij} \right)$
- $\nabla w_k^j = \sum_{i=1:r_{ij}=1}^{m} \left(\left(\boldsymbol{w}^j\right)^T \boldsymbol{x}^i - y_{ij} \right) x_k^i + \lambda w_k^j$
  - For k=1…K

Observations and Problems

- What do the weight/parameter vectors signify?
- Issues
  - Linear
  - New Users
  - New Items
  - Loss function improvements
  - Regularization improvements
  - Feature Extraction
    - Requires analysis of the content

Collaborative Filtering

- We don't know the features

MOVIE / USERS	Alice (1)	Bob (2)	Carol (3)	Dave (4)	(romance) x_1	(action) x_2
Movie 1	5	5	0	0	?	?
Movie 2	5	**?**	**?**	0	?	?
Movie 3	**?**	4	0	**?**	?	?
Movie 4	0	0	5	4	?	?
Movie 5	0	0	5	**?**	?	?

- Let's ask the user how much they like "romance" or "action"
  - $\boldsymbol{w}^1 = \begin{bmatrix}0\\5\\0\end{bmatrix}, \boldsymbol{w}^2 = \begin{bmatrix}0\\5\\0\end{bmatrix}, \boldsymbol{w}^3 = \begin{bmatrix}0\\0\\5\end{bmatrix}, \boldsymbol{w}^4 = \begin{bmatrix}0\\0\\5\end{bmatrix}$

How can we find the features?

- Now we want to assign values to the features $\boldsymbol{x}^i$ of each movie such that

  $(\boldsymbol{w}^j)^T \boldsymbol{x}^i \approx y_{ij}$ for all $r_{ij} = 1$

  – We want the features such that the given user preferences *explain* the data

Application of SRM

- Given: $\boldsymbol{w}^1, \dots, \boldsymbol{w}^u$
- Find: $\boldsymbol{x}^i = [1 \quad x_1^i \quad \dots \quad x_K^i], i = 1 \dots m$
- So as to solve the SRM structured as follows

$$min_{\boldsymbol{x}^i, i=1\dots m} \frac{1}{2}\sum_{j=1}^{u} \sum_{i=1:r_{ij}=1}^{m} \left(\left(\boldsymbol{w}^j\right)^T \boldsymbol{x}^i - y_{ij}\right)^2 + \frac{\lambda}{2}\sum_{i=1}^{m}\sum_{k=1}^{K}\left(x_k^i\right)^2$$

  – Error
  – Regularization

- Can be solved using gradient descent

$$x_k^i := x_k^i - \alpha \nabla x_k^i$$

Collaborative Filtering

- Given the movie features (and the movie ratings)
  - We can estimate the user parameters (weights)
- Given the user-parameters (and the movie ratings)
  - We can estimate the movie features
- We can, ask the users for initial preferences and then estimate the features and then update the preferences and keep going until convergence

$$\boldsymbol{w} \rightarrow \boldsymbol{x} \rightarrow \boldsymbol{w} \rightarrow \boldsymbol{x} \rightarrow \cdots$$

Collaborative Filtering

- Given $\boldsymbol{x}^i, i = 1 \ldots m$ estimate $\boldsymbol{w}^j, j = 1 \ldots u$

$$min_{\boldsymbol{w}^j, j=1\ldots u} \frac{1}{2}\sum_{j=1}^{u}\sum_{i=1:r_{ij}=1}^{m}\left(\left(\boldsymbol{w}^j\right)^T\boldsymbol{x}^i - y_{ij}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{u}\sum_{k=1}^{K}\left(w_k^j\right)^2$$

- Given $\boldsymbol{w}^j, j = 1 \ldots u$ estimate $\boldsymbol{x}^i, i = 1 \ldots m$, x_0^i=1

$$min_{\boldsymbol{x}^i, i=1\ldots m} \frac{1}{2}\sum_{j=1}^{u}\sum_{i=1:r_{ij}=1}^{m}\left(\left(\boldsymbol{w}^j\right)^T\boldsymbol{x}^i - y_{ij}\right)^2 + \frac{\lambda}{2}\sum_{i=1}^{m}\sum_{k=1}^{K}\left(x_k^i\right)^2$$

Simultaneously solve with x_0^i=1

$$min_{\boldsymbol{w}^j, j=1\ldots u, \boldsymbol{x}^i, i=1\ldots m} \frac{1}{2}\sum_{j=1}^{u}\sum_{i=1:r_{ij}=1}^{m}\left(\left(\boldsymbol{w}^j\right)^T\boldsymbol{x}^i - y_{ij}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{u}\sum_{k=1}^{K}\left(w_k^j\right)^2 + \frac{\lambda}{2}\sum_{i=1}^{m}\sum_{k=1}^{K}\left(x_k^i\right)^2$$

Optimization Algorithm

- Initialize (to small random values) $\boldsymbol{x}^i, i = 1 \dots m$ and $\boldsymbol{w}^j, j = 1 \dots u$
- Minimize the objective through gradient descent over the objective function
  - Multiple Iterations

– Solve for unknown movie ratings!

What can we do with this?

- Why is it called collaborative filtering?
- We can rank the movies that were not ranked by a user
- We can also identify similar movies
    - Nearest neighbors over $\boldsymbol{x}^i$
- Or similar users
    - Nearest neighbors over $\boldsymbol{w}^j$
- Or identify popular trends of movies
  – Average movie ratings across all users

Practical issues

- Normalization of means
- Different regularization parameters for movies and users
- Concept drift
  - Users can change over time
- Addition of a user?
- Addition of a movie?
- Non-linear behavior?
- How should we choose “K”?
  - Have features that are given by the users
  - And other features too

Hybrid Approaches

- Basilico, Justin, and Thomas Hofmann. 2004. “Unifying Collaborative and Content-Based Filtering.” In *Proceedings of the Twenty-First International Conference on Machine Learning*, 9 – . ICML ’04. New York, NY, USA: ACM. doi:10.1145/1015330.1015394.

Matrix Completion

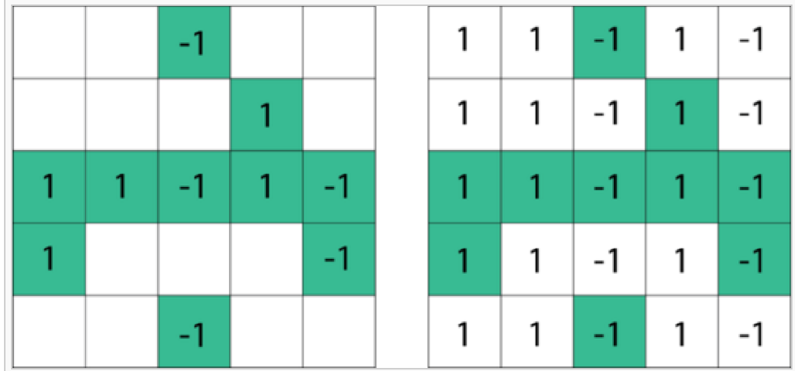
- This problem is also called matrix completion
- Low rank matrix completion

$$min_X \, rank(X)$$

Such that:

$$X_{ij} = M_{ij} \; \forall (i,j) \; with \; r_{ij} = 1$$

- Applications
  - Collaborative Filtering
  - System Identification



Matrix completion of a partially revealed 5 by 5 matrix with rank-1. Left: observed incomplete matrix; Right: matrix completion result.

https://en.wikipedia.org/wiki/Matrix_completion

- Abernethy, Jacob, Francis Bach, Theodoros Evgeniou, and Jean-Philippe Vert. 2009. “A New Approach to Collaborative Filtering: Operator Estimation with Spectral Regularization.” *J. Mach. Learn. Res.* 10 (June): 803–26.

Required Reading

- Slides adapted from Dr. Andrew Ng's videos on Recommender Systems
- https://www.youtube.com/playlist?list=PL_npY1DYXHPT-3dorG7Em6d18P4JRFDvH

Collaborative Filtering in Python

- Spark-MLIB
- Others

We want to make a machine that will be proud of us.

- Danny Hillis