# Multi-purpose Library of Recommender System Algorithms for the Item Prediction Task

Presentation of my Bachelor Thesis

Julius Kolbe

L3S Research Center / Leibniz University of Hanover
Hannover, Germany

July 1, 2013

# Contents

# Contents

# Implicit Feedback

| | Anna | Berta | Claudia | Dagmar |
|---|---|---|---|---|
| The Shawshank Redemption | 1 | | 1 | |
| The Godfather | | 1 | 1 | |
| The Godfather: Part II | | 1 | | 1 |
| Pulp Fiction | 1 | 1 | | 1 |
| The Good, the Bad and the Ugly | 1 | | 1 | |

# Item Prediction Task

| | Anna | Berta | Claudia | Dagmar |
|---|---|---|---|---|
| The Shawshank Redemption | 1 | | 1 | ? |
| The Godfather | | 1 | 1 | ? |
| The Godfather: Part II | | 1 | | 1 |
| Pulp Fiction | 1 | 1 | | 1 |
| The Good, the Bad and the Ugly | 1 | | 1 | ? |

# Notation

| | Anna | Berta | Claudia | Dagmar |
|---|---|---|---|---|
| The Shawshank Redemption | 1 | | 1 | |
| The Godfather | | 1 | 1 | |
| The Godfather: Part II | | 1 | | 1 |
| Pulp Fiction | 1 | 1 | | 1 |
| The Good, the Bad and the Ugly | 1 | | 1 | |

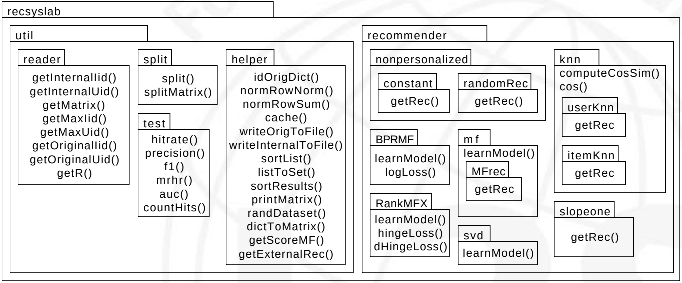
Items
Users
Interactions
Basket of $u$

# Contents

# Motivation for recsyslab

- Python for easy readable source code
- simple usage
- for education
- for research
- open source license: GPLv3

# General Structure



# Get recsyslab

github.com/Foolius/recsyslab

github.com/Foolius/recsyslab/archive/master.zip

```
$ git clone
    https://github.com/Foolius/recsyslab.git
```

# Contents

# k-Nearest-Neighbor [2]

1. Compute similarity of each item, item pair
2. For each item, save the $k$ items with the highest similarity ($=$ neighbors)
3. Compute the union of the neighbors of the basket of $u$
4. For each item in this set compute the sum of similarities to the basket of $u$
5. Sort by this score and return the first $N$ items

$$\text{sim}(i, j) = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{||\vec{i}||_2 ||\vec{j}||_2} \tag{1}$$

# k-Nearest-Neighbor [2]

| | Anna | Berta | Claudia | Dagmar |
|---|---|---|---|---|
| The Shawshank Redemption | 1 | 0 | 1 | 0 |
| The Godfather | 0 | 1 | 1 | 0 |
| The Godfather: Part II | 0 | 1 | 0 | 1 |
| Pulp Fiction | 1 | 1 | 0 | 1 |
| The Good, the Bad and the Ugly | 1 | 0 | 1 | 0 |

$$\text{sim}(i,j) = \cos(\vec{i},\vec{j}) = \frac{\vec{i}\cdot\vec{j}}{||\vec{i}||_2||\vec{j}||_2} = \frac{0}{2} = 0$$

# k-Nearest-Neighbor [2]

| | Anna | Berta | Claudia | Dagmar |
|---|---|---|---|---|
| The Shawshank Redemption | 1 | 0 | 1 | 0 |
| The Godfather | 0 | 1 | 1 | 0 |
| The Godfather: Part II | 0 | 1 | 0 | 1 |
| Pulp Fiction | 1 | 1 | 0 | 1 |
| The Good, the Bad and the Ugly | 1 | 0 | 1 | 0 |

$$\text{sim}(i,j) = \cos(\vec{i},\vec{j}) = \frac{\vec{i}\cdot\vec{j}}{||\vec{i}||_2||\vec{j}||_2} = \frac{2}{\sqrt{2}\sqrt{3}}$$

# Matrix Factorization [1]

| | Anna | Berta | Claudia | Dagmar |
|---|---|---|---|---|
| The Shawshank Redemption | 1 | 0 | 1 | 0 |
| The Godfather | 0 | 1 | 1 | 0 |
| The Godfather: Part II | 0 | 1 | 0 | 1 |
| Pulp Fiction | 1 | 1 | 0 | 1 |
| The Good, the Bad and the Ugly | 1 | 0 | 1 | 0 |

Find $W$ and $H$ so: $\hat{M} = W\ H^\top$.

$$\text{Score}(u, i) = W_u I_i^\top. \tag{2}$$

# Matrix Factorization, Training

```
U = randomly chosen user
I = randomly chosen item U interacted with
J = randomly chosen item U did not interact with

X=H[i] - H[j]
wx = dot product of W[u] and X
dloss = (derivative of the
        loss function of wx and 1) *
        learningRate

W[u] += dloss * (H[i] - H[j]) #These three lines
H[i] += dloss * W[u]          #have to be
H[j] += dloss * -W[u]         #executed at once
```

```
u = random.choice(R.keys())

userItems = [x[0] for x in R[u]]
# the positive example
i = userItems[np.random.random_integers(0, len(userItems) - 1)]
# the negative example
j = np.random.random_integers(0, m_items)
# if  j is also relevant for u we continue
# we need to see a negative example to contrast the positive one
while j in userItems:
    j = np.random.random_integers(0, m_items)

X = H[i] - H[j]
wx = np.dot(W[u], X)
dloss = dlossF(wx, y)

# temp
wu = W[u]
hi = H[i]
hj = H[j]
if dloss != 0.0:
    # Updates
    eta_dloss = learningRate * dloss
    W[u] += eta_dloss * (hi - hj)
    H[i] += eta_dloss * wu
    H[j] += eta_dloss * (-wu)
    W[u] *= scaling_factorU
    H[i] *= scaling_factorI
    H[j] *= scaling_factorJ
```

# Contents

# Leave-one-out Protocol

1. Randomly choose one interaction per user and hide them
2. Train the recommender system with the remaining interactions
3. Get recommendations for every user
4. Compute the chosen evaluation metric with the hidden items and the recommendations

# Hitrate/Recall@N [2, 5]

$$\text{Recall@N} = \frac{\sum_{u \in U} H_u \cap \text{topN}_u}{|H|} \tag{3}$$

| | |
|---|---|
| $H$ | hidden interactions |
| $H_u$ | the hidden interaction of $u$ |
| $U$ | set of users |
| $topN_u$ | N recommendations for $u$ |

# Precision [5]

$$\text{Precision} = \frac{\sum_{u \in U} H_u \cap \text{topN}_u}{N \times |U|} \tag{4}$$

$H$ hidden interactions

$H_u$ the hidden interaction of $u$

$U$ set of users

$topN_u$ $N$ recommendations for $u$

# F1 [5]

$$\text{F1} = \frac{2 \times \text{Recall@N} \times \text{Precision}}{\text{Recall@N} + \text{Precision}}. \tag{5}$$

| | |
|---|---|
| $H$ | hidden interactions |
| $H_u$ | the hidden interaction of $u$ |
| $U$ | set of users |
| $topN_u$ | $N$ recommendations for $u$ |

# Mean Reciprocal Hitrate [3]

$$\mathsf{MRHR} = \frac{1}{|U|} \sum_{u \in U} \frac{1}{\mathsf{pos}(\mathsf{topN}_u, H_u)}, \tag{6}$$

$H$ hidden interactions

$H_u$ the hidden interaction of $u$

$U$ set of users

$topN_u$ $N$ recommendations for $u$

$\mathsf{pos}(\mathsf{topN}_u, H_u)$ position of the hidden item in the list of recommendations

# Area under the ROC (AUC) [4]

$$\text{AUC} = \frac{1}{|U|} \sum_{u \in U} \frac{1}{|E(u)|} \sum_{(i,j) \in E(u)} \delta(x_{ui} > x_{uj}), \tag{7}$$

$$\delta(x) = \begin{cases} 1, & \text{if x is true,} \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

$$E(u) = \{(i,j) | (u,i) \in H \wedge (u,j) \notin (H \cup T)\}. \tag{9}$$

$H$ hidden interactions

$H_u$ the hidden interaction of $u$

$U$ set of users

$x_{ui}$ predicted score of the interaction between $u$ and $i$

# Contents

# Demonstration of recsyslab

Matrix factorization, June 2013.

G. Karypis.
Evaluation of item-based top-n recommendation algorithms.
In *Proceedings of the tenth international conference on Information and knowledge management*, CIKM '01, pages 247–254, New York, NY, USA, 2001. ACM.

X. Ning and G. Karypis.
Slim: Sparse linear methods for top-n recommender systems.
In D. J. Cook, J. Pei, W. Wang, O. R. Zaïane, and X. Wu, editors, *ICDM*, pages 497–506. IEEE, 2011.

S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme.
Bpr: Bayesian personalized ranking from implicit feedback.

In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 452–461, Arlington, Virginia, United States, 2009. AUAI Press.

B. M. Sarwar, G. Karypis, J. A. Konstan, and J. T. Riedl.
Application of dimensionality reduction in recommender system – a case study.
In *IN ACM WEBKDD WORKSHOP*, 2000.