Feature-based factorized Bilinear Similarity Model for Cold-Start Top-n Item Recommendation

Mohit Sharma [1] Jiayu Zhou [2] Junling Hu [2] George Karypis [1]

[1]University of Minnesota

[2]Samsung Research America

April 30, 2015

Recommender systems

- Recommender systems help the users to discover relevant items based on their preferences.
- They help the users to avoid exhaustive search through the entire catalog of items.

Item cold-start recommendation

- Traditional methods rely on historical preferences on the items provided by the users.
- These methods fail for the items with no prior preferences e.g., a newly published book or a recently released movie.
- Item cold-start methods rely on features of the items to generate the recommendations for the users.
  - e.g., book/movie description (genre, writer, characters, plot etc).

Existing approaches

- Estimate the preference of a user (u) on a new item (i) by aggregating the similarities of the items (j) that the user preferred in the past.

$$\tilde{r}_{u,i} = \sum_{j \in \mathcal{R}_u^+} sim(i,j),$$

- $\tilde{r}_{u,i}$ - estimated preference of user u on item i.
- $\mathcal{R}_u^+$ - set of all the items preferred by the user in the past.
- $sim(i,j)$ - computes similarity between the items i and j.
- The similarity function can be static e.g., cosine or jaccard similarity or it can be learned from the data.

User-specific Feature-based Similarity Models (UFSM)[1]

$$sim(i,j) = \mathbf{w}^T(\mathbf{f}_i \odot \mathbf{f}_j)$$

- f_i is n dimensional feature vector of the item i.
- $\odot$ is the element-wise Hadamard product operator.
- **w** determines the features' contribution towards similarity and it is estimated from the data by minimizing loss function.

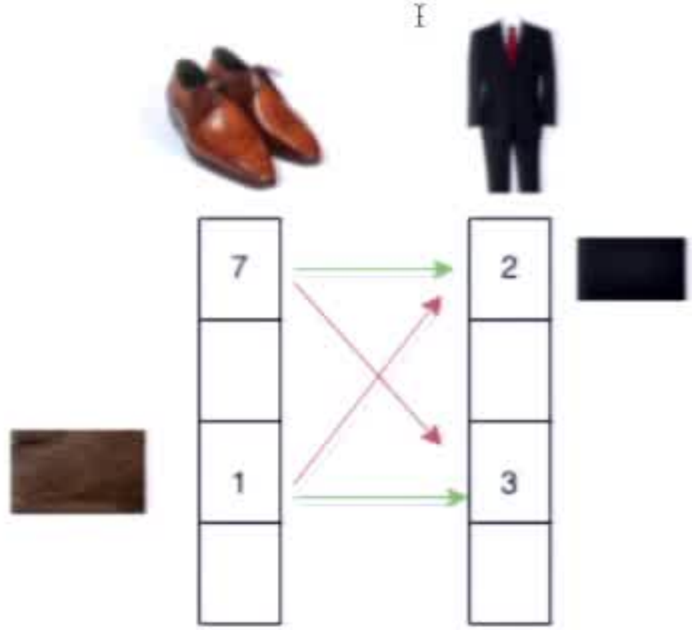
$$\mathcal{L}_{bpr}(w) \equiv -\sum_{u \in U} \sum_{\substack{i \in \mathcal{R}_u^+, \\ j \in \mathcal{R}_u^-}} \ln \sigma(\tilde{r}_{u,i}(w) - \tilde{r}_{u,j}(w)),$$

- In original paper, multiple similarity functions were used but single similarity function often gave the best performance on datasets.

[1] **Elbadrawy and Karypis, TIST'15.**

Limitations of linear model

- Linear similarity model can not learn cross interaction between features e.g., whether user prefers to wear leather shoes (fabric) with black clothes (color).



Bilinear model

$$sim(i,j) = f_i^T W f_j$$

- f_i - feature vector of item i.
- $W \in \mathcal{R}^{n \times n}$, off-diagonal elements contains cross interaction feature weights.
- W is learned as part of learning the model.
- Data is generally not sufficient to reliably estimate the interaction between all possible pair of features i.e., n^2 elements in W.
- The number of elements that needs to be estimated grows quadratically with the dimension of features.

Factorized bilinear similarity

- Represent W as sum of the diagonal weights and the low-rank approximation of off-diagonal weights.

$$sim(i,j) = f_i^T W f_j = f_i^T (D + V^T V) f_j$$

- $D \in \mathcal{R}^{n \times n}$ is a diagonal matrix with the diagonal elements denoted as vector d.
- $V \in \mathcal{R}^{h \times n}$ is a low-rank matrix with the column v_p representing latent factor of the feature p.
- The number of elements to be estimated is $(n + nh) << n^2$.

Factorized bilinear similarity model (FBSM)

- Preference of the user u for a new item i is given by aggregation of *bilinear* similarity with the items preferred in past:

$$\tilde{r}_{u,i} = \sum_{j \in \mathcal{R}_u^+} sim(i,j) = \sum_{j \in \mathcal{R}_u^+} f_i^T W f_j,$$
$$= \sum_{j \in \mathcal{R}_u^+} f_i^T D f_j + f_i^T V^T V f_j$$

  - f_i - feature vector of item i.
  - $\mathcal{R}_u^+$ is the set of items preferred by the user u in the past.

Model estimation

- Stochastic gradient descent update for a sampled pair of preferred item i and non-preferred item j of the user u.

$$D = D + \alpha\Big(\frac{1}{1+e^{\tilde{r}_{u,ij}}}\nabla_D \tilde{r}_{u,ij} - \beta D\Big),$$
$$v_p = v_p + \alpha\Big(\frac{1}{1+e^{\tilde{r}_{u,ij}}}\nabla_{v_p} \tilde{r}_{u,ij} - \lambda v_p\Big)$$

where,

- $\tilde{r}_{u,ij} = \tilde{r}_{u,i} - \tilde{r}_{u,j}$, represents the relative rank of the preferred item i w.r.t the non-preferred item j for the user u.
- v_p is the p^{th} column of matrix V, which corresponds to the latent factor of the feature p.
- α is the learning rate.

Datasets

Dataset	user	item	features	preferences	density(%)
CUL	3,272	21,508	6,359	180,622	0.13
BX	17,219	36,546	8,946	574,127	0.09
AMAZON	13,097	11,077	5,766	175,612	0.12
ML-IMDB	2,113	8,645	8,744	739,973	4.05
ML-HR(genre)	2,113	10,109	20	855,598	4.01

Comparison methods

- **Cosine Similarity (CoSim)**
  Non-collaborative method using cosine similarity.
- **User-specific Feature Based Similarity Model (UFSM)**
  Collaborative method similar to FBSM with no cross-feature interactions.
- **Regression Based Latent Factor Model (RLFM)**
  Collaborative method which transforms the item's features into item's latent factor. We used the method implemented in the factorization machine library LibFM which also accounts for the inter-feature interactions (RLFMI).

$$\tilde{r}_{(u,i)RLFM} = \alpha + b^T f_i + p_u^T A f_i$$

$$\tilde{r}_{(u,i)RLFMI} = \alpha + b^T f_i + p_u^T \sum_k f_{ik} v_k + \sum_k \sum_j f_{ik} f_{ij} v_i^T v_j$$

Evaluation methodology

- Split the user-item preference matrix R into three matrices i.e., R_{train}, R_{test} and R_{val}.
- These matrices are formed by splitting items (columns) of matrix R in 60%, 20% and 20% splits.
- Models are learned using R_{train} and best model is selected using R_{val}.
- Selected model is then used to estimate preferences over all the items in R_{test}.
- For each user the test items are sorted in decreasing order of estimated preference and the first n items are returned as Top-n recommendations for each user.

Metrics

- Recall at n

$$REC@n = \frac{|\{\text{Items liked by user}\} \cap \{\text{Top-n items}\}|}{|\text{Top-n items}|}$$

- Discounted cumulative gain at n

$$DCG@n = imp_1 + \sum_{p=2}^{n} \frac{imp_p}{\log_2(p)},$$

where the importance score imp_p of the item with rank p in the Top-n list is

$$imp_p = \begin{cases} 1/n, & \text{if item at rank } p \in R^+_{u,test} \\ 0, & \text{if item at rank } p \notin R^+_{u,test}. \end{cases}$$

- The main difference between Rec@n and DCG@n is that DCG@n is sensitive to the rank of the items in the Top-n list.

Results

Method	CUL		BX	
	Rec@10	DCG@10	Rec@10	DCG@10
CoSim	0.1791	0.0684	0.0681	0.0119
RLFMI	0.0874	0.0424	0.0111	0.0030
$UFSM_{bpr}$	0.2017	0.0791	0.0774	0.0148
$FBSM_{bpr}$	0.2026	0.0792	0.0776	0.0148

Method	ML-IMDB		ML-HR		AMAZON	
	Rec@10	DCG@10	Rec@10	DCG@10	Rec@10	DCG@10
CoSim	0.0525	0.1282	0.0050	0.0199	0.1205	0.0228
RLFMI	0.0155	0.0455	0.0120	0.0466	0.0394	0.0076
$UFSM_{bpr}$	0.0937	0.2160	0.0074	0.0233	0.1376	0.0282
$FBSM_{bpr}$	0.0964	0.2270	0.0120	0.0418	0.1392	0.0284

why can't the bilinear model lead to significant improvements?

Datasets

Dataset	user	item	features	preferences	density(%)
CUL	3,272	21,508	6,359	180,622	0.13
BX	17,219	36,546	8,946	574,127	0.09
AMAZON	13,097	11,077	5,766	175,612	0.12
ML-IMDB	2,113	8,645	8,744	739,973	4.05
ML-HR(genre)	2,113	10,109	20	855,598	4.01

Metrics

- Recall at n

$$REC@n = \frac{|\{\text{Items liked by user}\} \cap \{\text{Top-n items}\}|}{|\text{Top-n items}|}$$

- Discounted cumulative gain at n

$$DCG@n = imp_1 + \sum_{p=2}^{n} \frac{imp_p}{\log_2(p)},$$

where the importance score imp_p of the item with rank p in the Top-n list is

$$imp_p = \begin{cases} 1/n, & \text{if item at rank } p \in R^+_{u,test} \\ 0, & \text{if item at rank } p \notin R^+_{u,test}. \end{cases}$$

- The main difference between Rec@n and DCG@n is that DCG@n is sensitive to the rank of the items in the Top-n list.

Evaluation methodology

- Split the user-item preference matrix R into three matrices i.e., R_{train}, R_{test} and R_{val}.
- These matrices are formed by splitting items (columns) of matrix R in 60%, 20% and 20% splits.
- Models are learned using R_{train} and best model is selected using R_{val}.
- Selected model is then used to estimate preferences over all the items in R_{test}.
- For each user the test items are sorted in decreasing order of estimated preference and the first n items are returned as Top-n recommendations for each user.

Datasets

Dataset	user	item	features	preferences	density(%)
CUL	3,272	21,508	6,359	180,622	0.13
BX	17,219	36,546	8,946	574,127	0.09
AMAZON	13,097	11,077	5,766	175,612	0.12
ML-IMDB	2,113	8,645	8,744	739,973	4.05
ML-HR(genre)	2,113	10,109	20	855,598	4.01

Factorized bilinear similarity model (FBSM)

- Preference of the user u for a new item i is given by aggregation of *bilinear* similarity with the items preferred in past:

$$\tilde{r}_{u,i} = \sum_{j \in \mathcal{R}_u^+} sim(i,j) = \sum_{j \in \mathcal{R}_u^+} {f_i}^T W f_j,$$
$$= \sum_{j \in \mathcal{R}_u^+} {f_i}^T D f_j + {f_i}^T V^T V f_j$$

  - f_i - feature vector of item i.
  - $\mathcal{R}_u^+$ is the set of items preferred by the user u in the past.

Factorized bilinear similarity

- Represent W as sum of the diagonal weights and the low-rank approximation of off-diagonal weights.

$$sim(i,j) = f_i^T W f_j = f_i^T (D + V^T V) f_j$$

- $D \in \mathcal{R}^{n \times n}$ is a diagonal matrix with the diagonal elements denoted as vector d.
- $V \in \mathcal{R}^{h \times n}$ is a low-rank matrix with the column v_p representing latent factor of the feature p.
- The number of elements to be estimated is $(n + nh) << n^2$.

Factorized bilinear similarity model (FBSM)

- Preference of the user u for a new item i is given by aggregation of *bilinear* similarity with the items preferred in past:

$$\tilde{r}_{u,i} = \sum_{j \in \mathcal{R}_u^+} sim(i,j) = \sum_{j \in \mathcal{R}_u^+} f_i^T W f_j,$$
$$= \sum_{j \in \mathcal{R}_u^+} f_i^T D f_j + f_i^T V^T V f_j$$

  - f_i - feature vector of item i.
  - $\mathcal{R}_u^+$ is the set of items preferred by the user u in the past.

Factorized bilinear similarity

- Represent W as sum of the diagonal weights and the low-rank approximation of off-diagonal weights.

$$sim(i,j) = f_i^T W f_j = f_i^T (D + V^T V) f_j$$

- $D \in \mathcal{R}^{n \times n}$ is a diagonal matrix with the diagonal elements denoted as vector d.
- $V \in \mathcal{R}^{h \times n}$ is a low-rank matrix with the column v_p representing latent factor of the feature p.
- The number of elements to be estimated is $(n + nh) << n^2$.

Factorized bilinear similarity model (FBSM)

- Preference of the user u for a new item i is given by aggregation of *bilinear* similarity with the items preferred in past:

$$\tilde{r}_{u,i} = \sum_{j \in \mathcal{R}_u^+} sim(i,j) = \sum_{j \in \mathcal{R}_u^+} f_i^T W f_j,$$
$$= \sum_{j \in \mathcal{R}_u^+} f_i^T D f_j + f_i^T V^T V f_j$$

  - f_i - feature vector of item i.
  - $\mathcal{R}_u^+$ is the set of items preferred by the user u in the past.

Factorized bilinear similarity

- Represent W as sum of the diagonal weights and the low-rank approximation of off-diagonal weights.

$$sim(i,j) = f_i^T W f_j = f_i^T (D + V^T V) f_j$$

- $D \in \mathcal{R}^{n \times n}$ is a diagonal matrix with the diagonal elements denoted as vector d.
- $V \in \mathcal{R}^{h \times n}$ is a low-rank matrix with the column v_p representing latent factor of the feature p.
- The number of elements to be estimated is $(n + nh) << n^2$.

Model estimation

- Bayesian personalized ranking (BPR) loss

$$\mathcal{L}_{bpr}(\Theta) \equiv -\sum_{u \in U} \sum_{\substack{i \in \mathcal{R}_u^+, \\ j \in \mathcal{R}_u^-}} \ln \sigma(\tilde{r}_{u,i}(\Theta) - \tilde{r}_{u,j}(\Theta)),$$

  - $\Theta = (D, V)$ represent the model parameters.
  - $\mathcal{R}_u^+$ is the set of items preferred by the user u.
  - $\mathcal{R}_u^-$ is the set of items **not** preferred by the user u.
  - σ is Sigmoid function i.e., $\sigma(x) = \frac{1}{(1+e^{-x})}$.
- It tries to rank a preferred item higher than a non-preferred item for the user.