Spectral Collaborative Filtering

(Lei Zheng, 2018)

Presented by Jiin Seo

December 7, 2018

Outline

1. Introduction
2. Definitions
3. Proposed Model
4. Experiments

Outline

1. Introduction

(Problem Definition)

- Given a user set $\mathcal{U}$ and an item set $\mathcal{I}$, for each user $u \in \mathcal{U}$ who has liked/clicked/viewed an item set $\mathcal{I}_u^+ \subseteq \mathcal{I}$, we aim to recommend a ranked list of items from $\mathcal{I}_u^-$ that are of interests to the user.
- The connectivity information of the graph $\Rightarrow$ alleviating the cold-start problem for CF.

Outline

1. Introduction
2. Definitions
3. Proposed Model
4. Experiments

2. Definitions

Def. 2.1 (BipartiteGraph)

- $\mathcal{B} = \{\mathcal{U}, \mathcal{I}, \mathcal{E}\}$: Bipartite user-item graph with N vertices and E edges
- $\mathcal{U}$, $\mathcal{I}$: disjoint vertex sets
- $\forall e \in \mathcal{E}$ has the form $e = (u, i)$ where $u \in \mathcal{U}$ and $i \in \mathcal{I}$ and denotes that user u has interacted with item i in the training set.

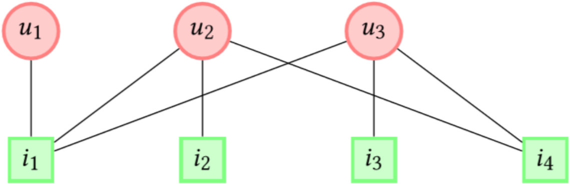


Figure: User-item bipartite graph $\mathcal{B}$ with edges representing observed user-item interactions.

2. Definitions

Def. 2.2 (Implicit Feedback Matrix)

$$\mathbf{R}_{r,j} = \begin{cases} 1, & \text{if } (u_r, i_j) \text{ interaction is observed.} \\ 0, & \text{otherwise.} \end{cases} \quad , \quad \mathbf{R} : |\,\mathcal{U}\,| \times |\,\mathcal{I}\,| \text{ matrix.} \tag{1}$$

Def. 2.3 (Adjacent Matrix)

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{R} \\ \mathbf{R}^T & \mathbf{0} \end{bmatrix} \quad , \quad \mathbf{A} : N \times N \text{ matrix.}$$

2. Definitions

Def. 2.4 (Laplacian Matrix) The random walk laplacian matrix **L** is

$$\mathbf{L} = \mathbf{I}\,`\,\mathbf{D}^{-1}\mathbf{A}$$

, where **D** is the $N \times N$ diagonal degree matrix defined as $\mathbf{D}_{nn} = \sum_j \mathbf{A}_{n,j}$.

Def. 3.1 (Graph Signal) Given any graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ and $\mathcal{E}$ are a vertex and an edge set, respectively, a graph signal is defined as a state vector $\mathbf{x} \in \mathbb{R}^{|\mathcal{V}| \times \mathbf{1}}$ over all vertices in the graph, where x_j is the j-th value of **x** observed at the j-th vertex of $\mathcal{G}$.

Outline

3. Proposed Model

[Spectral Collaborative Filtering]

- Graph fourier transform on a bipartite graph $\mathcal{B}$
- Spectral convolution filter on vertices (users and items)
- Polynomial approximation
- Multiple spectral convolution layers

3. Proposed Model

We assume that a graph signal $\mathbf{x} \in \mathbb{R}^{|\mathcal{V}| \times \mathbf{1}}$ is observed on a graph $\mathcal{G}$

Graph fourier transform and its inverse on $\mathcal{G}$:

$$\hat{x}(l) = \sum_{j=0}^{N-1} x(j)\mu_l(j) \quad \text{and} \quad x(j) = \sum_{l=0}^{N-1} \hat{x}(l)\mu_l(j),$$

,where $\boldsymbol{\mu}_l$ denotes the l-th eigenvector of $\mathbf{L}$; $\hat{\mathbf{x}}$ represents a graph signal which has been transformed into the spectral domain.

Matrix form of Graph fourier transform :

$$\hat{\mathbf{x}} = \mathbf{U}^T\mathbf{x} \quad \text{and} \quad \mathbf{x} = \mathbf{U}\hat{\mathbf{x}}$$

,where $\mathbf{U} = \{\boldsymbol{\mu}_0, \boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_l, \cdots, \boldsymbol{\mu}_{N-1}\}$ are eigenvectors of $\mathbf{L}$.

3. Proposed Model

Two types of graph signals for a bipartite graph $\mathcal{B}$

$\mathbf{x}^u \in \mathbb{R}^{|\mathcal{U}| \times \mathbf{1}}$ and $\mathbf{x}^i \in \mathbb{R}^{|\mathcal{I}| \times \mathbf{1}}$, associated with user and item vertices.

$$\begin{bmatrix} \hat{\mathbf{x}}^u \\ \hat{\mathbf{x}}^i \end{bmatrix} = \mathbf{U}^T \begin{bmatrix} \mathbf{x}^u \\ \mathbf{x}^i \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \mathbf{x}^u \\ \mathbf{x}^i \end{bmatrix} = \mathbf{U} \begin{bmatrix} \hat{\mathbf{x}}^u \\ \hat{\mathbf{x}}^i \end{bmatrix}.$$

3. Proposed Model

Spectral Convolution Filtering

Convolution filter, parameterized by $\boldsymbol{\theta} \in \mathbb{R}^N$, as $g_\theta(\boldsymbol{\Lambda}) = diag([\theta_0\lambda_0, \theta_1\lambda_1, \cdots \theta_{N-1}\lambda_{N-1}])$:

$$\begin{bmatrix} \mathbf{x}^u_{new} \\ \mathbf{x}^i_{new} \end{bmatrix} = \mathbf{U} g_\theta(\boldsymbol{\Lambda}) \begin{bmatrix} \hat{\mathbf{x}}^u \\ \hat{\mathbf{x}}^i \end{bmatrix} = \mathbf{U} g_\theta(\boldsymbol{\Lambda}) \mathbf{U}^T \begin{bmatrix} \mathbf{x}^u \\ \mathbf{x}^i \end{bmatrix}.$$

,where $\boldsymbol{\Lambda} = \{\lambda_0, \lambda_1, \cdots, \lambda_{N-1}\}$ denotes eigenvalues of the graph laplacian matrix $\mathbf{L}$.

3. Proposed Model

- Learning complexity of the filter is $\mathcal{O}(N)$.
- We can approximate the convolution filters by using first P polynomials as the following:

$$g_\theta(\mathbf{\Lambda}) \approx \sum_{p=0}^{P} \theta_p^{'} \mathbf{\Lambda}^p.$$

[Polynomial Approximation.]

The set of all convolution filters $\mathcal{S}_g = \{g_\theta(\mathbf{\Lambda}) = diag([\theta_0\lambda_0, \theta_1\lambda_1, \cdots \theta_{N-1}\lambda_{N-1}]), \boldsymbol{\theta} \in \mathbb{R}^N\}$ is equal to the set of finite-order polynomials $\mathcal{S}_h = \{h_{\boldsymbol{\theta}'}(\mathbf{\Lambda}) = \sum_{p=0}^{N-1} \theta_p^{'} \mathbf{\Lambda}^p, \boldsymbol{\theta}^{'} \in \mathbb{R}^N\}$.

3. Proposed Model

We limit the order of the polynomial, P, to 1

$$\begin{bmatrix} \mathbf{x}^u_{new} \\ \mathbf{x}^i_{new} \end{bmatrix} = (\theta^{`}_0 \mathbf{U}\mathbf{U}^T + \theta^{`}_1 \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T) \begin{bmatrix} \mathbf{x}^u \\ \mathbf{x}^i \end{bmatrix}.$$

By setting $\theta^{`}_0 = \theta^{`}_1 = \theta^{`}$, where $\theta^{`}$ is a scalar :

$$\begin{bmatrix} \mathbf{x}^u_{new} \\ \mathbf{x}^i_{new} \end{bmatrix} = \theta^{`} (\mathbf{U}\mathbf{U}^T + \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T) \begin{bmatrix} \mathbf{x}^u \\ \mathbf{x}^i \end{bmatrix}.$$

3. Proposed Model

Spectral convolution operation

- General version (C-dimensional graph signals)

$$sp(:; \mathbf{U}, \mathbf{\Lambda}, \mathbf{\Theta}^{`}) \equiv \begin{bmatrix} \mathbf{X}^{u}_{new} \\ \mathbf{X}^{i}_{new} \end{bmatrix} = \sigma(\mathbf{U}\mathbf{U}^{T} + \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{T}) \begin{bmatrix} \mathbf{X}^{u} \\ \mathbf{X}^{i} \end{bmatrix} \mathbf{\Theta}^{`})$$

, where $\mathbf{X}^{u} \in \mathbb{R}^{|\mathcal{U}| \times C}$, $\mathbf{X}^{i} \in \mathbb{R}^{|\mathcal{I}| \times C}$ and $\mathbf{\Theta} \in \mathbb{R}^{C \times F}$

3. Proposed Model

Spectral Collaborative Filtering (SpectralCF)

- K-layered deep spectralCF

$$\begin{bmatrix} \mathbf{X}_K^u \\ \mathbf{X}_K^i \end{bmatrix} = sp(\cdots sp(\begin{bmatrix} \mathbf{X}_0^u \\ \mathbf{X}_0^i \end{bmatrix}; \mathbf{U}, \mathbf{\Lambda}, \mathbf{\Theta}_0^{'}) \cdots ; \mathbf{U}, \mathbf{\Lambda}, \mathbf{\Theta}_{K-1}^{'})$$

- We concatenate them into our final latent factors .

$$\mathbf{V}^u = [\mathbf{X}_0^u, \mathbf{X}_1^u, \cdots, \mathbf{X}_K^u] \quad \text{and} \quad \mathbf{V}^i = [\mathbf{X}_0^i, \mathbf{X}_1^i, \cdots, \mathbf{X}_K^i]$$

where $\mathbf{V}^u \in \mathbb{R}^{|\mathcal{U}| \times (C+KF)}$ and $\mathbf{V}^i \in \mathbb{R}^{|\mathcal{I}| \times (C+KF)}$.

3. Proposed Model

Optimization and Prediction

- BPR loss

$$\mathcal{L} = \arg\min_{\mathbf{V}^u, \mathbf{V}^i} \sum_{(r,j,j^{'}) \in \mathcal{D}} -ln\sigma(\mathbf{v}_r^{u\,T}\mathbf{v}_j^i - \mathbf{v}_r^{u\,T}\mathbf{v}_{j^{'}}^i + \lambda_{reg}(||\mathbf{V}^u||_2^2 + ||\mathbf{V}^i||_2^2)$$

where $\mathcal{D}$ is generated as : $\mathcal{D} = \{(r, j, j^{'}) \mid r \in \mathcal{U} \wedge j \in \mathcal{I}_i^+ \wedge j^{'} \in \mathcal{I}_i^-\}$

- RMSprop
- The final item recommendation for a user r is given according to the ranking :

$$r : j_1 \gg j_2 \gg \cdots \gg j_n \Rightarrow v_r^{u\,T} v_{j_1}^i > v_r^{u\,T} v_{j_2}^i > \cdots > v_r^{u\,T} v_{j_n}^i$$

3. Proposed Model

Algorithm 1: SpectralCF

Input: Training set: $\mathcal{D} := \{(r, j, j') | r \in \mathcal{U} \wedge j \in \mathcal{I}_i^+ \wedge j' \subseteq \mathcal{I}_i^-\}$, number of epochs E, batch size B, number of layers K, dimension of latent factors C, number of filters F, regularization term λ_{reg}, learning rate λ, laplacian matrix $\boldsymbol{L}$ and its corresponding eigenvectors $\boldsymbol{U}$ and eigenvalues $\boldsymbol{\Lambda}$.

Output: Model's parameter set: $\Psi = \{\Theta_0', \Theta_1', \ldots, \Theta_{K-1}', \boldsymbol{X}_0^u, \boldsymbol{X}_0^i\}$.

```
Randomly initialize X_0^u and X_0^i from a Gaussian distribution
  N(0.01, 0.02);
for e = 1, 2, ···, E do
    Generate the e_th batch of size B by uniformly sampling from U,
      I_i^+ and I_i^-;
    for k = 0, 1, ···, K − 1 do
        Calculate X_{k+1}^u and X_{k+1}^i by using Eq. (10);
    end
    Concatenate [X_0^u, X_1^u, ..., X_K^u] into V^u and
      [X_0^i, X_1^i, ..., X_K^i] into V^i;
    Estimate gradients ∂L/∂Ψ_e by back propagation;
    Update Ψ_{e+1} according to the procedure of RMSprop
      optimization [29];
end
return Ψ_E.
```

Outline

4. Experiments

Datasets

- MovieLens-1M(1.0% density)
- HetRec(0.3% density)
- Amazon Instant Video(0.12% density)

Comparative Methods

- CF-based Models : ItemKNN, BPR, eALS, NCF
- Graph-based Models : GNMF, GCM
- Evaluating the performance of the top-M recommendations : MAP@M , Recall@M

4. Experiments

- SpectralCF always outperforms all comparative models regardless of the sparsities of the datasets.

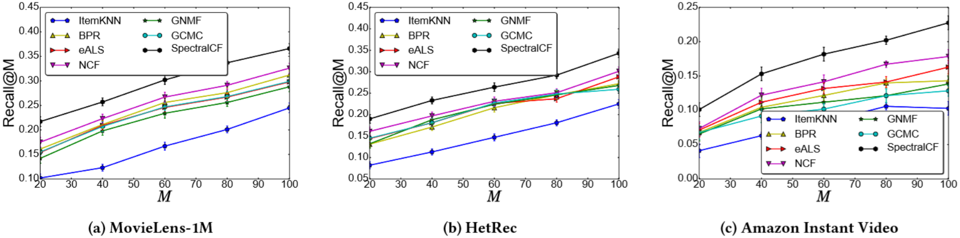
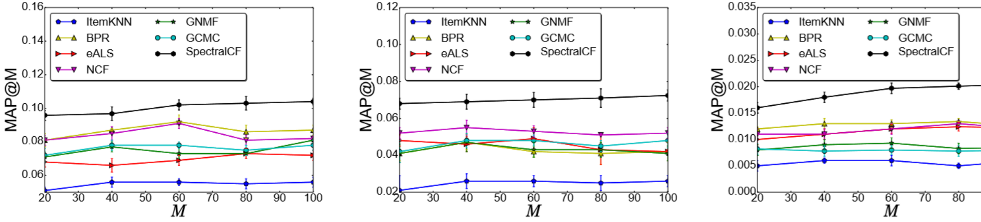


Figure: Performance comparison in terms of recall@M with M

4. Experiments

- SpectralCF always outperforms all comparative models regardless of the sparsities of the datasets.



(a) MovieLens-1M (b) HetRec (c) Amazon Instant Video

Figure: Performance comparison in terms of recall@M with M

4. Experiments

- SpectralCFcan better handle cold-start users and provide more reliable recommendations.

	P	1	2	3	4	5
Recall @20	BPR	0.021 (0.003)	0.029 (0.004)	0.031 (0.003)	0.034 (0.004)	0.038 (0.003)
	SpectralCF	**0.031** (0.003)	**0.039** (0.003)	**0.042** (0.002)	**0.045** (0.003)	**0.051** (0.003)
	Improvement	47.6%	34.5%	35.5%	32.4%	34.2%
MAP @20	BPR	0.014 (0.002)	0.017 (0.002)	0.021 (0.002)	0.024 (0.003)	0.027 (0.003)
	SpectralCF	**0.019** (0.002)	**0.024** (0.002)	**0.028** (0.003)	**0.031** (0.003)	**0.035** (0.002)
	Improvement	35.7%	41.2%	33.3%	29.2%	29.6%

Figure: Performance Comparison in the sparse training sets.
(P : Degrees of sparsity (1 5)