

Vasily インターン課題

山崎卓朗

東京大学大学院工学系研究科修士1年

課題1

本課題ではレコメンデーションアルゴリズムとして

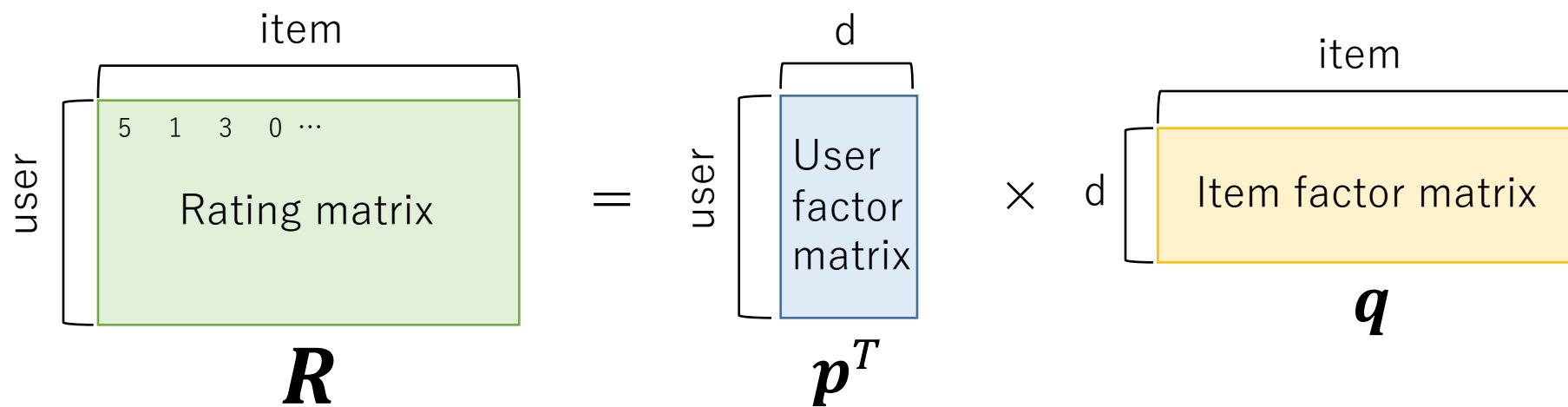
- Matrix Factorization(MF)
- Restricted Boltzmann Machine(RBM)
- Slope One

の3種類を実装した。ここではMatrix Factorizationについて説明を行う。

Matrix Factorization

■ Matrix Factorizationとは？

- Simon Funkが2006年に自身のブログで発表
- user-itemのrating 行列をuserとitemそれぞれの**特徴空間**にマッピングするアルゴリズムである.



d:dimension of factor space

Matrix Factorization

具体的には以下のような最小化問題を考える

$$\min_{\mathbf{p}, \mathbf{q}} \sum_{(u,i) \in S} (R_{ui} - \mathbf{p}_u^T \cdot \mathbf{q}_i)^2 + \lambda(\|\mathbf{p}_u\|^2 + \|\mathbf{q}_i\|^2)$$

S : ratingが行われたuser-itemペアの集合

値の更新は確率的勾配降下法(Stochastic gradient descent : SGD)を用いて行われる.
更新式は以下のとおり

$$\mathbf{E} = \mathbf{R} - \mathbf{p}^T \cdot \mathbf{q}$$

$$\Delta \mathbf{p} = \gamma(\mathbf{E} \cdot \mathbf{p} - \lambda \mathbf{q})$$

$$\Delta \mathbf{q} = \gamma(\mathbf{E} \cdot \mathbf{q} - \lambda \mathbf{p})$$

$$\mathbf{p} = \mathbf{p} + \Delta \mathbf{p}$$

$$\mathbf{q} = \mathbf{q} + \Delta \mathbf{q}$$

Matrix Factorization

疑似コードは以下のとおり

Algorithm 2.4 Matrix factorization - Stochastic gradient descent

Inputs: a training set \mathcal{S} , a learning rate γ

Outputs: two matrices P and Q such that $R_{\mathcal{S}} \approx PQ^T$

Initialize P and Q with zeroes or random values;

$n := 0$;

Compute the prediction error Err_n at iteration n ;

repeat

for $(u, i) \in \mathcal{S}$ **do**

$E(u, i) := R_{\mathcal{S}}(u, i) - P(u)^T Q(i)$;

$Q'(i) := Q(i)$;

$Q(i) := Q(i) + \gamma(E(u, i)P(u) - \lambda Q(i))$;

$P(u) := P(u) + \gamma(E(u, i)Q'(i) - \lambda P(u))$;

end for

$n := n + 1$;

 Compute the prediction error Err_n at iteration n ;

until $|Err_n - Err_{n-1}| < \epsilon$

← P と Q をランダムに初期化

誤差が閾値未満になる

← or 指定の反復回数に達する
まで値の更新を繰り返す

Matrix Factorization

- 今回実装したMFはこれまで説明したものにuser, itemごとのバイアスを追加したMF

■ バイアスとは？

- user, itemごとの評価傾向の違いのこと

例えば「aさんはbさんよりも厳しい評価をする」や「映画mは映画nよりも平均的に高い評価を得ている」など

- ユーザー u, アイテム iのバイアスは以下のように定義される

$$b_{ui} = \mu + b_u + b_i$$

全評価の平均

アイテムiに対する評価の平均-全評価の平均(μ)
→アイテム固有の評価傾向

ユーザーuの評価の平均-全評価の平均(μ)
→ユーザー固有の評価傾向

Matrix Factorization

対象となる最小化問題は以下のように変更

$$\min_{\mathbf{p}, \mathbf{q}} \sum_{(u,i) \in S} (R_{ui} - (\mathbf{p}_u^T \cdot \mathbf{q}_i + \mu + \mathbf{b}_u + \mathbf{b}_i))^2 + \lambda(\|\mathbf{p}_u\|^2 + \|\mathbf{q}_i\|^2 + \|\mathbf{b}_u\|^2 + \|\mathbf{b}_i\|^2)$$

S : ratingが行われたuser-itemペアの集合

SGDによる更新式も以下のように変更

$$\mathbf{E} = \mathbf{R} - (\mathbf{p}^T \cdot \mathbf{q} + \mu + \mathbf{b}_u + \mathbf{b}_i)$$

$$\mathbf{p} = \mathbf{p} + \Delta \mathbf{p} \quad (\Delta \mathbf{p} = \gamma(\mathbf{E} \cdot \mathbf{p} - \lambda \mathbf{q}))$$

$$\mathbf{q} = \mathbf{q} + \Delta \mathbf{q} \quad (\Delta \mathbf{q} = \gamma(\mathbf{E} \cdot \mathbf{q} - \lambda \mathbf{p}))$$

$$\mathbf{b}_u = \mathbf{b}_u + \Delta \mathbf{b}_u \quad (\Delta \mathbf{b}_u = \gamma(\mathbf{E}_u - \lambda \mathbf{b}_u)) \quad \mathbf{E}_u : \text{ユーザー} u \text{の平均error}$$

$$\mathbf{b}_i = \mathbf{b}_i + \Delta \mathbf{b}_i \quad (\Delta \mathbf{b}_i = \gamma(\mathbf{E}_i - \lambda \mathbf{b}_i)) \quad \mathbf{E}_i : \text{アイテム} i \text{の平均error}$$

課題2

■ Github URL

https://github.com/zakktakk/Movie_Recommendation

■ Code

predictor.py : データフレームの作成, 予測の実行

MatrixFactorization.py : Matrix Factorizationを実装

RBM.py : RBMを実装

SlopeOne.py : 3種類のSlope Oneを実装

evaluate.py : RMSEによる予測結果の評価

課題3

■ 実装した手法の改善点を挙げ、説明してください

1. 情報が得られていない新規のユーザー，アイテムに対するレコメンドが難しい

この問題は一般に**コールドスタート問題**と呼ばれている．今回実装したMFはいわゆる**協調フィルタリング**と呼ばれるアプローチを取っている．協調フィルタリングでは類似度の高いユーザ同士は嗜好する映画も類似するという仮定の元でレコメンドを行う．

そのため，ユーザ同士，アイテム同士の相対的な関係を求める必要がある．しかし，全く情報のない新規のユーザー，アイテムに対してはその相対的な関係が定義できないため，協調フィルタリングによるレコメンドが難しい．

この対処法として次の改善点でも挙げるように映画，ユーザの属性情報を元にしたレコメンドが挙げられる．

課題3

■ 実装した手法の改善点を挙げ、説明してください

2. 映画のコンテンツ情報，ユーザーの属性情報を用いていない

MovieLensのデータセットでは映画のコンテンツ情報として，映画のタイトル，映画のジャンル，公開年が与えられている．また，ユーザの属性情報として年齢，性別，職種，zipコードが与えられている．

これらの情報を用いたレコメンドのことをコンテンツベースフィルタリングと呼ぶ．コンテンツベースの強みは1つ目に挙げたコールドスタート問題に強いという部分である．協調フィルタリングは他のユーザ，アイテムとの相対的な関係によりレコメンドを行っているが，コンテンツベースのレコメンド結果はユーザ，アイテムの絶対的な属性に依存するためコールドスタート問題は起こりにくい．そこで，行動情報が少ない新規ユーザにはコンテンツベースのレコメンドを行い，熟練ユーザには協調フィルタリングによるレコメンドを行うなどの役割分担が考えられる．

具体的な方法として同じ属性情報を持つ

ユーザの評価の平均値を新規ユーザーの初期予測評価値とするなどが挙げられる．

課題3

■ 実装した手法の改善点を挙げ、説明してください

3. ユーザーの評価の時間変化を考慮していない

ユーザーの嗜好は時間とともに変化する。例えば「SF好きの少年が成長するにつれて恋愛映画を好きになる」や「見る映画が多くなるにつれ、評価が厳しくなる」などが挙げられる。

そのため、近年の傾向をより重視したアルゴリズムが求められる。実際MFに時系列性を追加したtime-SVDという手法を2009年にKorenが発表した。time-SVDは評価のバイアス項を時間の関数とすることで既存のSVDよりも高い予測精度を得た。