VASILY　サマーインターン　課題

木上智貴（KINOUE TOMOKI）

課題 1

レコメンドシステムを実現する際に有用と思われるアルゴリズムや手段をひとつ挙げ、説明してください

アルゴリズムとしてはユーザーベースに基づく協調フィルタリングを用いた。これは、評価（好み）が似ているユーザー同士では、同じ映画について同様の評価を下す傾向にあると考えるアルゴリズムである。
`u.data` に用意されているユーザーの映画に対する評価から、２ユーザー間の類似度をスコアとして計算し、類似度の高いL人のユーザを選び、L人のユーザーの評価から重み付けを行い推薦を行う。
今回の実装では、類似性はコサイン類似度を用いた。また、このアルゴリズムでは、評価を行うユーザー数が多いほど類似度が高くなってしまう（沢山の人に見られた映画ほど評価が高くなる）ため、類似度スコアの合計値で割ることにより正規化を行っている。
スコアの高い上位K本の映画を推薦するシステムを作成した。

実装はPythonで行った。提供されているデータを `pandas` で読み込み、 `numpy` を用いて前処理・演算を行っている。
合計５個の関数から構成されており、それぞれ以下の役割を果たしている。

- `make_user_item_pairs()`
  提供されているデータから、user-itemの２次元配列を作成する関数

- `search_movie_title(id_list,score_list)`
  `movie_id` に対応する `movie_title` と、類似度スコアを降順リストに変換して返す。引数は推薦された映画のidのリストと、類似度スコアのリスト

- `cos_similarity(x,y)`
  ２つのベクトルを引数にとり、コサイン類似度を計算する関数。

- `user_user_similarity(pairs)`
  ユーザー同士の類似度を２次元配列で返す関数。引数は `make_user_item_pairs()` で作成されたペアの２次元配列。

- `movie_recommend(users_sim, user_item_pairs)`
  `user_id` とユーザー間類似度と、user/itemの対応リストを与えた時に、 `user_id` のユーザーに映画をリコメンドする。正規化した類似度スコアの内、上位K件の `item_id` とその類似度スコアを返す。

出力サンプル

```
~/Downloads/vasily $ python recommendation.py
user_idを入力してください : 100
推薦する映画の本数を入力してください : 25
対象にする類似ユーザー数を入力してください : 30
```

```
[['Full Monty, The (1997)', 5.0],
 ['Remains of the Day, The (1993)', 5.0],
 ['Dead Man Walking (1995)', 5.0],
 ['Fugitive, The (1993)', 5.0],
 ['Glimmer Man, The (1996)', 5.0],
 ["It's a Wonderful Life (1946)", 5.0],
 ['Underneath, The (1995)', 5.0],
 ['Chairman of the Board (1998)', 5.0],
 ['Fargo (1996)', 5.0],
 ['Interview with the Vampire (1994)', 5.0],
 ['3 Ninjas: High Noon At Mega Mountain (1998)', 5.0],
 ['Conspiracy Theory (1997)', 5.0],
 ['They Made Me a Criminal (1939)', 5.0],
 ['Jingle All the Way (1996)', 4.4881664884032633],
 ['Rosewood (1997)', 4.456390528987213],
 ['Passion Fish (1992)', 4.035636884634112],
```

精度の評価

精度の評価は平均二乗誤差で検討を行った。 `u.data` を８：２の比率で分割した訓練データとテストデータが用意されており、訓練データで算出した評価値とテストデータとの平均二乗誤差を計算した。 計算値は以下の通りである。

- u1：1.6928205358853987
- u2：1.6784707273912878
- u3：1.6787418898832105
- u4：1.663198426754202

課題３

2.で実装した手法の改善点を挙げ、説明してください

今回の実装は既に `u.data` や `u.item` が与えられており、それらを用いて計算を行っているが、実際のシステムでは新規ユーザーや新作映画に対応する必要がある。
また、ユーザー間の類似度を計算するためにはある程度のレビューが無いといけないため、新規ユーザーのIDを登録しても、類似度の高い他のユーザーが存在しないという事態に陥ってしまう。

改善点の１つとしては、ユーザーベースだけでなくアイテムベースの類似度を考えることが挙げられる。これは、似たような映画は同様の評価を受けるだろうと考えるアルゴリズムであり、新規のユーザーに対しても効力を持つが、新作映画では類似度の高い映画の数が少ないという問題点もある。

次の改善点として、実行に少し時間を要してしまうことが挙げられる。ユーザー同士の類似度は組み合わせの数が多く、計算に時間がかかってしまっている。そのため、アルゴリズムの最適化やより効率的なオーダーでの計算を考える必要がある。

その他の改善点としては、今回の実装では評価を行っていない時に０という値を用いているが、これは「評価が最低である」と捉えることもできてしまう。そのため、実際に評価された作品のみで計算を行うためにMatrix Factorizationというアルゴリズムが提案されており、実際に賞を勝ち取った歴史も存在している。