協調フィルタリングを用いた 映画評価予測プログラムの試作 -スロープワンアルゴリズムによる協調フィルタリング-

Trial Program for Movie Rating Prediction using Collaborative Filtering -Collaborative Filtering by the Slope-One Algorithm-

谷 研究室 森口 麻里
Mari Moriguchi

概要

映画の評価情報などの巨大データから、効率良く、より良い推薦を提供するような技法が期待されている。本研究では協調フィルタリングを用いて、より最適な推薦を得られるよう比較実験をする。私はその中でも Slope One という手法に注目し、それを基に提案を行った。

1 はじめに

今日、ネットワークの発展に伴い、手軽に多くの情報が収集できるようになった。一方、収集した情報からユーザの必要とする情報を獲得するには、多くの時間を要するようになった。このような大量の情報の中から必要な情報を抽出する例として、レコメンデーションが挙げられる。レコメンデーションとは、ユーザの好みを分析し、各ユーザごとに興味のありそうな情報を選択して表示するサービスである。例えば、Amazon で扱われている「この本を買った人はこんな本も買っています」はレコメンデーションの有名な実例である。

今回、我々は米オンラインビデオレンタル最大手の会社である NetFlix が主催している NetFlix Prize[2] に参加することを目標に、レコメンデーションシステムの実装を行った。コンテストの内容は、大量の映画評価データをもとに、ユーザーがある映画にどれだけの評価をするのかを予測するというものである。その予測にはレコメンデーションの手法の一つである協調フィルタリングが用いられる。

協調フィルタリングの中にも様々な手法があるため、その手法の中から有効なものを探そうとしたが、NetFlix のデータセットは巨大すぎて扱いづらい。よって、本研究では NetFlix よりも規模の小さいデータを提供している MovieLens[3] という映画評価情報を集めたデータセットを扱い、そのデータをもとにプログラムの実装、実験・比較を行うことで、より有効な手法を探すこととした。また、その手法をもとに提案をし、そのプログラムの実装・実験も行った。

以下、第 2 章で協調フィルタリングの紹介を行い、第 3 章ではその手法の一つである Slope One について記し、第 4 章で提案を述べる。そして第 5 章で実験、第 6 章で結果・考察を述べ、まとめとする。

2 協調フィルタリングについて

協調フィルタリングとは、たくさんのユーザーの嗜好情報を元に、あるユーザーに対し、類似した嗜好情報を持つ別のユーザーの情報を使って自動的に推論を行う方法論であり、1992 年に DavidGoldberg らの論文 [4] で登場した。ここで、推薦対象のユーザーと趣味嗜好の似ているユーザーが好むアイテムを、推薦対象のユーザーも好むであろう、ということが重要な考えとしてあげられる。

協調フィルタリングには様々な手法がある。その中で最も有名なものは、ユーザーベースと呼ばれる手法である。この手法は、ユーザー間の類似度を測定し、その数値を元にアイテムを推薦するもので、Paul Resnick らの論文 [5] で発表された。

次に、アイテムベースという手法がある。この手法では、ユーザーが評価したアイテムを元に類似度を測定し、推薦するもので、Badrul Sarwar らの論文 [6] で発表された。ユーザーベースに比べ、データセットが小さくても高いクオリティの予測が行えるといわれている。

Slope One は、このアイテムベースの派生として存在するもので、2005 年に Daniel Lemire らによって発表された [1]。手法としては、ユーザがつけたアイテム間の評価の平均偏差からアイテムの評価値を予測するもので、単純で比較的性能の高いアルゴリズムとして広まった。

協調フィルタリングの問題点としては、次の点が挙

げられる。1 つはデータセットの密度が疎の場合精度が悪く、またデータセットが大きくなるほど計算が困難になる点である。もう 1 つは、過去のデータのないユーザーに対しては推薦を行うことができないというもので、これはコールドスタート問題と呼ばれている。

3 Slope One

3.1 Slope One

Slope One とは、個々のユーザのアイテム評価値間差分を算出し、全ユーザについて算出されたアイテム評価値間差分の平均値を算出し、この算出結果を元に結果を算出するものである。
計算方法としては、まず以下の公式を用いてアイテム間の平均偏差を求める。

$$dev_{j,i} = \sum_{u \in U_{j,i}} \frac{R_{u,j} - R_{u,i}}{|U_{j,i}|}$$

ここで、u はユーザー、i, j はアイテム、$R_{u,i}$ はユーザー u がアイテム i を評価した値、$U_{j,i}$ はアイテム i, j を両方評価したユーザーの集合、$|U_{j,i}|$ はその集合の要素数を示す。
次に、以下の公式を用いて結果を算出する。

$$P(R_{u,j}) = \frac{1}{R_u} \sum_{i \in R_u} (dev_{j,i} + u_i)$$

ここで、R_u はユーザー u が評価したアイテムの集合、u_i はユーザー u がアイテム i を評価した値を示す。

3.2 Weighted Slope One

平均偏差について、1 人の評価を元に出した値と、1000 人の評価を元に出した値では、より多くの人が評価した方が信頼性が高いと予想される。
よって Weighted Slope One では、結果を出す際に評価人数で重みをかけている。

$$P^w(R_{u,j}) = \frac{\sum_{i \in R_u} (dev_{j,i} + u_i)|U_{j,i}|}{\sum_{i \in R_u} |U_{j,i}|}$$

3.3 bi-polar Slope One

予測対象ユーザーが好きなアイテムを基に結果を算出する場合、そのアイテムを同じく好きなユーザーのデータを基に推薦すべきであると思われる。そこで bi-polar Slope One では好き (like)・嫌い (dislike) に分けて評価を算出する。

ここでは、基準より評価の高いものは好き、低いものは嫌いと判断する。基準には 3 を用いる方法と、各ユーザーの評価の平均を用いる方法が挙げられるが、今回は各ユーザーの評価の平均を用いる。なぜならば、ユーザーによっては評価が高い方に固まっていることがあるからである。

そこで、平均偏差の求め方を以下のように変更する。

$$dev^{like}_{j,i} = \sum_{u \in U^{like}_{j,i}} \frac{R_{u,j} - R_{u,i}}{|U^{like}_{j,i}|}$$

dislike に関する式も同様である。
それとともに、結果の算出方法を以下のように変更する。

$$P^{bp}(R_{u,j}) = \frac{\sum_{i \in R^{like}_u} dev^{like}_{j,i}|U_{j,i}| + \sum_{i \in R^{dislike}_u} dev^{dislike}_{j,i}|U_{j,i}|}{\sum_{i \in R^{like}_u} |U^{like}_{j,i}| + \sum_{i \in R^{dislike}_u} |U^{dislike}_{j,i}|}$$

この式によって、ユーザーの嗜好の合ったユーザーのデータのみを用いて判定することが出来るようになる。

4 Slope One をもとにした提案

Weighted Slope One をもとに
誰にも評価されていないアイテムがあった場合、その重みは 0 になってしまう。その場合、予測される評価も 0 になってしまうため、最低の重みを 1 に変更してみた。これで、誰にも評価されていないアイテムを評価するときには、ユーザーの平均を用いることが出来るようになる。

5 実験

5.1 実験環境

CPU	Athlon64 3200+
メモリ	1024MB
OS	CentOS 5.2

5.2 データセット

アルゴリズムの性能評価に MovieLens のデータセットを利用する。
MovieLens のデータセットはユーザー数:943、映画数:1682、評価数:10 万で構成されていて、各ユーザーは評価数が 20 以上であることが保証されている。今回はこのファイルの中から各ユーザー毎に 10 個ずつ評価

をランダムに抜き出し、予測対象データと元データに分ける。
ここで、コールドスタート問題を避けるため、各ユーザーの最低評価数 10 は保証することとする。
そのファイルを 10 組作り出した。そしてその 10 組のファイルを用いて実験を行い、精度を求め、その平均を比較する。
この手法は分割学習法と呼ばれる。

5.3 方法

精度を比較するための指標として MAE を用いる。

$$MAE = \frac{1}{N}\sum_{i=1}^{N} | \text{予測値} - \text{実際の値} |$$

ここで、N は予測対象データの総数である (=9430)。MAE の値が 0 に近づけば近づくほど誤差が少ない、つまり精度が良いということになる。

6 実験結果、考察

6.1 実験結果

以下に実験の結果を紹介する。

既存アルゴリズムの実験結果

	精度 (MAE)
Slope One	0.7630811
Weighted Slope One (重み 0)	0.7626921
bi-polar Slope One	0.7683743

提案アルゴリズムの実験結果

	精度 (MAE)
Weighted Slope One (重み 1)	0.7617637

また、比較のために、同研究を行った田中・吉野の実験結果を以下に記す。

田中が実装した既存アルゴリズムの実験結果

	精度 (MAE)
ユーザーベース	1.768743
アイテムベース	1.276004

田中・吉野が提案したアルゴリズムの実験結果

	精度 (MAE)
田中	0.822261
吉野	0.860496

6.2 考察

以上のことから、アイテムベース・ユーザーベース・Slope One を比較すると、Slope One の精度が一番高いことがわかる。また、Slope One に重みをかけることも有効であることが確認された。
全体を通してみると、Weighted Slope One の重みを 1 に変更したものが、一番精度がいいようである。
今回、予想では bi-polar Slope One が一番精度が高くなると思われたが、実験結果はそうならなかった。bi-polar Slope One はデータ量が多いほど精度が高くなるとされているので、データセットの量の少なさが原因で精度が下がってしまったと思われる。

7 今後の課題

NetFlix のデータセットと比較すると、MovieLens のデータセットはとても小さい。したがって、大規模データに対応できるような手法を取り入れるか、またはデータベースの利用の検討や、マシンスペックの改善をしたほうがいいと思われる。

8 参考文献

[**1**] Daniel Lemire, Anna Maclachlan: Slope One Predictors for Online Rating-Based Collaborative Filtering(2005)

[**2**] NetFlix Prize http://www.netflixprize.com//index

[**3**] MovieLens Data Sets http://www.grouplens.org/node/73

[**4**] David Goldberg, David Nichols, Brian M. Oki, Douglas Terry: Using collaborative filtering to weave an information tapestry(1992)

[**5**] Paul Resnick, Neophytos Iacovou, Mitesh Suchak,Peter Bergstrom,John Riedl : GroupLens: An Open Architecture for Collaborative Filtering of Netnews(1994)

[**6**] Badrul Sarwar, George Karypis, Joseph Konstan, John Riedl: Item-Based Collaborative Filtering Recommendation Algorithms(2001)