インターフェイスの街角 (93) – 本棚演算

増井俊之

協調フィルタリング

最近、自分と似た嗜好をもつ人の行動をもとに情報を取捨選択する〝協調フィルタリング (Collaborative Filtering) システム〟がよく利用されています。これは、自分の嗜好に関するデータをネットワーク上で他人と共有することで自分と趣味が近い人を捜し、その人たちの情報選択状況を勘案してフィルタリングする手法です。

趣味を同じくする友人の本棚には自分にとって興味深い本が並んでいそうですし、同じ傾向の音楽をよく聴いている人が好きな曲は自分も気に入る可能性が高いでしょう。また、自分と意見が近い新聞の記事や似た趣味をもつ人の blog を読むのはごく普通の行為なので、協調フィルタリングの原理は直感的で、計算機上でこのような手法を利用すれば有用な場面が多いと思われます。

GroupLens

協調フィルタリングの先駆的なシステムとしては、NetNews の記事を自動的ににランクづけする GroupLens システム [1] が有名です。

このシステムでは、ユーザーは最初にいくつかの記事に対する評価値を入力します。この結果をほかのユーザーの評価と比較することにより、そのユーザーと自分の類似度を計算します。自分がまだ読んでいない記事の評価を知りたい場合には、他人の評価を類似度で重みづけして足しあわせ、その記事を自分がどう評価するを予測します。

GroupLens のアルゴリズム

たとえば、6 件のニュース記事について、4 人のユーザーが図 1 のように 5 段階の評価点をつけたとしましょう。2 番の記事をみると、Ken は 5 点をつけていますが、Lee は 2 点しかつけていません。全員がすべての記事を読むわけではないので、点がついていないところもあります。

図 1　記事番号と評価値

記事番号	Ken	Lee	Meg	Nan
1	1	4	2	2
2	5	2	4	4
3			3	
4	2	5		5
5	4	1		1
6	?	2	5	?

ここで、評価値ベクトルの相関係数をユーザーの類似度として利用します。相関係数は、2 つのベクトルの共分散を各ベクトルの標準偏差の積で割れば計算できます。たとえば、Ken と Lee はともに記事 1、2、4、5 に対して評価をおこなっており、Ken の評価値ベクトルは (1, 5, 2, 4)、Lee の評価値ベクトルは (4, 2, 5, 1) となっていますが、これらの相関係数 r_{KL} は以下のようにして計算できます。ここで、σ_K, σ_L は標準偏差、$\overline{K}, \overline{L}$ は評価値の平均、$Cov(K,L)$ は共分散で、Ken の評価値の平均は $\overline{K}=3$、Lee の評価値の平均は $\overline{L}=3$ です。

$$
\begin{aligned}
r_{KL} &= \frac{Cov(K,L)}{\sigma_K \sigma_L} \\
&= \frac{\sum_i (K_i-\overline{K})(L_i-\overline{L})}{\sqrt{\sum_i (K_i-\overline{K})^2}\sqrt{\sum_i (L_i-\overline{L})^2}} \\
&= \frac{-2-2-2-2}{\sqrt{10}\sqrt{10}} \\
&= -0.8
\end{aligned}
$$

初校 [2005 年 10 月 30 日]

Ken と Lee の評価には、大きな負の相関があることが分かります。一方、Ken と Meg の類似度は 1、Ken と Nan との類似度は 0 になります。

Ken が 6 番の記事をどう評価するかを予測したいときは、Ken 以外のユーザーによる評価値に類似度の重みをつけて足しあわせて計算します。Ken と類似度が高い Meg が 5 点をつけているため、計算結果は以下のように 4.56 という大きな値になっています。

$$
\begin{aligned}
K_6\text{予測値} &= \overline{K} + \frac{\sum_{J\in\ \text{評価者}}(J_6 - \overline{J})r_{KJ}}{\sum_J |r_{KJ}|} \\
&= 3 + \frac{2r_{KM} - r_{KL}}{|r_{KM}| + |r_{KL}|} \\
&= 4.56
\end{aligned}
$$

協調フィルタリングは情報検索手法の一種ですが、その方式はかなり異色です。一般的な検索システムでは、データの内容やデータ間のリンクのような明示的な情報にもとづいて検索するのが普通ですが、GroupLens などの協調フィルタリング・システムではデータの内容はまったく考慮せず、他人の評価だけを利用します。このため、内容にもとづく直接的な検索はできませんが、どのようなデータに対しても同じ手法が適用できますし、本当に自分が求める情報を高い精度で得られる可能性があります。

お手軽な協調フィルタリング

GroupLens のような方式の協調フィルタリングが有効に働くには、ユーザーの嗜好を示す大量のデータが必要です。自分の嗜好情報をたくさん入力しても、比較するユーザーがいなければ意味がありません。たとえユーザーの数が多くても、嗜好の比較が可能な人がいなければうまくいかないでしょう。また、最初に多くの対象に点数をつける作業は面倒ですし、新しいデータが出てくるたびにいちいち評価する作業は（すくなくとも私には）長続きするとは思えません。そもそも、自分があまり好きではない対象を評価し、わざわざ悪い点をつけるような人は少ないでしょうから、低い評価と無評価の区別がつかない可能性もあります。

このような理由のためか、得点づけが必要な協調フィルタリング・システムはそれほど普及していないようです。

一方、明示的に採点処理をおこなう必要のない協調フィルタリング手法もあります。たとえば、Amazon.com で本を購入すると、〝この本を買った人はこういう本も買っています〟というお薦めの本が表示されることがあります。推薦される本は、ユーザーの購入履歴や本のジャンルにもとづいて選ばれているのだと思いますが、ユーザーは本の購入以外に特別な作業をしていないのに、それなりに適切な本がリストアップされるようです。

GroupLens などの明示的な評価作業が必要な協調フィルタリング・システムよりも、Amazon.com のように暗黙的に評価がおこなわれる方式のほうが、普及する可能性が高いのではないでしょうか。Amazon.com には膨大な情報が蓄積されているため、本を購入したという単純な情報をもとに、それなりに有用なフィルタリングがおこなえるのでしょう。

インターネットには、このような用途に使えそうな膨大なデータが蓄えられています。たとえば、Mixi や GREE などの SNS (Social Networking System) には、参加者の人間関係や趣味に関する膨大な情報があります。あるいは、del.icio.us のようなソーシャル・ブックマークのシステムには、参加者の興味の対象がタグや URL のかたちで蓄積されています。これらの情報は、コミュニケーションのために集められたものであり、フィルタリングに使う目的で登録している人は少ないでしょう。しかし、情報共有やフィルタリングにも役立つと分かれば、より適切なリンクを張ったり、適当なタグづけをおこなうようになるかもしれません。

本棚演算システム

今回は、1 月号で紹介した「本棚.org」[1]の書籍情報を活用し、協調フィルタリングなどの計算をおこなうシステムを試作してみました。

本棚.org ではユーザーが自由に〝本棚〟を作り、書籍を登録できるようになっています。棚ごとに書籍の集合が対応している点は Amazon.com と似ていますが、それぞれの棚は 1 人の人間に対応しているわけではなく、気に入った本が登録されているともかぎりません。しかし、それぞ

1 http://hondana.org/

図 2　本棚.org のデータ構造

本の番号	1	2	3	4	5	6	7
増井の本棚		1	1			1	1
萌え専科の本棚	1				1		1
ベストセラーの本棚			1	1	1		
ハッカーの本棚		1				1	
……							

れの棚には特徴があるので、各種の演算を適用すればいろいろな用途に使えそうです。

たとえば、私は〝増井の本棚〟[2]という本棚を作り、持っている本や購入予定の本を登録しています。この場合、同じような本が登録されている本棚を調べれば、関連のある本をみつけられる可能性があります。

また、〝萌え専科〟という本棚[3]があり、ここではいろいろなユーザーが自由に〝萌え〟関連の本を登録しています。これらのデータを利用すると、ある本が〝萌え的〟か否かを判断したり、〝萌え的〟な本が登録されている別の本棚を検索し、別の本をみつけられるかもしれません。

このようにして簡単な協調フィルタリングがおこなえますが、各本棚データのあいだで演算を実行すれば、性格の異なるおもしろいデータが抽出できるはずです。たとえば、10 以上の本棚に登録されている本を計算すれば、ポピュラーな本のリストが得られます。ある本棚のデータからポピュラーな本のリストを引き算すれば、その本棚の特徴を表すリストが作れるでしょう。

このような計算は無限に考えられるので、本棚やユーザーに関する基本的な演算子を用意し、これを組み合わせて協調フィルタリングをおこなったり、特定の目的に合致する本のリストを作ったりすることもできます。このような〝本棚演算〟の実験をおこなってみました。

本棚演算のデータ構造

本棚データは、図 2 のような大きな行列で表現できます。2005 年 10 月現在、約 3,000 の本棚に約 10 万点の本が登録されていますが、登録書籍総数は 20 万冊程度ですからきわめて疎な行列になっています。

このようなデータに対し、以下の演算を定義します。

- 本棚に含まれる本のリストを作成
- 本のリストの加算/減算
- 本のリストのソート
- 本のリストにある本を含む本棚のリストを作成
- 本のリストにマッチする本棚のリストを作成

さらに、行と列を反転させた以下のような演算も定義します。

- ある本を含む本棚のリストを作成
- 本棚のリストの加算/減算
- 本棚のリストのソート
- 本棚のリストにある本棚に含まれる本のリストを作成
- 本棚のリストにマッチする本のリストを作成

本と本棚のリストをそれぞれ Ruby のクラスとして表現し、これらの演算をメソッドとして用意しておけば、各種の演算を簡単に試すことができます。

本棚演算の実例

これらの演算を組み合わせて、実際に意味のある情報が得られるかを試してみましょう。

私の興味を惹きそうな本の抽出

まず、私に推薦する本の計算方法を考えてみます。

まず、〝増井の本棚〟に登録されている本のリストを取得します。本棚名を指定して BookList クラスのインスタンスを生成すれば、リストが得られます。

```
#  増井の本棚の本リストを取得
masuibooks = BookList.new('増井')
masuibooks.dump
```

実行結果は以下のようになります。

```
1 448866301X 星を継ぐもの（ジェイムズ・P・ホーガン，
 池 央耿）
1 0262232340 Metacreations: Art and Artificial
  Life (Mitchell Whitelaw)
1 4257722118 栞と紙魚子（1）（諸星 大二郎）
1 1558605339 Readings in Information
  Visualization: Using Vision to Think (Morgan
  Kaufmann Series in Interactive Technologies)
  (Stuart K. Card, Jock D. MacKinlay, Ben
  Shneiderman)
1 4093661111 入門講座 2万5000分の1地図の読み方（平塚
  晶人）
1 410112115X 砂の女（安部 公房）
1 4062574314 新・脳の探検（上）（フロイド・E・ブルーム，
  久保田 競，中村 克樹）
  ……
```

2 http://www.hondana.org/C1FDB0E6/
3 http://www.hondana.org/CBA8A4A8C0ECB2CA/

このリストに対して、これと同じような本が登録された本棚のリストを計算する演算子 similar を適用すると、〝増井の本棚〟の内容に近い本棚のリストが得られます。

```
#  増井の本棚の本のリストを取得
masuibooks = BookList.new('増井')

#  上記のリストに近い本をもつ本棚のリストを取得
similarshelves = masuibooks.similar
similarshelves.dump
```

以下の実行結果は、〝dai5d の本棚〟が〝増井の本棚〟にもっとも近いことを示しています。

```
1 dai5d
1 sakai
1 matznaga
1 suchi
1 m-use
1 増井1
1 svslab
1 ystt
  ……
```

このリストから上位 40 個を取得し、それらに登録されている本を調べて 2 つ以上の本棚に登録されている本のリストを作ると、協調フィルタリング的な観点からみた〝私の興味を惹きそうな本〟のリストが作成できます。

```
#  増井の本棚の本のリストを取得
masuibooks = BookList.new('増井')

#  上記のリストに近い本をもつ本棚のリストを取得
sshelves = masuibooks.similar

# 内容が近い本棚を40取得し、そのなかの2つ以上に
# 登録されている本のリストを作成
sbooks = sshelves[0...40].booklist.sort.major(2)
sbooks.dump
```

実行結果は以下のようになります。

```
17 4839912653 Code Reading―オープンソースから学ぶプログラミングテクニック（トップスタジオ，まつもとゆきひろ，平林 俊一，鵜飼 文敏）
17 4844317210 Rubyソースコード完全解説（青木 峰郎，まつもと ゆきひろ）
14 4314005564 利己的な遺伝子（リチャード・ドーキンス，日高 敏隆，岸 由二，羽田 節子，垂水 雄二）
14 4797318325 Wiki Way―コラボレーションツールWiki（ボウ ルーフ，ウォード カニンガム，Bo Leuf，Ward Cunningham，yomoyomo）
14 4756136494 プログラミング作法（ブライアン カーニハン，ロブ パイク，Brian Kernighan，Rob Pike，福崎 俊博）
14 489471163X 計算機プログラムの構造と解釈（ジェラルド・ジェイ・サスマン，ジュリー・サスマン，ハロルド・エイブルソン，Gerald Jay Sussman，Julie Sussman，Harold Abelson，和田 英一）
13 4798102040 コモンズ（ローレンス・レッシグ，山形 浩生）
13 4320026926 プログラミング言語C ANSI規格準拠（B. W.カーニハン，D.M. リッチー，石田 晴久）
13 478850362X 誰のためのデザイン?―認知科学者のデザイン原論（野島 久雄，D.A. ノーマン）
  ……
```

たしかに、このリストには私が読みたい本も含まれています。しかし、大部分はすでに持っていますし、計算機関連の本がやたらと多いのも気になります。このような本をリストから除けば、推薦する本のリストが作れそうです。

さいわい、このようなフィルタリングに使えそうな、計算機関連の本だけが登録されている〝正統派ハッカーの本棚〟があります。ここにある本と同じものがリストアップされている本棚の一覧を作成し、そこに含まれる本のリストを計算すれば、計算機関連の本のリストが作れるでしょう。この分野の本の登録が多そうな本棚を 100 選び、2 つ以上の本棚に登録されているものを〝計算機関連書籍〟と解釈してみます（誌面の都合上、⇒ で折り返しています。以下同様）。

```
#  正統派ハッカー本棚の本のリストを取得
hackbooks = BookList.new('正統派ハッカー')

#  上記のリストに近い本をもつ本棚のリストを取得
sshelves = hackbooks.similar

# 計算機関連と思われる本のリストを作成
compbooks = sshelves[0...100].booklist.sort.⇒
major(2)
compbooks.dump
```

実行結果は以下のようになりました。

```
46 4320026926 プログラミング言語C ANSI規格準拠（B. W. カーニハン，D.M. リッチー，石田 晴久）
42 4756136494 プログラミング作法（ブライアン カーニハン，ロブ パイク，Brian Kernighan，Rob Pike，福崎 俊博）
36 4839912653 Code Reading―オープンソースから学ぶプログラミングテクニック（トップスタジオ，まつもとゆきひろ，平林 俊一，鵜飼 文敏）
34 4844317210 Rubyソースコード完全解説（青木 峰郎，まつもと ゆきひろ）
25 4894712369 珠玉のプログラミング―本質を見抜いたアルゴリズムとデータ構造（ジョン ベントリー，Jon Bentley，小林 健一郎）
25 4274065979 ハッカーと画家 コンピュータ時代の創造者たち（Paul Graham，川合 史朗）
24 4756118089 Effective C++（Scott Meyers，吉川 邦夫）
24 475611895X プログラミング言語C++第3版（Bjarne Stroustrup，長尾 高弘）
23 4894714531 プログラミングRuby―達人プログラマーガイド（デビット トーマス，アンドリュー ハント，David Thomas，Andrew Hunt，田和 勝，まつもと ゆきひろ）
```

……

計算機関連の本をうまくリストアップできたようです。なお、『プログラミング言語 C』は 100 の本棚のうち 46 個に登録されていました。

これらの結果を用いて減算を実行すれば、計算機関連の本を除く推薦本リストが得られるはずです。最終的なスクリプトは以下のようになります。

```
#  増井の本棚の本のリストを取得
masuibooks = BookList.new('増井')

#  上記のリストに近い本をもつ本棚のリストを取得
sshelves = masuibooks.similar

# 内容が近い本棚を40取得し、そのうちの2つ以上に
# 登録されている本のリストを作成
sbooks = sshelves[0..40].booklist.sort.major(2)

#  正統派ハッカー本棚 の本のリストを取得
hackbooks = BookList.new('正統派ハッカー')

#  上記のリストに近い本をもつ本棚のリストを取得
sshelves = hackbooks.similar

# 計算機関連と思われる本のリストを作成
compbooks = sshelves[0..100].booklist.sort.⇒
major(2)

# 増井の本棚にもともと含まれていた本と
# 計算機関連の本を除く
result = sbooks.remove(masuibooks).remove⇒
(compbooks)
result.dump
```

実行結果は、以下のようになりました。

```
10 406313248X 攻殻機動隊（1）（士郎 正宗）
10 4140807431 新ネットワーク思考―世界のしくみを読み
  解く（アルバート・ラズロ・バラバシ，青木 薫）
9 4756133126 ロボットにつけるクスリ―誤解だらけのコン
  ピュータサイエンス（星野 力）
9 4167330083 ぼくはこんな本を読んできた―立花式読書論、
  読書術、書斎論（立花 隆）
7 4063211444 げんしけん（1）（木尾 士目）
7 4061495755 動物化するポストモダン―オタクから見た日
  本社会（東 浩紀）
7 410401303X 博士の愛した数式（小川 洋子）
7 415011451X しあわせの理由（グレッグ・イーガン，Greg
  Egan，山岸 真）
7 4480877533 あなたの話はなぜ「通じない」のか（山田 ズ
  ーニー）
6 4088727177 ヒカルの碁(1)（ほった ゆみ，小畑 健，梅
  沢 由香里）
6 4167330032 精神と物質―分子生物学はどこまで生命の謎
  を解けるか（立花 隆，利根川 進）
6 4063198456 スピリット オブ ワンダー（鶴田 謙二）
6 4150102074 月は無慈悲な夜の女王（ロバート A.ハイン
  ライン，矢野 徹）
6 4101001340 世界の終りとハードボイルド・ワンダーランド
  〈上〉（村上 春樹）
6 4480084398 唯脳論（養老 孟司）
6 4150113378 祈りの海(グレッグ・イーガ
ン，Greg Eagan，
  山岸 真）
6 4121010876 ゾウの時間 ネズミの時間―サイズの生物学
  （本川 達雄）
  ……
```

図 3　萌え専科の本棚

たしかに、私の読書欲をそそる本のリストになっています（事実、このリストに挙がっているものの何点かは、すでに持っていて、本棚.org には登録していないものでした）。

˝萌え本˝の検索

今度は、同様の計算を˝萌え専科の本棚˝に適用してみましょう。

˝萌え専科の本棚˝には図 3 のような本が登録されていますが、ここに含まれていない˝萌え本˝はまだまだたくさんあると思われます。そこで、以下のスクリプトを用いて、私への推薦本の計算と同様の方法で未登録の˝萌え本˝を計算してみましょう。

```
# 萌え本のリストを取得
moebooks = BookList.new('萌え専科')
#
sbooks = moebooks.similar[0..40].booklist.⇒
major(2).remove(moebooks)
sbooks.dump
```

この計算の結果、図 4 のような本のリストが得られました。どうやら、未登録の˝萌え本˝が検索できたようです。

本にもとづく計算

これまでの例では、1 つの本棚をもとに関連本を取得していましたが、1 冊の本をもとに計算してみましょう。例

図 4 moesimilar

として、鯨 統一郎の『邪馬台国はどこですか』という本を含む本棚をリストアップし、それらの本棚に含まれる本のリストを以下のスクリプトで計算してみます。

```
# 「邪馬台国はどこですか」を含む本棚リストを作成
yamataishelves = ShelfList.new('4488422012')

# 似た本棚に含まれる本のリスト
yamataishelves.similar[0..10].sort.dump
```

実行結果は以下のようになりました。

```
1 4043730012 あすなろの詩（鯨 統一郎）
1 4043432011 覆面作家は二人いる（北村 薫）
1 404343202X 覆面作家の愛の歌（北村 薫）
1 4122034140 謎物語―あるいは物語の謎（北村 薫）
1 4061854348 どんなに上手に隠れても（岡嶋 二人）
1 4061822365 タイムスリップ森鴎外（鯨 統一郎）
1 4334074308 九つの殺人メルヘン（鯨 統一郎）
1 4488416012 生ける屍の死（山口 雅也）
1 4041665094 旌旗流転―アルスラーン戦記 9（田中 芳樹）
1 4101438129 光射す海（鈴木 光司）
……
```

これで、『邪馬台国はどこですか』と系統の似た本のリストが得られました。

演算結果の評価

本棚データに対して簡単な演算を定義し、これをうまく組み合わせれば、有用な情報が計算できることが分かりました。普通の協調フィルタリング・システムでは、計算機関連の本を除くといった処理をユーザーが指定することはできませんが、ユーザーが自分で演算を組み合わせてプログラムを作れる仕組みがあれば、単純な協調フィルタリング・システムとは異なる、おもしろい計算が可能でしょう。

おわりに

大量のデータから意味のある情報を抽出する協調フィルタリングなどの手法は、以前から多くの人工知能研究者によってさかんに研究されています。近年は、扱えるデータの種類や量が膨大になったため、今後、実用的なシステムも増えてくると思われます。

現在、Google や Yahoo!などによる検索はひろく利用されていますが、これらのシステムでは明示的に指定された情報しか検索できません。一方、SNS やソーシャル・ブックマーク、メーリングリスト、本棚.org のように、いろいろなものの相互関係を活用するサービスが急増しており、これらのシステム上では特徴のある膨大な情報が蓄積されつつあります。Google では〝自分の興味を惹きそうな本〟は調べられませんが、このような関係性をもつデータに対して今回のような手法を適用すれば、簡単に計算できそうです。

また、Web ページ上の記述は、かならずしも情報の性格を正しく表現しているわけではありません。しかし、このようなシステム上のリンク関係は、ユーザーの意図を反映した行動にもとづいて自動的に生成されるため、情報の性格を正しく反映していると考えられます。自分の趣味や興味を正確に書き表すのは難しく、それを仔細に書いて Web で公開する人はほとんどいないでしょう。本の購入や SNS への入会といった行動はユーザーの興味や欲求の純粋な反映であり、その意味ではユーザーや情報の性格を正しく表現していることになります。新しく蓄積されたリンク情報を情報検索に有効活用する手法は、新世代の情報検索において重要になってくると思います。

今回利用したデータと Ruby ライブラリは私の Web ページ[4]で公開しているので、いろいろな演算にご利用ください。

（ますい・としゆき　産業技術総合研究所）

[参考文献]

[1] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom and John Riedl, “GroupLens: An Open Architecture for Collaborative Filtering of Netnews” In *Proceedings of the 1994 Computer Supported Collaborative Work Conference*, pp.175–186, 1994

4 http://pitecan.com/Enzan/