DBSJ Journal, Vol.7, No.4
March 2009

Locality Sensitive Hashing を用いたスケーラブルなハイブリッド型推薦システム

Scalable and Hybrid Recommender System using Locality Sensitive Hashing

森 紘一郎♦ 村上 知子♠
Kouichirou MORI Tomoko MURAKAMI

本研究の目的は、ユーザ数、コンテンツ数が増加しても高速に動作するスケーラビリティのあるハイブリッド型推薦システムを構築することにある。この目的を達するためクエリと類似したコンテンツを高速に検索可能な LSH というデータ構造を用いた。予めコンテンツとユーザを同一の LSH へインデキシングすることで、類似コンテンツ、類似ユーザの検索を同時に行う。その検索結果を用いることでコンテンツベース型推薦と協調フィルタリング型推薦を結合したハイブリッド型推薦を実現する。Book-Crossing データセットを用いて評価したところ従来の協調フィルタリングより 9 倍高速で統計的に有意な精度向上が確認できた。

The purpose of our work is to develop a scalable hybrid recommender system against the increase of users and contents. We adopt Locality Sensitive Hashing that is able to search similar contents to a query at the very high speed. Our method creates a LSH index of contents and users in advance, and search similar contents and users at the same time in the recommendation. Using the search results, we construct the hybrid recommender system. The evaluation of the recommender system with a book-crossing dataset has revealed that it is 9 times faster and statistically significant increases of precision than the traditional collaborative filtering method.

1. はじめに

近年、コンテンツのデジタル化にともない多数のコンテンツへのアクセスが可能になってきている。たとえば、書籍、Webサイト、ニュース記事、ブログ、番組、写真、音楽、動画などインターネット上には膨大なデジタル化されたコンテンツが蓄積されている。そのため、それらの膨大なコンテンツの中から興味のあるコンテンツをユーザが手動で探し出すのは難しくなってきている。このような状況を改善するためにユーザの好みをシステムが把握し、ユーザが好むコンテンツを提示するコンテンツ推薦システムの重要性が高まっている。

コンテンツ推薦システムは大きく分けるとコンテンツベース型と協調フィルタリング型に分類できる[1]。コンテンツベース型推薦は、コンテンツの内容に基づく手法の総称であり、ユーザが好むコンテンツと内容が類似したコンテンツを推薦するのが基本的な方針である。一方、協調フィルタリング型推薦は、他人のユーザプロファイルを利用する手法の総称であり、自分と好みの傾向が類似しているユーザを探し、そのユーザが好むコンテンツを推薦するのが基本的な方針である。

本研究で解決したい課題は二つある。一つめは、コンテンツベース型推薦と協調フィルタリング型推薦のトレードオフである。両者は互いに逆の特徴を持っていて相補的な関係にある[1]。そのため、どちらか単独の利用では不十分な推薦しかできないという課題がある。

二つめは、推薦のスケーラビリティである。スケーラビリティの高い（スケーラブルな）推薦システムとは、対象とするデータ規模（ユーザ数、コンテンツ数）が大きくなっても高速に動作するシステムを指す。先ほど述べたようにコンテンツベース型推薦は類似コンテンツを探す、協調フィルタリング型推薦は類似ユーザを探すのが基本的なアプローチである。そのため、従来のコンテンツベース型推薦では、コンテンツ数が増えるのにしたがって、協調フィルタリング型推薦では、ユーザ数が増えるのにしたがってスケーラビリティが低下するという課題がある。

一つめの課題を解決するハイブリッド型推薦アルゴリズムは、すでに多くのアルゴリズムが提案されているが、本論文では、上記二つの問題を同時に解決するスケーラブルなハイブリッド型推薦アルゴリズムを提案する。

近年、類似したコンテンツを高速に検索する手法あるいはデータ構造としてLocality Sensitive Hashing (LSH)が注目を集めている[2][3][4][5][6]。LSHは、コンテンツをあらかじめハッシュテーブルと呼ばれるデータ構造に格納する（インデキシング）ことでクエリコンテンツと類似したコンテンツを大規模なコンテンツ集合から非常に高速に検索できる。

本研究で提案するハイブリッド型推薦アルゴリズムLSHをベースにしている。提案手法が、従来のLSHを用いた類似コンテンツ検索システム[7][8][9]と異なる点は、同一のLSHにコンテンツとユーザを格納することでクエリと類似したコンテンツとユーザを同時に検索する点にある。LSHで検索した類似コンテンツをもとにコンテンツベース型推薦、類似したユーザをもとに協調フィルタリング型推薦を行い両者の結果を結合してハイブリッド型推薦を実現する。LSHはスケーラビリティに優れており、格納するコンテンツ数、ユーザ数が非常に大きくなっても類似検索が高速であるためユーザ数、コンテンツ数の増加に対してスケーラブルなハイブリッド型推薦が実現できる。

2章では、ハイブリッド型推薦、LSHを用いた類似コンテンツ検索に関する従来研究をまとめる。3章では、提案手法を具体的に説明する。4章では、提案手法の効果を確認するため大規模データセットを用いた評価実験結果について述べる。5章では、全体をまとめ、今後の課題を述べる。

2. 従来研究

推薦アルゴリズムは、主にコンテンツベース型と協調フィ

♦ 正会員 株式会社東芝 研究開発センター
kouichirou1.mori@toshiba.co.jp
♠ 株式会社東芝 研究開発センター
tomoko.murakami@toshiba.co.jp

ルタリング型に分類され、すでに多数のアルゴリズムが提案されている[1]。

ハイブリッド型は、コンテンツベース型と協調フィルタリング型の特徴を組み合わせた推薦手法であり、そのアプローチは四つに分類できる[1]。一つめは、協調フィルタリングとコンテンツベースを別々に行い、推薦結果のコンテンツやレイティングを結合する方法である[10]。二つめは、協調フィルタリングを元にしてコンテンツベースの特徴を織り込む方法である。たとえば、ユーザの嗜好ベクトルをコンテンツ ID の集合ではなく、単語などコンテンツ内容で表現し、嗜好ベクトルを用いて類似ユーザを検索する方法がある[11]。三つめは、コンテンツベースを元にして協調フィルタリングの特徴を織り込む方法である。たとえば、コンテンツ内容で各ユーザの嗜好ベクトルを表現し、Latent Semantic Indexing[12]で次元縮約する方法がある[13]。四つめは、コンテンツベースと協調フィルタリングを統一した確率モデルで表現する方法である[14]。提案手法のハイブリッド型推薦は、主に一つめのアプローチを用いている。また、嗜好ベクトルをコンテンツ内容で表現する方法を取れば（3.4 節）二つめのアプローチも用いることができる。

LSH を用いたスケーラブルな検索・推薦の研究として類似楽曲の検索[7]、ニュース記事の推薦[8]、関連 Web ページの検索[9]などがある。いずれもスケーラビリティを重視した研究であるが、クエリの類似コンテンツまたは類似ユーザのみを検索しており、提案手法のようなハイブリッド性は有していない。

3. LSH を用いたハイブリッド型推薦

3.1 システム構成

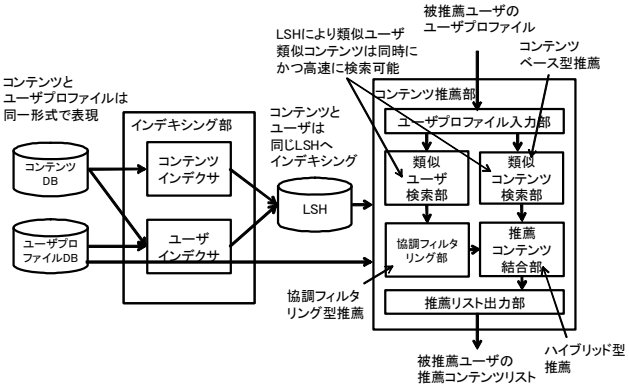

図 1 提案手法のモジュール構成
Fig.1 Module diagram of proposed method

図 1 は提案システムの全体構成図である。コンテンツ DB、ユーザプロファイル DB はそれぞれコンテンツ、ユーザプロファイルを格納するデータベースである。ユーザプロファイルは、ユーザの好むコンテンツ ID の集合で表現されるとする。インデキシング部は、コンテンツやユーザプロファイルを特徴ベクトルに変換し、LSH へ格納するモジュールであり、コンテンツをインデキシングするコンテンツインデクサとユーザプロファイルをインデキシングするユーザインデクサから成る。インデックスは LSH というデータ構造で表される。コンテンツとユーザプロファイルは同じ形式の特徴ベクトルで表現されると仮定しているためコンテンツのインデックスとユーザプロファイルのインデックスは同一の LSH に格納する。コンテンツ推薦部は、被推薦ユーザ（コンテンツ推薦を受けたいユーザ）に対してコンテンツを推薦するモジュールである。ユーザプロファイル入力部に被推薦ユーザのユーザプロファイルが入力されると類似ユーザ検索部において被推薦ユーザと嗜好の傾向が類似するユーザを LSH から検索する。また同時に類似コンテンツ検索部において被推薦ユーザの好むコンテンツと類似したコンテンツを LSH から検索する。協調フィルタリング部は、検索された類似ユーザから協調フィルタリングを用いて推薦コンテンツを選択する。推薦コンテンツ結合部は、類似コンテンツ検索部の推薦コンテンツと協調フィルタリング部の推薦コンテンツを結合することでハイブリッド型推薦を実現する。

提案手法は、コンテンツ、ユーザともに同一の LSH へ格納するためコンテンツ数、ユーザ数が多くなっても類似コンテンツ、類似ユーザの検索は高速である。以下、文書（Web ページ、書籍、番組メタデータなど）を推薦するシステムを例に提案手法を具体的に説明する。

3.2 文書のシグネチャへの変換

(a) 索引語・文書行列

	P1	P2	P3	P4	P5	P6
世界	1	1	0	1	0	0
遺産	1	1	0	0	0	0
背景	1	0	0	0	0	0
歴史	1	1	0	0	1	0
紹介	1	0	0	1	0	0
旅	0	1	1	1	0	0
番組	0	1	1	0	1	0
日本	0	0	1	0	0	0
温泉	0	0	1	0	0	0
グルメ	0	0	1	1	0	0
珍味	0	0	0	1	0	0
人物	0	0	0	0	1	0
モデル	0	0	0	0	1	0
アニメ	0	0	0	0	1	0
今日	0	0	0	0	0	1
料理	0	0	0	0	0	1

(b) 乱数列

H1	H2	H3	H4
13	14	11	10
2	3	15	13
7	15	16	14
14	6	7	12
10	5	10	9
3	11	6	5
6	8	13	15
1	1	2	8
15	4	9	7
12	16	1	3
11	7	12	11
5	12	8	2
9	9	3	6
8	2	14	1
4	10	5	4
16	13	4	16

(c) シグネチャ行列

	P1	P2	P3	P4	P5	P6
H1	2	2	1	3	5	4
H2	3	3	1	5	2	10
H3	7	6	1	1	3	4
H4	9	5	3	3	1	4

図 2 Min-Hashing の例
Fig.2 Min-Hashing example

図 2 (a) は、ある文書から作成した索引語・文書行列の例である。索引語・文書行列は、各行に索引語、各列に文書を割り当てた行列で、行列の値は文書が索引語を含む場合に 1、含まない場合に 0 を取る。この索引語・文書行列は、各文書の特徴ベクトルを表現している。たとえば、文書 P1 の特徴ベクトルは、P1 の列を縦に見た(1, 1, 1, 1, 1, 0, 0, …, 0)という 16 次元ベクトルである。今回は、索引語を含むか含まないかで 0 か 1 の値を用いたが、TF-IDF 法のスコアを用いることもできる。どのような値を用いるかはどの LSH を使うかに依存する。たとえば、この例では Jaccard 係数を用いたため 0 か 1 の 2 値とした。

次に、索引語・文章行列からシグネチャを作成する。シグネチャとは、特徴ベクトルの次元を縮約した要約表現である。シグネチャも特徴ベクトルと同じくベクトルで表現される。特徴ベクトルを次元縮約してシグネチャに変換する手法は、さまざまな方法が提案されているがここでは Min-Hashing [2][8]と呼ばれる手法を利用する。Min-Hashing は、索引語・

文書行列のような疎（要素の大部分が 0）な行列に適した次元縮約法である。Min-Hashing を使って次元縮約するためには、図 2 (b) のような複数の乱数列が必要になる。図 2 の(a)に Min-Hashing を適用すると(c)のシグネチャ行列が得られる。

Min-Hashing の大きな特徴は、2 つのシグネチャの一致する要素の割合が元の 2 つの特徴ベクトルにおける Jaccard 係数の近似値になっていることである。式で表すと

$$Sim_H(Sig(P1),Sig(P2)) \fallingdotseq Jaccard(F(P1),F(P2)) \quad (1)$$

である。ここで、F(P)はコンテンツ P の特徴ベクトル、Sig(P)はコンテンツ P のシグネチャ、$Sim_H(X,Y)$は、2 つのベクトル X、Y の一致する要素の割合、Jaccard(X,Y)は、2 つのベクトル X、Y の Jaccard 係数である。たとえば、図 2 (a) の索引語・文書行列において F(P1)と F(P2)の Jaccard 係数を求めると 3/7=0.43 となる。一方、シグネチャにおいて Sig(P1)と Sig(P2)の一致する要素の割合を求めると 2/4=0.5 となり、Jaccard 係数のよい近似となっている。乱数列の数を増やしてシグネチャの次元数を大きくするほど近似精度が上がる。Min-Hashing を用いてシグネチャへ変換すると Jaccard 係数ではなくシグネチャの一致する要素の割合で類似度を計算できる。

3.3 コンテンツインデキシング

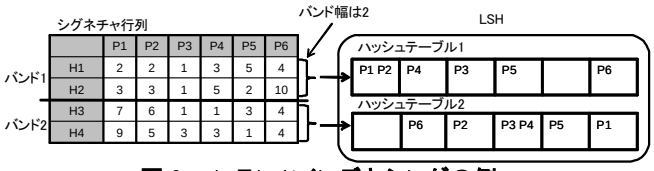

図 3 コンテンツインデキシングの例
Fig.3 Content indexing example

インデックスには、LSH というデータ構造を用いる。一般的なハッシュテーブルは「同一の」コンテンツが同一のビンに格納されるのに対し、LSH は、「類似した」コンテンツほど同一のビンに格納される確率が高くなるという特徴がある。

図 3 に示すように LSH は複数のハッシュテーブルで構成される。LSH は、どのような類似尺度を用いるかによって構成方法が大きく異なる。Jaccard 係数[2]、コサイン類似度[3]、Earth Mover's Distance (EMD)[3]、Lp ノルム[4]、ハミング距離[5]の LSH 構成方法は明らかになっている。ここでは、対象が文書なので Jaccard 係数を用いたが、先に述べたように特徴ベクトルをシグネチャに変換すると一致する要素の割合で類似度が近似できる。

図 3 は、コンテンツを LSH へインデキシングする例である。まず、シグネチャをいくつかのバンドに分割する。例では、バンド 1 とバンド 2 というバンド幅 2 の 2 つのバンドに分割した。次にバンドごとにハッシュテーブルを用意し、分割されたシグネチャをキーとして各コンテンツ ID をハッシュテーブルに格納する。たとえば、バンド 1 の P1 と P2 は同じシグネチャ (2,3) を持つのでハッシュテーブル 1 の同じビンに格納される。バンド 2 の P3 と P4 も同じシグネチャ (1,3) をもつのでハッシュテーブル 2 の同じビンに格納される。同じビンにハッシュされたコンテンツ同士はシグネチャの一部が同一であり、類似している確率が高いと考えられる。つまり、LSH へインデキシングすることで内容が似ているコンテンツを同じビンに集めることができる。たとえば、図 4 では、コンテンツ P1 と P2、P3 と P4 は互いに類似している確率が高い。

3.4 ユーザインデキシング

表 1 ユーザプロファイル
Table1 User profile

ユーザ ID	好きなコンテンツ ID
A	P1, P2, P5
B	P3, P4, P6
C	P1, P3

次にユーザプロファイル DB 内のすべてのユーザをユーザインデクサで LSH にインデキシングする。表 1 は、ユーザプロファイルの例である。ユーザプロファイルは、各ユーザがどのコンテンツを好むかを表している。たとえば、ユーザ A は P1、P2、P5 をユーザ B は P3、P4、P6 を好んでいる。

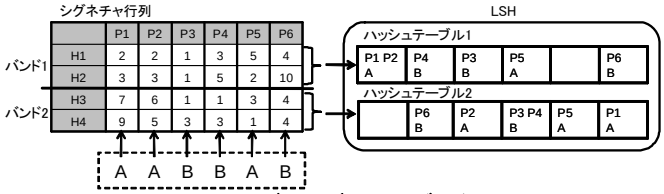

図 4 ユーザインデキシングの例
Fig.4 User indexing example

ユーザインデキシングは、すべてのユーザプロファイルを LSH へ格納する処理である。ユーザインデキシングでは、ユーザプロファイル内のコンテンツをすべて LSH へ格納する。このとき、格納するのはコンテンツ ID ではなく、ユーザ ID を格納するのがポイントである。コンテンツインデキシングと同様に LSH の同一のビンに入ったユーザは、互いに嗜好が類似したユーザである確率が高いと考えられる。

図 5 は、ユーザ A、B を LSH へインデキシングする例である。たとえば、ユーザ A をインデキシングすることを考える。ユーザ A のプロファイル中の文書は、P1、P2、P5 であるため、この 3 つの文書をバンドごとに LSH へハッシュする。ここで用いる LSH はコンテンツをインデキシングしたときに用いた LSH と同じものを使用する。そして、P1、P2、P5 のハッシュ先のビンへは、コンテンツ ID ではなくユーザ ID である A を格納する。ユーザ B も同様である。

ユーザインデキシングには、上記のようなユーザの好きなコンテンツ集合をハッシュするほかにもさまざまな方法が考えられる。たとえば、ユーザの好きなコンテンツ集合をコンテンツベクトルと同じ空間上のただ 1 つの特徴ベクトル（嗜好ベクトル）で表現するより洗練された手法が提案されている[1][11]。嗜好ベクトルが作成できたらコンテンツと同じく Min-Hashing で嗜好ベクトルをシグネチャへ変換してから LSH へユーザ ID を格納すればよい。この場合、各ユーザ ID はただ 1 つのビンに格納されることになる。

3.5 新規ユーザへの推薦

最後に新規ユーザへコンテンツを推薦する方法を説明する。図 5 は、ユーザ C へコンテンツを推薦する例である。ま

ず、LSH を用いてユーザ C の好きなコンテンツと類似したコンテンツを検索する。ユーザ C の好きな P1、P3 を LSH のすべてのハッシュテーブルに対してハッシングし、ハッシング先のビンに格納されているコンテンツを抽出する。この場合、類似コンテンツとして P1、P2、P3、P4 が得られる。ユーザ C はすでに P1 と P3 は知っているので除くと、P2 と P4 がユーザ C の好きなコンテンツに内容が類似するコンテンツとして推薦候補となる。ここでは、少なくとも 1 つのハッシュテーブルでハッシュ値が一致したコンテンツを類似コンテンツの候補と見なしている。ここで用いた推薦方式はコンテンツの内容に基づいているためコンテンツベース型の推薦である。ただし、LSH を用いてコンテンツをインデキシングしているためコンテンツ数に対してスケーラブルであり、膨大なコンテンツが対象であっても高速に類似コンテンツを探し出せるのが特徴である。

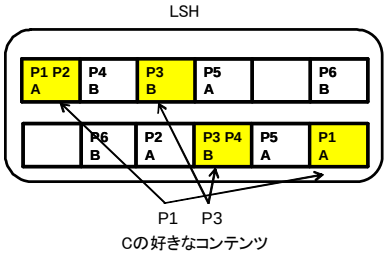

図 5 新規ユーザへの推薦
Fig.5 Recommendation for new user

次に、LSH を用いてユーザ C と嗜好が類似したユーザを検索する。先ほどと同様ユーザ C の好きなコンテンツ P1 と P3 を LSH のすべてのハッシュテーブルに対してハッシングし、ハッシング先のビンに格納されているユーザを抽出する。この場合、類似ユーザとして A、B が得られる。このユーザ A と B は、被推薦ユーザ C と好きなコンテンツが一部共通しているユーザである。つまり、被推薦ユーザ C と嗜好が似ているユーザの候補だと考える。

次に嗜好が似ているユーザの候補を用いて協調フィルタリングを実行する。この例では、もっとも簡単な方法であるが、嗜好が似ているユーザが好んでいるコンテンツの中で被推薦ユーザがレイティングしていないコンテンツを推薦することにする。ユーザプロファイルからユーザ A の好きなコンテンツは P1、P2、P5 であり、ユーザ B の好きなコンテンツは P3、P4、P6 である。ここから C が好きなコンテンツ P1 と P3 を除くと P2、P5、P4、P6 となり、ユーザ C へ推薦される。これらのコンテンツは、被推薦ユーザ C と嗜好が類似したユーザに基づいているため協調フィルタリング型の推薦と見なせる。ただし、LSH を用いてユーザをインデキシングしているためユーザ数に対してスケーラブルであり、膨大なユーザが対象であっても高速に類似ユーザを探し出せるのが特徴である。さらに洗練されたコンテンツ選択手法として、類似ユーザの好きなコンテンツの累積頻度をカウントして、頻度の大きい順に推薦するなどの方法がある[16]。

最後に、コンテンツベース型と協調フィルタリング型の推薦コンテンツを結合する。評価実験では、単純に両方の推薦コンテンツの上位 N 件ずつを結合する方法を取った。より複雑な方法として、推薦の開始時はコールドスタート問題を避けるためにコンテンツベース型を重視した推薦にし、ユーザの履歴が蓄積されてきたら協調フィルタリング型重視に切り替える方法やユーザにどちらを重視するか選択する手段を提供する方法も考えられる。

4. 評価実験

提案手法の有効性を確認するため評価実験を行った。従来の協調フィルタリングと提案手法の間で推薦時間や推薦精度を比較した。実験に用いた機器は、CPU Xeon X5482 3.2GHz 4 コア、メモリ 8GB である。プログラムはすべてスクリプト言語 Python で実装した。

4.1 テストデータセット

テストデータには、Book-Crossing Dataset[15]という公開データセットを用いた。このデータは、書籍の交流サイトをクローリングして収集したデータであり、各ユーザの書籍へのレイティング情報が含まれている。ユーザ数は 278,858、コンテンツ(書籍)数は271,379、レイティング数は1,149,780である。Book-Crossing Dataset は、協調フィルタリングの実験用に収集されたデータのためコンテンツ情報として書籍の ISBN、著者名、タイトル名、出版社、出版年という限られたデータしか付与されていない。そこで、Amazon Web サービス API を用いて ISBN をキーとして Amazon から書籍内容を収集し、書籍内容に含まれるすべての単語を索引語とした。Book-Crossing Dataset のレイティングは、1-10 の 10 段階である。レイティングは大きいほど好きなことを意味する。評価実験では、レイティングが 5 以上の書籍のみ好きな書籍と見なしてユーザプロファイルに追加した。

4.2 インデキシング時間とインデックスサイズ

表 2 インデキシング時間
Table1 Indexing time

処理	実行時間(秒)
コンテンツのシグネチャへの変換	73276.47
コンテンツインデキシング	273.00
ユーザインデキシング	6.35

表 2 は、コンテンツインデキシングとユーザインデキシングに要した実行時間である。コンテンツからシグネチャへの変換に約 20 時間ともっとも多くの時間がかかっており、LSH への格納自体はコンテンツ、ユーザ合わせて 15 分程度である。コンテンツインデキシングに比べてユーザインデキシングの時間が極端に短いのは、提案手法がコンテンツとユーザを同一の LSH へ格納するためである。つまり、コンテンツインデキシングが確保したメモリ領域にユーザ ID を格納するだけなのでユーザインデキシングは短時間で済む。一方、コンテンツインデキシングは、メモリ確保処理の分だけ余計に時間がかかっている。

以下の実験では、コンテンツのシグネチャ、ユーザプロファイル、インデックス（LSH）は、すべてメモリにロードした状態で実行時間を測定した。使用したメモリ領域は約 2.7 ギガバイトである。

4.3 LSH バンド幅の影響

LSH において、ある類似度を持つ 2 つの文書を類似と判断するかはバンド幅で調節できる。図 6 は、シグネチャを

120 次元としたときの 2 つの文書間類似度（Jaccard 係数）と LSH 上で 2 つの文書が類似と判定される（少なくとも 1 つのハッシュテーブルで同じビンに入る）確率の関係を表したグラフである。たとえば、文書間の Jaccard 係数が 0.5 の 2 つの文書があるとする。シグネチャを 120 次元とし、バンド幅を 2 とするとこの 2 つの文書は、ほぼ 1 の確率で類似文書と判定される。グラフからバンド幅を大きくしていくと類似文書と判定される確率が低くなっていくことがわかる。たとえば、バンド幅 5 のときは 0.5 の確率で類似文書と判定され、バンド幅 12 のときは、Jaccard 係数 0.5 の 2 つの文書は類似と判定されない。つまり、バンド幅が大きいほど 2 つの文書が類似と判定される基準が厳しくなる。バンド幅のサイズ決定は、対象としているデータに依存するため設計者がどの程度まで類似と判定するか判断する必要がある。ここでは、試行錯誤の結果、8 または 12 とした。

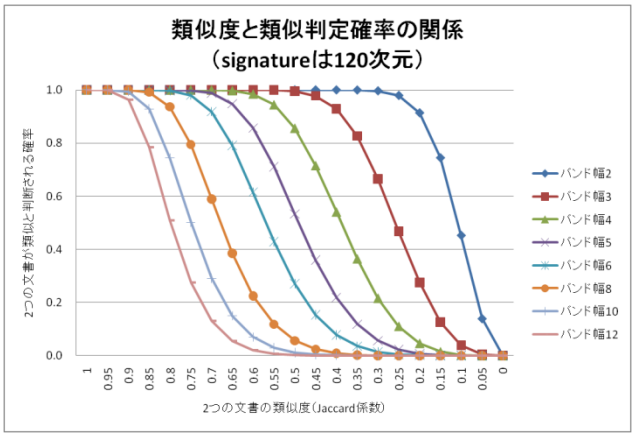

図 6 LSH バンド幅の影響
Fig.6 Effect of band-width

4.4 推薦時間と推薦精度

実験では、レイティングデータを 80%の訓練データと 20%のテストデータに分割した。推薦精度の測定には、テストデータからレイティング数の多い 100 人を選択し、その 100 人に対して訓練データを用いて 10 件の書籍を推薦する Top-10 推薦方式[16]を用いた。推薦精度は、100 人分の平均適合率、平均再現率である。上記の手順を 5 回繰り返し、5-fold Cross Validation で評価した。

図 7 は、各手法の平均推薦時間と平均推薦精度を示した結果である。CF は、従来のユーザベース協調フィルタリング[1]であり、線形探索で類似ユーザを探している。LSH_CF は、LSH を用いて類似ユーザ候補を探す方法である。括弧の中の数字は LSH のバンド幅である。ユーザインデキシングには、好きな書籍の集合でプロファイルを表す手法を用いた。CF、LSH_CF ともに類似ユーザを探すための類似指標には、コサイン類似度を用い、コサイン類似度が高い上位 k 人を類似ユーザとした。LSH_HYBRID は、ユーザインデキシングとコンテンツインデキシングを両方行い、協調フィルタリングの推薦結果とコンテンツベースの推薦結果を結合した推薦手法である。推薦結果の結合には、協調フィルタリングの推薦結果から上位 5 件、コンテンツベースの推薦結果から上位 5 件を選択した。

推薦時間

図 7 からわかるように、従来の線形探索を用いた協調フィルタリングより LSH_CF は 15 倍、LSH_HYBRID は 9 倍高速という結果になった。この高速化は、ユーザインデキシング、コンテンツインデキシングによって類似ユーザの検索、類似コンテンツの検索が高速化されたためである。

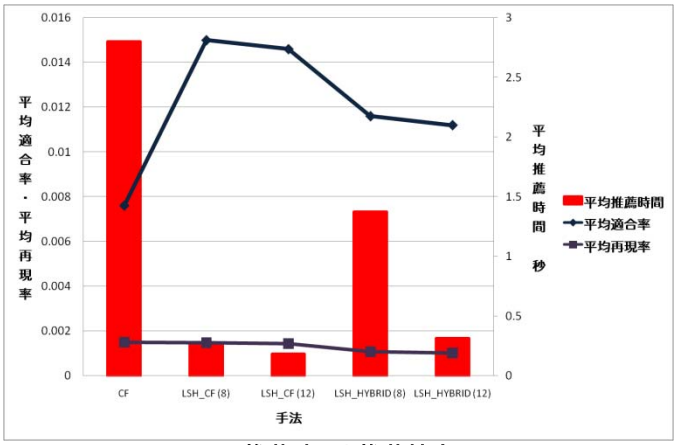

図 7 推薦時間と推薦精度
Fig.7 Recommendation time and accuracy

ハイブリッド型推薦では、コンテンツベース型の類似検索も行うが、被推薦ユーザの好きなコンテンツと類似したコンテンツの候補が多いことが問題点として明らかになった。たとえば、図 7 の LSH_HYBRID のバンド幅 8 とバンド幅 12 の平均推薦時間を比べるとバンド幅 8 の方が長くなっている。これは、先に述べたようにバンド幅を緩めることで類似するコンテンツの候補が非常に多く検索されてしまいその中から類似度が高い上位 k のコンテンツを絞り込む必要があったためである。しかし、ハイブリッド型推薦は、コンテンツベースと協調フィルタリングベースを組み合わせた時間にもかかわらず従来の協調フィルタリングより推薦時間が短いという結果が得られている点では提案手法の有効性が示せた。

従来のコンテンツベース型推薦や協調フィルタリング型推薦は、コンテンツ数やユーザ数の増加に対して線形オーダー O(n) に計算時間が増加するのに対し、LSH では、ハッシングに要する計算量が O(1)なので計算時間の増加幅は小さい。つまり、LSH を用いた方法はスケーラビリティがあり、実験で用いたデータよりユーザ数やコンテンツ数が増えても提案手法の計算時間の増加幅は小さいと予想できる。ただし、バンド幅を大きくしないと類似ユーザ、類似コンテンツの候補が増えてしまい候補の中から k-NN を検索する時間が増加する可能性は高い。候補をどのように絞り込むかは今後の課題となった。

推薦精度

LSH_HYBRID の推薦適合率は 1.1%と非常に低い値を示したが CF の推薦精度を 5%有意に上回るという結果になり、ハイブリッド推薦の効果がある程度は確認できた。しかし、LSH_CF と比較するとそこにコンテンツベースの推薦結果を追加した LSH_HYBRID の精度が低くなってしまった。

これは、Amazon から取得した書籍内容記述が不十分という根本的な問題もあるが、協調フィルタリング型の推薦コンテンツとコンテンツベース型の推薦コンテンツの結合方法の問題でもある。誤りを解析した結果、協調フィルタリング型、

コンテンツベース型のどちらか一方の方法で推薦して正解となるコンテンツが存在したが、結合する際に正解コンテンツが選ばれず精度を下げてしまうケースが散見された。たとえば、協調フィルタリング型とコンテンツベース型の上位から各5件で選んだところ下位5件にランクインできた正解を見逃していた。

今回の実験で適合率の絶対値が低い理由は以下のように考えられる。適合率を正しく測定するには、ユーザはデータセット中の好きな書籍をすべてチェックして事前にレイティングしておかなければならない。しかし、このデータは実データであり、すべてのユーザがすべての書籍に対して事前にレイティングしておくのは不可能である。Book-Crossing Dataset は、書籍数が 20 万件と非常に多く、ユーザはそのうちほんの一部の書籍にしかレイティングをつけていないことを考えると、実験結果のように適合率が低くなるのは必然といえる。実際、データを提供した Ziegler の論文[15]でもユーザベース協調フィルタリングの Top-10 推薦の適合率は 3.69%と報告しており、今回測定した協調フィルタリングの適合率に近い値である。推薦された非適合コンテンツがレイティング対象である可能性も考慮し、推薦結果に対するユーザフィードバックを得るなど別の方法での評価が重要だと考える。

5. おわりに

本論文では、従来の推薦システムが持つ推薦のスケーラビリティ、コンテンツベース型と協調フィルタリング型のトレードオフという 2 つの課題を解決するために Locality Sensitive Hashing (LSH) を用いたハイブリッド型推薦システムを提案した。推薦時間は従来の協調フィルタリングに比べて大幅な改善が得られたが、ハイブリッド型推薦の精度向上が確認できなかったのは課題となった。嗜好をベクトルとして表現する方法など嗜好モデリングに関する研究は多くなされており、推薦精度の向上に向けて今後検討していく必要がある。

今回の実験を通して LSH はコンテンツ数の増加に対する類似検索のスケーラビリティが非常に高いことがわかった。Book-Crossing Dataset より大規模なデータであっても十分なスケーラビリティが確保できると考えている。

今後の課題としてテキストだけではなく、画像、音楽、動画などのマルチメディアデータの類似検索・推薦を対象にしていきたい。

【文献】

[1] G. Adomavicius and A. Tuzhilin, Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions, IEEE Trans. on Knowledge and Data Eng. 17(6), pp.734-749, 2005.
[2] A. Z. Border, On the Resemblance and Containment of Documents, Proc. Compression and Complexity of Sequences, 1998.
[3] M. S. Charikar, Similarity Estimation Techniques from Rounding Algorithms, Proc. 34th Ann. ACM Symposium on Theory of Computing, pp.380-388, 2002.
[4] M. Datar, N. Immorlica, P. Indyk and V. S. Mirrokni, Locality-Sensitive Hashing Scheme based on p-Stable Distributions, Proc. 20th Ann. Symposium on Computational Geometry, pp.253-262, 2004.
[5] A. Gionis, P. Indyk and R. Motwani, Similarity Search in High Dimensions via Hashing, Proc. 25th VLDB Conf., 1999.
[6] M. Slaney and M. Casey, Locality-Sensitive Hashing for Finding Nearest Neighbors, IEEE Signal Processing Magazine 25(2), pp.128-131, 2008.
[7] R. Cai, C. Zhang, L. Zhang and W. Y. Ma, Scalable Music Recommendation by Search, Proc. 15th Intl. Conf. Multimedia, pp.1065-1074, 2007.
[8] A. Das, M. Datar and A. Garg, Google News Personalization: Scalable Online Collaborative Filtering, Proc. 16th Intl. Conf. WWW, pp.271-280, 2007.
[9] T. H. Haveliwala, A. Gionis, D. Klein and P. Indyk, Evaluating Strategies for Similarity Search on the Web, Proc. 11th Intl. Conf. on WWW, pp.432-442, 2002.
[10] M. Claypool, A. Gokhale and T. Miranda, Combining Content-based and Collaborative Filters in an Online Newspaper, ACM SIGIR '99 Workshop Recommender Systems: Algorithms and Evaluation, 1999.
[11] M. Pazzani, A Framework for Collaborative, Content-Based, and Demographic Filtering, AI Review, pp.393-408, 1999.
[12] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, R. Harshman, Indexing by Latent Semantic Analysis, Journal of the American Society for Information Science, Vol.41, pp.391-407, 1990.
[13] I. Soboroff and C. Nicholas, Combining Content and Collaboration in Text Filtering, Proc. Intl. Joint Conf. AI, 1999.
[14] R. Popescul, L. H. Ungar, D. M. Pennock and S. Lawrence, Probabilistic Models for Unified Collaborative and Content-based Recommendation in Sparse-data Environments, Proc. 17th Conf. on Uncertainty in AI, 2001.
[15] C. N. Ziegler, S. M. McNee, J. A. Konstan and G. Lausen, Improving Recommendation Lists Through Topic Diversification, Proc. 14th Intl. Conf. on WWW, 2005.
[16] B. Sarwar, G. Karypis, J. Konstan and J. Riedl, Analysis of Recommendation Algorithms for E-Commerce, Proc. 2nd ACM Conf. on Electronic Commerce, pp.158-167, 2000.

森 紘一郎 Kouichirou MORI
2005 年早稲田大学大学院理工学研究科修士課程修了。同年(株)東芝入社。現在、同社研究開発センターシステム技術ラボラトリーに勤務。主に機械学習、データマイニング、推薦システムの研究開発に従事。情報処理学会、日本データベース学会、人工知能学会各会員。

村上 知子 Tomoko MURAKAMI
1998 年慶應義塾大学大学院政策・メディア研究科修士課程修了。同年(株)東芝入社。現在、同社研究開発センターにて、機械学習、データ・テキストマイニング技術とその応用の研究開発に従事。博士（政策・メディア）。情報処理学会、人工知能学会各会員。2007 年度人工知能学会全国大会優秀賞受賞。