東海大学大学院2011年度 修士論文

擬準同型性を満たす類以度による
秘匿分散協調フィルタリングプロトコル

The Distributed Collaborative Filtering Protocol
based on Quasi-homomorphic Simirality

指導教員 菊池 浩明 教授

東海大学大学院 工学研究科 情報理工学専攻

9KDRM001 青木 良樹

目 次

第1章　序論

1.1　背景

現在，我々は家から出ることなく様々な買い物を楽しむことができる．Amazon[1)]に始まり，楽天[2)]など多くのインターネットショッピングサイトが我々の生活に定着している．いつでも，家にいながら簡単にボタンをクリックするだけで欲しい商品を注文することができ，次の日には手元に届く．そこで目にするのが,「おすすめ商品」や「セール情報」などの広告である．これらの情報は，ショッピングサイトだけにかかわらず，多くのウェブサイトで見かけることができる．いわゆる，情報推薦システムである．

この情報推薦は，カスタマーのユーザ登録時に入力した情報や，商品の閲覧履歴，検索ワードなどの「好み」を基にデータマイニングをすることによって実現されている．データマイニングとは，商品の閲覧履歴や購入履歴などの蓄積されるデータを解析し，データを見ただけでは分からないような,「モノ」のパターンや相関関係を探しだす技術である．その代表的な手法の一つに**協調フィルタリング** (Collaborative Filtering) がある．協調フィルタリングは複数のユーザによって複数のアイテムが評価付けされているデータベースにおいて，他のユーザの値をもとに，評価されていないアイテムの評価値を予測する技術である [4, 12, 13, 14]．本論文はこの協調フィルタリングを主に扱っていく．

一方で，クラウドコンピューティング技術が発達し Google[3)]を始めとする様々な企業が，新しいサービスを世の中に発信している．その結果，多くのユーザが様々なデータをインターネット上に保存している．クラウドコンピューティングとは，インターネットを経由して様々なソフトウェアやハードウェアなどのコンピュータ資源を利用することができるサービスである．

例えば，手元のコンピュータに文書作成ソフトがインストールされていなくても，インターネットに接続すれば文書作成ソフトが使用することが可能になる．他にも，手元のコンピュータには少しの保存容量がない場合でも，オンラインストレージと呼ばれるクラウドサービスを利用することによって，インターネット上にある保存領域を使用することができ

[1)] アメリカの大手インターネットショッピングサイト．本から家電，衣料品まで様々な標品を取り扱っている．アメリカや日本だけでなく現在 7 カ国で利用されている．(2011 年 7 月 28 日)

[2)] 日本の大手インターネットショッピングサイト．幅広いジャンルをカバーしており，個人で出店することも可能．(2011 年 7 月 28 日)

[3)] アメリカの大手検索サイト．検索エンジンだけにかかわらず，メールや，カレンダー，OS など様々なサービスを基本的に無料で提供している．(2011 年 7 月 28 日)

る．有名なサービスとして電子ノートブックサービスを提供している Evernote[4)]や，オンラインストレージサービスを提供している Dropbox[5)]などが存在する．

1.2 目的

現在の情報推薦は，自組織が管理する顧客のデータのみを活用して行われている． もし，単一の組織が管理する顧客の秘密情報だけではなく，クラウドコンピューティング技術によってインターネット上に保存されている情報や，他の組織の管理するユーザの情報を学習することができれば，より的確に商品の推薦やサービスの提供などを行うことができるのではないかと考えられる．

しかし，これらの情報は組織にとって，外部に知られたくはない重要な秘密の情報であり，プライバシを保護しなければならない対象でもある．インターネット上に保管されている個人情報は，内容が外部に知られないように，暗号化されている．

本研究では，情報推薦をするとき複数組織の管理しているデータベースを一つに統合し評価値を予測する問題を考え，組織間で所持している，垂直分割されたデータベースを統合せず，分散したまま統合したデータベースから得られる予測値とほぼ等しくなるような協調フィルタリングを行う方式を提案する．

この分散計算をするにあたり，準同型性をもつ公開鍵暗号を用いてナイーブに協調フィルタリングに適用すると，暗号文の生成回数がユーザ数に比例し，非常に大きなコストが発生してしまう [5, 6, 11]．

そこで，本論文では以下の三つのアプローチで分散環境における，秘匿計算の手法を提案すると共に，パフォーマンスの向上を試みる．

1. **擬準同型類似度の導入**
   和集合について準同型性を満たす類似度を導入し，分散したままでの類似度計算を行う．

2. **事前計算**
   評価値を事前計算し，再利用することによって暗号化にかかるコストを削減する．

3. **k-近傍による計算**
   k-近傍のユーザについてのみ類似度を評価することで，評価にかかるコストの削減を試みる．

また，試験実装に基づいて，提案方式の計算量の改善，推薦精度，安全性を評価する．

[4)]電子ノートブックのクラウドサービス．ちょっとしたメモから，ウェブサイトのクリップ，PDF，画像など様々な情報を保存することができ，後で参照や検索が可能である．OCR 機能も備わっており，画像の中に書いてある文字も検索することができる．

[5)]オンラインストレージサービス．無料で最大 4GB の保存領域を利用することができる．

1.3 論文構成

本論文の構成は次の通りである.

第 2 章で情報推薦に関する基礎や，方式の種類，情報推薦を行うにあたって考慮すべき点などについて説明し，第 3 章で，要素技術や擬準同型類似度の定義，提案方式について説明する．そして，第 5 章で精度や処理性能を報告し，第 6 章で本論文の結論を述べ，今後の課題を示す.

第２章
情報推薦

2.1　情報推薦とは

情報推薦とは，まだ自分が所持していない，または知らないであろう商品や情報を予測し，推薦することである．

インターネットを利用すれば分かるように，膨大な情報や商品が溢れている．これらの増え続けていく膨大な情報を隅から隅まで見ることは不可能であり，その中から自分にとって有益なものを見つけ出すことは非常に困難だといえる．そこで，利用されているシステムが推薦システム (Recommender System) である．

最も身近な例は，インターネットショッピングサイトの Amazon でよくみられる，おすすめ商品である．Amazon のページにアクセスをすると，“閲覧履歴からお勧め”(図 2.1) や “これにも注目”(図 2.2) などの項目に商品が表示される．これらは，閲覧履歴や以前購入した商品から推測された情報である．

図 2.1: Amazon による “閲覧履歴からお勧め” された商品

この第２章では，この情報推薦（システム）にはどのような種類や，方法があるのかを説明していく．

図 2.2: Amazon の推薦によって “これにも注目” に表示された商品

2.2 情報推薦に利用するデータの種類

推薦システムで利用する方法は，大きく**内容ベースフィルタリング** (Content based Filtering) と，**協調フィルタリング** (Collaborative Filtering) の二種類に分けられる．この節では，この二つの手法について簡単に説明する．

2.2.1 内容ベースフィルタリング（Contant based filtering）

内容ベースフィルタリングは，はじめにユーザがどのような情報を欲しがっているかを入力してもらい，その内容にあったフィルタリング[1)]結果を提示する方式である．例えば，あるユーザが「1990 年代」,「アクション映画」という情報を提示した場合，それにマッチした「Toy Story(1995)」や「Jumanji (1995)」,「GoldenEye (1995)」をフィルタリング結果として提示する．

この内容ベースフィルタリングの利点は，ユーザの要望を基に情報をフィルタリングするため，求めている情報を提示し易いといえる．逆に欠点は，ユーザがわざわざ情報を提示しないとフィルタリングが出来ないという点や，検索される範囲が限定されるため，別ジャンルの意外な推薦が出来ないという点がある．

2.2.2 協調フィルタリング（Collaborative Filtering）

協調フィルタリングとは内容ベースフィルタリングとは異なり，ユーザ間，またはアイテム[2)]間の類似度[3)]を利用してフィルタリングを行う方式である．この方法は，自分と嗜好が

[1)] 膨大な情報に対して，不要な情報を取り除く為，本論文ではこの操作を「フィルタリング」と呼ぶこととする．

[2)] ここで，アイテムとは団体が取り扱っている商品や情報などの事を指す．

[3)] 類以度とは対象 A と対象 B がどのくらい似ているか（類似しているか）を示す指標である．一般的に数値が高ければ高いほど似ているとみなされる．

似ているユーザを類以度によって探し出し，その人がお勧めするアイテムをフィルタリング結果として返す事によって実現され，最も有名な手法として広く利用されている．つまり，自分と同じ嗜好を持った人が居たとして，その人が好きなアイテムは「おそらく」自分も好きであろうと予測するのである．

メモリベース法（Memory-Based Methods）

メモリベース法は，ユーザが評価した値をそのまま利用して予測する方法である．

ユーザが評価した値の例を表 2.1 に示す．u_j は j 番目のユーザを示し，i_ℓ は ℓ 番目のアイテム示す．例えば，ユーザ u_1 がアイテム i_6 に与えた評価は 5 となる．

この方法には，アイテム間の類以度に注目したアイテムタイプと，ユーザ間の類以度に注目したユーザタイプの二種類がある．表 2.1 を映画に対する評価だとする．u_2 に新しい映画を推薦しようとした時に，まず u_2 と類似しているユーザを探す．最も，類似しているユーザが u_4 だったとすると，u_2 がまだ観たことがなく，なおかつ u_4 が高い評価をつけている映画は i_7 になる．最終的に推薦システムは u_2 に i_7 を推薦することが出来る．

これらの詳細については，2.3 節で具体的な例を用いて紹介する．

表 2.1: メモリベース法で利用されるデータ例

	i_1	i_2	i_3	i_4	i_5	i_6	i_7
u_1	1	1	3	4	2	5	
u_2	3	2		5	5	1	
u_3		2	3	2			1
u_4	4			4		2	5

モデルベース法（Model-Based Methods）

モデルベース法は，クラスタリングなどを行い，先にモデルを構築しておき，推薦対象者をそのクラスタに当てはめ，そのクラスタ内で推薦する方式である．

クラスタリングとは，ある集合 U を重複部分が無いように分類することで，分類されたそれぞれのグループをクラスタと呼ぶ．クラスタリングの手法に関しては最短距離法や最長距離法による階層型クラスタリングや，k-means や SOM などの非階層型クラスタリングがある．

例えば，ある集合 U を 3 つのクラスタ C_1, C_2, C_3 に分類すると，$U = C_1 \cup C_2 \cup C_3$ となり，$\phi = C_1 \cap C_2 \cap C_3$ となる．ユーザ u_1 と各クラスタとの距離（類以度）を求め，最も近いクラスタ無いで評価が高いアイテムを推薦する方法である．

2.3 類以度の求め方

協調フィルタリングのメモリベース法において，類以度を計算する対象として，ユーザ間とアイテム間の二つがある．

2.3.1 アイテム間（Item-to-Item）

アイテム間類以度を利用するタイプは，アイテムに与えられた評価値をアイテムのベクトルとみなし，二つのアイテムベクトル間の類以度を計算する方法である．

これは，好きなアイテムと似ているアイテムであれば，そちらも好きであろうというというアイディアである．Amazon でよくみられる “これにも注目” などの，既に閲覧したアイテムと似たような（類以度の高い）アイテムが表示されるのは，この方式ではないかと考える．

第 3 章で提案する変形ユークリッド距離（Modified Euclideam Disatance）を利用して，i_1 とその他のアイテムの類以度を計算した例を表 2.2 に示す．

ここで $s_{j,\ell}$ はアイテム i_j とアイテム i_ℓ の類以度を表す．つまり，i_1 と i_2 の類以度 $s_{1,2}$ は 0.5 となる．

表 2.2: アイテム間の類以度

	i_1	i_2	i_3
u_1	1	1	5
u_2	1	1	4
u_3	2	2	2
u_4	4	4	1
u_5	5	4	1
s_{u_1,u_k}	-	0.5	0.02

2.3.2 ユーザ間（User-to-User）

ユーザ間類以度を利用するタイプは，ユーザに与えられた評価値をユーザのベクトルとみなし，二つのユーザベクトル間の類以度を計算する方法である．

これは，似たような嗜好を持ったユーザが好きなモノであれば，そのユーザも好きであろうというというアイディアである．Amazon で言うと，“この商品を買った人はこんな商品も買っています” に相当するのではないかと考える．

アイテム間と同様に，変形ユークリッド距離を利用して，u_1 とその他のユーザの類以度を計算した例を表 2.3 に示す．

ここで $s_{j,\ell}$ はユーザ u_j とユーザ u_ℓ の類以度を表す．つまり，u_1 と u_2 の類以度 $s_{1,2}$ は 0.5 となる．

表 2.3: ユーザ間の類以度

	i_1	i_2	i_3	$s_{1,\ell}$
u_1	1	1	5	-
u_2	1	1	4	0.5
u_3	2	2	2	0.08
u_4	4	4	1	0.03
u_5	5	4	1	0.02

2.4 データベースにおける分散環境のタイプ

また，本研究では分散環境におけるデータベースを対象にしているため，データベースにおける分散環境のタイプについて説明する．

データベースの分割モデルは**垂直分割**（Vertically Partitioned Model）と**水平分割**（Horizontally Partitioned Model），混合分割に分けられる．

2.4.1 垂直分割（Vertically Partitioned Model）

垂直分割はデータベースを縦に分割したモデルである．

Party A と Party B がいた場合，この二つのパーティは，共通ユーザの情報は所持しているが，異なったアイテムの情報を持っているような場合が垂直分割モデルである．

今回提案する方式はこの垂直分割モデルを想定している．

Party A と Paety B における垂直分割の例を表 2.4 に示す．

表 2.4: Party A と Paety B における垂直分割の例

	Party A				Party B		
	i_1	i_2	i_3	i_4	i_5	i_6	i_7
u_1	1	1	3	4	2	5	
u_2	3	2		5	5	1	
u_3		2	3	2			1
u_4	4			4		2	5

2.4.2 水平分割（Horizontally Partitioned Model）

水平分割はデータベースを横に分割したモデルである．

Party A と Party B がいた場合，この二つのパーティは，共通アイテムの情報は所持しているが，異なったユーザの情報を持っているような場合が水平分割モデルである．Party A と Paety B における水平分割の例を表 2.5 に示す．

また，各ユーザ毎にスライスした P2P モデルもこの水平分割に分類される．

表 2.5: Party A と Paety B における水平分割の例

		i_1	i_2	i_3	i_4	i_5	i_6	i_7
Party A	u_1	1	1	3	4	2	5	
	u_2	3	2		5	5	1	
Party B	u_3		2	3	2			1
	u_4	4			4		2	5

2.4.3 混合分割（Arbitrarily Partitioned Model）

混合分割は水平分割と垂直分割を合わせたモデルである．

混合モデルには垂直分割や水平分割で示されるような単純なパターン存在しない．

2.5 セレンディピティ（Serendipity）

セレンディピティとは，モノを探している時に，偶然に素晴らしいモノに出会うことを指す．

好きなものを持っているときは，好きなものと似ているものも持っている可能性が高い．

例えば，新しく発売された音楽 CD を買った時，その CD の歌手が過去に発売した CD が推薦されることが多々ある．もし，その CD を所持していないのであれば購入するきっかけになるかもしれない．

しかし，多くの場合，新発売 CD を買うようなユーザは過去の CD も既に持っている可能性が高い．

したがって，情報推薦において，意外性や真新しさは重要な要素である．今まで自分が知らなかった新しい発見や，自分が想像していないものを推薦されたときにセレンディピティが高いという．

内容ベースフィルタリングに比べ，協調フィルタリングの方が，このセレンディピティが高いと言われている [13]．

これは，内容ベースフィルタリングは，始めに対象の情報を入力して推薦するアイテムの範囲を絞ってしまうため，フィルタリング条件にマッチしないものはその時点で弾かれてしまう．そのため，意外な推薦がし辛いのである．

また，アイテム間類以度を利用した協調フィルタリングよりも，ユーザ間類以度を利用した協調フィルタリングの方がセレンディピティの点に関して有利であると言われている．

2.6 考えられるユーザの特徴

情報推薦をするにあたって考慮しなくてはならないポイントの一つに，ユーザ特徴がある．ここで言う，ユーザの特徴とは評価値の付け方や購買率を言う．以下に代表的なユーザの特徴例を示す．

1. 辛口ユーザ
   どんなアイテムも辛口に評価するため，評価値の平均が低い．
2. 甘口ユーザ
   どんなアイテムも甘口に評価するため，評価値の平均が高い．
3. 評価が極端
   お気に入りのアイテムは極端に高く，好きではないアイテムは極端に低いため，評価値の分散が大きい．
4. 購買率が高い
   多くのアイテムを購入しているユーザで，そのユーザに関するスパース率が低い．
   スパース率とは，全アイテム中いくつ評価されているかという，データベースの密度を表す．データベースが**疎**であるとこの値は高く，**密**であると低い値を示す．
5. 購買率が低い
   アイテムの購入数が少ないため，そのユーザに関するスパース率が高い．

表 2.6 に評価値にみられるユーザの特徴例を示す．ここで，σ はそのユーザが評価値の標準偏差を表す．

2.7 一般的なデータセットの特徴

Movie Lens Dataset[8] や Netflix[9] のデータセットにみられる特徴は，非常にスパース（疎）であるということである [1]．おそらく，Amazon で扱われている商品は数えきれないほど存在する一方で，各ユーザが購入してる商品はほんの 1% にも満たない．実際に，自分

表 2.6: 評価値にみられるユーザの特徴例

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	mean	σ	sparsity
辛口ユーザ	1	1	2	3	2	3		2	0.81	85%
甘口ユーザ	4	3		5	5	3		4	0.89	71%
評価が極端		1	5	1	4	5	1	2.83	1.86	85%
購買率が高い	4	2	2	4	3	2	5	3.14	1.12	100%
購買率が低い	4				3			3.5	0.5	28%

のケースに当てはめてみても，Amazon で販売されている商品を買った個数は 20 個にも満たない．

これらのスパースなデータベースに対応するため，Breese らによって標準投票（Default Voting）などが提案されている [3]．この手法は，デフォルト値を決定しておき，欠損値を補完する手法で，最も中立的な値である投票値の平均が利用される．

また，高島らによって Prediction Voting[15] が提案されている．Default Voting による欠損値の補完を，平均値ではなく相関係数法による予測結果にするというアイディアである．

しかし，評価方法のばらつきやノイズが存在する為，そのまま適応すると予測性能が低下してしまうという問題がある．そこで，平山らは Prediction Voting において，いかに良いフィードバック行うかを提案し，その精度向上を図っている [4]．

2.8 従来研究

多くの情報推薦に関する研究がなされてきた．ここでは，プライバシを保護した従来研究を紹介する．

プライバシ保護には，大きく分けて**暗号による保護**と**摂動化による保護**の二つのアプローチが研究されている．

暗号による保護は，準同型暗号を使い，値を暗号化したまま計算できるため精度に優れるているが，暗号文生成や暗号計算になど膨大な計算コストがかかる．また，暗号文のサイズが大きいため通信コストも大きくなってしまう．

摂動化による保護は，値に乱数などを加え計算をするため，暗号を使ったアプローチに比べ，正確な計算をすることができないが，計算コストが高い暗号計算をする必要がなく，コストが安いメリットを持っている．

本研究では，暗号化によるアプローチを行っている．

2.8.1 暗号化（Encryption）によるアプローチ

Canny は特異値分解（Singular Value Decomposition）を準同型暗号を用いて，情報を秘匿したまま協調フィルタリングを行う手法を提案している．木澤らは，コサイン類以度と準同型暗号を用いた秘匿協調フィルタリング [5] や，秘匿性集合プロトコルを利用してプライバシを保護した協調フィルタリングを提案している [6]．これらは水平分割された P2P モデルを想定している手法である．

また，多田らは秘匿関数計算を利用し水平分割，アイテムタイプの秘匿協調フィルタリングを提案している [7].

佐久間らは P2P 環境で非同期に値を秘匿したまま平均を計算プロトコル [49] や，値を秘匿したまま k-means クラスタリングを行う手法を提案している [50].

Vaidya らは，水平分割で Nïve bayes を利用したクラスタリングを秘匿したまま行う手法 [17] や，秘匿アソシエーションルールマイニングを提案している [18].

2.8.2 摂動化（Perturbation）によるアプローチ

Agrawal らは，Randomized Response を利用して決定木学習を行う方式を提案している [19].

また，Polat らによって加法摂動化による協調フィルタリング方式 [21] が提案され，Huang らがそのアタック方式と，改良方式を提案している [20].

望月らは， Randomized Response を利用した摂動方式で摂動化し，アイテムタイプの協調フィルタリングを提案している [22].

第3章
擬準同型類似度を用いた秘匿分散協調フィルタリング

3.1 要素技術

3.1.1 準同型暗号（Homomorphic Encryption）

暗号とは

現在，インターネットでは勿論，様々なデバイスで暗号技術が利用されている．そもそも，暗号とは何かというと，メッセージを第三者が見ても内容が分からないように，元の情報を加工することである．イメージとしては，伝えたいメッセージを箱に入れて鍵を掛けることに相当する．まず，暗号化されていない情報のことを平文（Plain Text) と呼び，暗号化されたものを暗号文（Cipher Text）と呼ぶ．暗号文を生成するには平文を鍵（Key）と呼ばれる秘密情報と暗号アルゴリズムが必要である．

普及している暗号技術は大きく分けて，共通鍵暗号と公開鍵暗号の二つに分類される．共通鍵暗号とは，我々が普段から利用している家の鍵などと同じで，暗号化に利用する鍵と復号時に利用する鍵が等しい暗号のことを言う．代表的なものに，DES やトリプル DES, AES などがある．共通鍵暗号の欠点は，相手にどのようにして安全に鍵を渡すかという問題や，一人につき一つの鍵が必要であるという問題がある．相手も同じ鍵を所持していなければ復号することができないため，如何にして安全に鍵を渡すかという問題に対しては，有名な鍵共有の方式として，Diffie-Hellman 鍵共有がある．鍵の数に関しては，5 人の間で相互に情報をやりとりする場合，各ユーザは自分以外の 4 人と通信するために 4 個の鍵を管理する必要がある．

それに対し，Diffie-Hellman 鍵共有のアイディアを基に公開鍵暗号が発明された．これは暗号化に利用する鍵と，復号に利用する鍵が共通のものではなく，公開鍵と秘密鍵という別々の鍵を使う方式である．代表的なものとして．RSA 暗号 [23] や，ElGamal 暗号 [24] などがある．

準同型暗号（Homomorphic Encryption）

平文を秘匿したまま，平文同士を加算 (乗算) した暗号文を計算することができる性質を加法 (乗法) に関して準同型性を持つ，という．RSA 暗号 [23] や，ElGamal 暗号 [24] は乗法準同型性（Multiplicative Homomorphic）を満たす暗号として有名である，

また，加法準同型性（Additive Homomorphic）を満たす暗号として，Modified-ElGamal 暗号，Paillier 暗号 [25] などがある．本研究では加法に関して準同型性を持つ Paillier 暗号を用いた．

Paillier 暗号

入力を平文 m，暗号化関数を $E(\cdot)$ とした時，Paillier 暗号は次の三つの性質を持つ．

1. 異なる暗号文
   g を生成元，r を乱数，n を大きな二つの素数 p, q の合成数 pq とした時，暗号文の生成は，
   $$g^m r^n \pmod{n}^2 \tag{3.1}$$
   で計算される．そのため，同じ平文から暗号文を生成しても，乱数 r が異なるため生成された暗号文は一致しない．このとき，g, n は公開鍵，p, q は秘密鍵となる．

2. 平文の加算
   暗号文同士を乗算することによって，平文の加算を行うことが出来る．
   $$E(m_1) * E(m_2) = E(m_1 + m_2). \tag{3.2}$$

3. 平文の乗算
   暗号文に平文をべき乗する事によって，平文の乗算を行うことが出来る．
   $$E(m_1)^{m_2} = E(m_1 * m_2). \tag{3.3}$$

3.1.2 協調フィルタリング（Collanorative Filtering）

協調フィルタリングは，ユーザ間あるいはアイテム間の類似度に基づき，未知のアイテムに対する評価値を予測するアルゴリズムである．

U をユーザの集合，$n = |U|$ とする．I_A，I_B をそれぞれ，組織 A と B が管理しているアイテムの集合とする．$m_A = |I_A|$，$m_B = |I_B|$ とし，その和を $I = I_A \cup I_B$，$m = m_A + m_B$ とする．ユーザ u によるアイテム i の評価値を $r_{u,i}$ と示す．

ユーザ u のアイテム i に対する予測値 $\hat{r}_{u,i}$ は次の式で定める．

$$\hat{r}_{u,i} = \overline{r}_u + \frac{\sum_{v \in U-\{u\}} s_{u,v}(r_{v,i} - \overline{r}_v)}{\sum_{v \in U-\{u\}} s_{u,v}} \tag{3.4}$$

ただし，ここで $s_{u,v}$ はユーザ u と v の類似度，$\overline{r}_u$ はユーザ u の平均評価値を示す．

3.2 擬準同型類似度（Qasi-homomorphic similarity）

3.2.1 擬準同型類似度とは

分散計算に望ましい類似度は，べき等律 $s_{u,u} = 1$，交換律 $s_{u,v} = s_{v,u}$，推移性 $s_{u,o} + s_{o,v} \geq s_{u,v}$ を満たすことに加えて，データセットの和に対して**準同型性**を持つことが望ましい．すなわち，ユーザ u の評価値を m 次元ベクトル $(r_{u,1}, \ldots, r_{u,m})$ とする．この m 次元ベクトルを m_A 次元の $\boldsymbol{a}_u$ と m_B 次元の $\boldsymbol{b}_u$ の和 $\boldsymbol{a}_u + \boldsymbol{b}_u$ で表すとき，ユーザ $\boldsymbol{a}$ と $\boldsymbol{b}$ の類似度 $\boldsymbol{a} * \boldsymbol{b}$ を与える演算が，

$$(\boldsymbol{a}_u + \boldsymbol{b}_u) * (\boldsymbol{a}_v + \boldsymbol{b}_v) = f(\boldsymbol{a}_u * \boldsymbol{b}_u, \boldsymbol{a}_v * \boldsymbol{b}_v)$$

を満たす演算 f が存在することと定める．例えば，$\boldsymbol{a} * \boldsymbol{b} = I_A \cup I_B$ に対して，$f(x, y) = x + y$ である．ピアソンの相関係数やコサイン尺度などは準同型性を満たさない．一方，ユークリッド距離は準同型性を満たすが，類似度になっていない（類似しているほど大きな値）．

準同型性を満たしていないが，局所的な類似度の合成（右辺）で全域的な 類似度を近似できるとき，すなわち，ある小さな定数 ϵ があり，

$$|(\boldsymbol{a}_u + \boldsymbol{b}_u) * (\boldsymbol{a}_v + \boldsymbol{b}_v) - f(\boldsymbol{a}_u * \boldsymbol{b}_u, \boldsymbol{a}_v * \boldsymbol{b}_v)| < \epsilon$$

であるとき，この類似度は ϵ **擬準同型性**を満たすという．全ユーザの類似度の和が 1 になるように，類似度 $s_{u,v}$ の**正規化**を

$$\tilde{s}_{u,v} = \frac{s_{u,v}}{\sum_{\ell \in U-\{u\}} s_{u,\ell}} \tag{3.5}$$

と定める．組織 A の持つアイテム集合 I_A について求めた類似度を**局所類似度**と呼び，$s^A_{u,v}$ と書く．組織 B についても同様に $s^B_{u,v}$ とする．

3.3 定義

3.3.1 モデル

本研究では，二組織間で秘匿したまま協調フィルタリングをおこなう方式を提案する．その組織を A と B とする．また，U をユーザの集合，$n = |U|$ とし．I_A，I_B をそれぞれ，組織 A と B が管理しているアイテムの集合とする．$m_A = |I_A|$，$m_B = |I_B|$ とし，その和を $I = I_A \cup I_B$，$m = m_A + m_B$ とする．ユーザ u によるアイテム i の評価値を $r_{u,i}$ と示す．

3.3.2 予測方式

本研究ではメモリベースのユーザタイプ協調フィルタリングを利用し，分割方式は垂直分割とする．

また，本研究の目的は秘匿したまま分散計算を行うことであるため，式 (3.4) を一部簡略化した次式を利用する．

$$\hat{r}_{u,i} = \frac{\sum_{v \in U-\{u\}} s_{u,v} r_{v,i}}{\sum_{v \in U-\{u\}} s_{u,v}}. \tag{3.6}$$

更に，A と B はプロトコルに従うが，互いに信用していない状況，すなわち，semi-honest モデルを仮定する．

3.3.3 評価値の値域

アイテムに与えられる評価値は，未評価 ϕ を含めた $\{\phi, 1, 2, 3, 4, 5\}$ の五段階，六種類とする．1 が低評価で 5 を高評価とする．ただし，表などでは未評価値を空白で表記する．

3.3.4 欠損値の扱い

欠損値とは，評価されていない評価値を示す．本研究では，欠損値を次のように扱う．ユーザ間類以度を求める時，欠損値は考慮せず，二ユーザ間の共通集合を利用して計算する．ただし，予測対象アイテムのベクトルに欠損値が含まれていた場合，この値を五段階評価の中央値の 3 とする．

3.4 類以度評価関数

ここで，本研究でユーザ間類以度を求める為に利用した類以度評価関数を説明する．$\mathbf{a}$ をユーザ u_a の持つ評価値ベクトル $(a_1, \ldots, a_n)$，$\mathbf{b}$ をユーザ u_b の持つ評価値ベクトル $(b_1, \ldots, b_n)$ とする．

3.4.1 ユークリッド距離（Euclidian Distance）

ユークリッド距離は

$$EuclideanDistance(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2} \tag{3.7}$$

で定義される．この値は 0 以上を取り，0 に近い値を示せば二つのベクトルの距離は近く，値が大きければ距離は遠くなる．

ユークリッド距離を二次元空間で表した例を図 3.1 に示す．

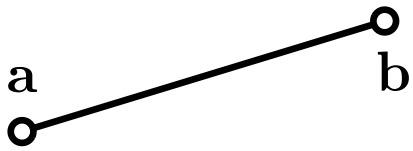

図 3.1: **a** と **b** 間のユークリッド距離

3.4.2 変形ユークリッド距離 （Modified Euclidian Distance）

式 (3.7) で示したユークリッド距離は上限値がない．それに加え，類以度とは値が高ければ似ている，低ければ似ていないというものであるため，ユークリッド距離は類以度である条件を満たしていない．

そこで，ユークリッド距離の二乗に 1[1)]を加算し，逆数をとることによって，これらの条件を満たした変形ユークリッド距離を式 (3.8) に示す．

$$ModifiedEuc(\mathbf{a},\mathbf{b}) = \frac{1}{1+\sum_{i=1}^{n}(a_i-b_i)^2} \tag{3.8}$$

この変形ユークリッド距離は，本論文では，主にこの類以度を用いて進めていく．

3.4.3 コサイン類以度（Cosine Measure）

コサイン類以度は二つのベクトル間の角度を類以度としたもので，次の式で定義される．

$$cos(\mathbf{a},\mathbf{b}) = \frac{\mathbf{a}\cdot\mathbf{b}}{\|\mathbf{a}\|\|\mathbf{b}\|} \tag{3.9}$$

ここで，分子は **a** と **b** の内積（Inner Product）である．

$$\mathbf{a}\cdot\mathbf{b} = \sum_{i=1}^{n} a_i * b_i \tag{3.10}$$

また，分母はそれぞれの距離ノルムの積で表される．距離ノルムは

$$\|\mathbf{a}\| = \sqrt{\sum_{i=1}^{n} a_i^2} \tag{3.11}$$

で計算される．

コサイン類以度を二次元空間で表した例を図 3.2 に示す．

コサイン類以度の値域は -1 から 1 で，類以度が 1 時と，-1 の時の例をそれぞれ図 3.3, 図 3.4 に示す．

[1)]式 (3.8) の分母に加算されている 1 は，ユーザ間のユークリッド距離の総和が 0 になってしまった時に，0 で除算してしまう事を防ぐスムージングパラメータである．

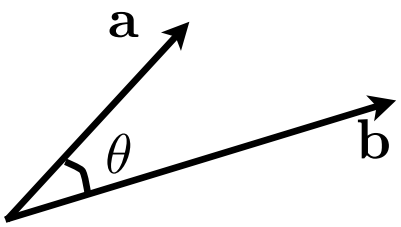

図 3.2: **a** と **b** の成す角度

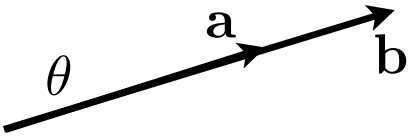

図 3.3: 類以度が 1 の時の例

3.4.4 ピアソン相関係数（Pearson Correlation Coefficient）

ピアソン相関係数は二つのベクトル間の相関度類以度にしたもので，精度が高く広く利用されている．

ピアソン相関係数は次の式で定義される．

$$PCC_{a,b} = \frac{\sum_{k=1}^{N}(r_{a,k} - \overline{r}_a)(r_{b,k} - \overline{r}_b)}{\sqrt{\sum_{k=1}^{N}(r_{a,k} - \overline{r}_a)^2 \sum_{k=1}^{N}(r_{b,k} - \overline{r}_b)^2}} \tag{3.12}$$

ピアソン相関係数の値域は -1 から 1 で，1 に近ければ近いほど**正の相関**が強く，-1 に近ければ近いほど**負の相関**が強い，0 の時は**無相関**を表す．

3.4.5 スピアマン順位相関係数（Spearman Rank Correlation Coefficient）

ピアソン相関係数の式 (3.12) において，値の代わりに順位を利用したものをスピアマン順位相関係数という．スピアマン順位相関係数は

$$Spear(a,b) = \frac{\sum_{k=1}^{N}(rank_{a,k} - \overline{rank}_a)(rank_{b,k} - \overline{rank}_b)}{\sqrt{\sum_{k=1}^{N}(rank_{a,k} - \overline{rank}_a)^2 \sum_{k=1}^{N}(rank_{b,k} - \overline{rank}_b)^2}} \tag{3.13}$$

で定義される．

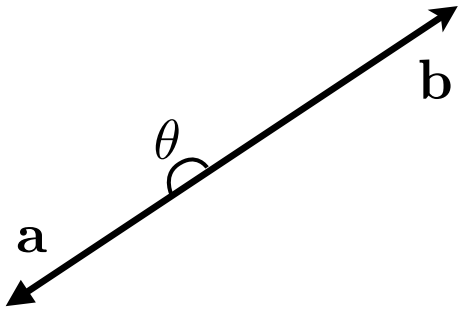

図 3.4: 類以度が -1 の時の例

しかし，五段階評価のような狭い離散的な定義域では，同順位が多く発生してしまうため，高い精度は得ることができないと言われている [16].

3.4.6 ソマーズの d 類以度（Somers' d Mearsure）

ソマーズの d 類以度 [2] は，複雑な計算を必要とせず簡単な演算で計算することができる．

また，前述した類以度はベクトル同士の計算であるため，欠損値が存在すると Default Voting や Prediction Voting などで補完したり，二つの集合間の積集合（Intersection）を取得する必要がある．

しかし，ソマーズの d 類以度は積集合ではなく和集合（Union）でも計算することが出来る．

ソマーズ d 類以度は Concordance, Discordance, Tied, Number of item, を数えることによって計算される．また，表 3.1 の値を例に説明していく．

表 3.1: Somers' d 類以度のサンプルデータ

	i_1	i_2	i_3	i_4	i_5	$\overline{r}$
u	4	2	5	1	5	3.4
v	1	4	3		4	3

まず，次式で $f_{u,i}$ を計算する．

$$f_{u,i} = r_{u,i} - \overline{r}_u \tag{3.14}$$

次に，この $f_{u,i}$ を使い次の条件に従い，C, D, T, N の四つをカウントする．

1. **C（Concordance）**

$$f_{u,i} > 0 \wedge f_{v,i} > 0,$$

$$f_{u,i} < 0 \wedge f_{v,i} < 0.$$

2. **D（Discordance）**

$$f_{u,i} > 0 \wedge f_{v,i} < 0,$$
$$f_{u,i} < 0 \wedge f_{v,i} > 0.$$

3. **T（Tied）**

$$f_{u,i} = 0,\ f_{v,i} = 0,$$
$$r_{u,i} = \phi,\ r_{v,i} = \phi.$$

4. **N（total Number of item）**
   アイテムの総数．また，N は $N = C + D + T$ を満たす．

表 3.1 の値をについて f を計算した結果を表 3.2 に示す．

表 3.2: 表 3.1 について f の計算結果

	i_1	i_2	i_3	i_4	i_5
$f_{u,*}$	1	-1	1.6	-2	2
$f_{v,*}$	-2	1	0		1
	D	D	T	T	C

そして次式で Somers' d 類以度は定義される．

$$Somers'(u,v) = \frac{C-D}{N-T} \tag{3.15}$$

表 3.2 の例では，$C = 1, D = 2, T = 2, N = 5$ となり，類以度は

$$Somers'(u,v) = \frac{1-2}{5-2} = -0.3 \tag{3.16}$$

と計算される．

Somers' d 類以度は -1 から 1 の値を取り，1 に近ければ一致度は高くなり，-1 に近ければ一致度は低くなる．全て Tied の時は分母が 0 になってしまうため，特別な処理が必要である．

Lathia らによると，簡単な計算で，ピアソン相関係数を用いた場合と同等の精度を得ることが出来る [1]．また，推移性（Transitivity）を持っている．推移性とは，$A \subset B, C \subset A$ である時，$C \subset B$ が成り立つことを言う．この類以度においての推移性とは，r_a, r_b, r_c について図 3.5,3.63.7 に示す関係が成り立つ．ただし，r_c は $r_c \neq \bar{r}_c$, $r_c \neq \phi$ とする．また，具体的な数値例を図 3.8 に示す．

Lathia らはこの性質を使った水平分割環境におけるユーザタイプの秘匿協調フィルタリング方式を提案している [1]．

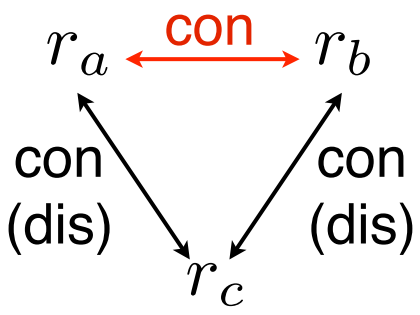

図 3.5: Concordance のケース

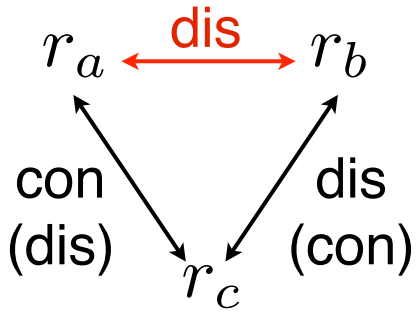

図 3.6: Discordance のケース

3.5 概要

準同型性を満たす公開鍵アルゴリズムを用いることで，二者間で互いの値を秘匿したまま，式 (3.6) を評価する (ナイーブな方式)．しかし，類似度を分散して計算する時に，次の二つの問題が挙げられる．

1. 評価に用いるユーザ数 $(n^2 - n)$ の数だけ暗号文を計算する必要があり，計算が非常に複雑になる．
2. 通信コストもユーザ数 n に比例し，多くのユーザを所持する大規模データベースで推薦することは現実的に困難である．

そこで，本稿では次の三つの提案を行う．

1. 擬準同型性を満たす類似度
   類似度に着目し，分散計算に適しているかどうかを検討する．各組織について正規化した類似度をそれぞれのデータセットの大きさに比例して平均をとることで，全体の

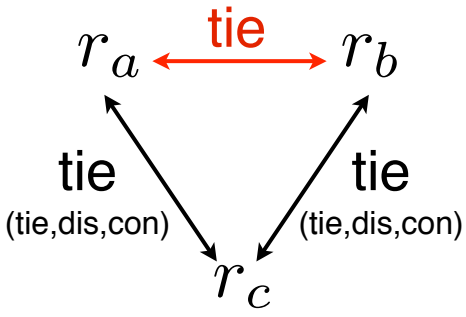

図 3.7: Tied のケース

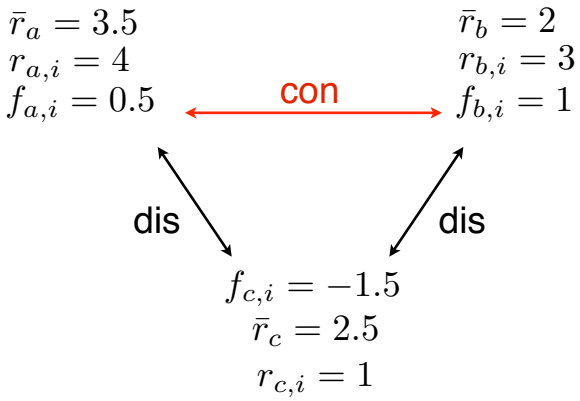

図 3.8: 推移性を示す具体的な数値例

類似度を近似する．この合成類似度を，準同型性に準ずる擬準同型性を満たしていることとして特徴付ける．

2. 評価値の事前計算
   推薦システムで用いられる評価値は離散的な小さな定義域で与えられることが多いことに着目し，事前に計算した値を組み合わせることで計算コストを削減する．

3. k 近傍による推薦
   n に比例する通信コストを削減する方法として，よく用いられている k 最近傍による推薦を行う．ただし，全域的な近傍はコストがかかるので，局所的に近傍を定める．

3.6 Naive 秘匿分散協調フィルタリング

ユーザ u のアイテム o に対する評価値の予測値 $\hat{r}_{u,o}$ は，

$$\hat{r}^{AB}_{u,o} = \frac{\sum_{v\in U-\{u\}} r_{v,o}/(1+r^A_v+r^B_v)}{\sum_{v\in U-\{u\}} 1/(1+r^A_v+r^B_v)} \tag{3.17}$$

$$= \frac{\sum_{v\in U-\{u\}} r_{v,o}\prod_{\ell\neq v}(1+r^A_\ell+r^B_\ell)}{\sum_{v\in U-\{u\}}\prod_{\ell\neq v}(1+r^A_\ell+r^B_\ell)} \tag{3.18}$$

であることに注意すると，各組織内で計算できる部分と，組織を超えて計算する部分からなることがわかる．ただしここで，

$$r^A_v = \sum_{i\in I_A}(r_{v,i}-r_{u,i})^2 \tag{3.19}$$

とするこの処理を秘匿したままで全域的な類似度にする推薦値 $r^{AB}_{u,o}$ を計算するプロトコルを Algorithm 1 に示す．

Algorithm 1 ナイーブな CF 法

入力: A の評価値 $r_{u,i}$ $i\in I_A$，B の評価値 $r_{u,i}$ $i\in I_B$

出力: 推薦値 $\hat{r}_{u,o}$

1. A は I_A の全てのアイテム i について，式 (3.19) に従って局所的に r^A_i を計算する．
2. B は I_B の全てのアイテム i について，式 (3.19) に従って局所的に r^B_i を計算する．
3. B は $E(r_{v,o}\prod_{i\neq v} r^B_i)$ を A に送る．
4. A は，$y=\prod_{v\in U} E(\prod_{\ell\neq v} r^B_\ell r_{u,o})^{r^A_v}$，
   $z=\prod_{v\in U} E(\prod_{\ell\neq v} r^B_\ell r_{u,o})$ を送り返す．
5. B は，$r^{AB}_{u,o}=\frac{D[y]}{D[z]}$ により推薦値を得る．

3.7 基本方式

ナイーブな方式で利用した式 (3.18) は計算が複雑な上，ユーザ数に比例して計算量が大きくなる．そこで，基本方式では式 (3.18) の代わりに次の式で求める．

$$\hat{r}^{A+B}_{u,o} = r^A_{u,o}\cdot w_A + r^B_{u,o}\cdot w_B \tag{3.20}$$

ここで，w は組織の重みを表し，次式で定義する．

$$w_A = \frac{m_A}{m_A+m_B} \tag{3.21}$$

さらに，類似度に式 (3.8) を 1 に正規化した式 (3.5) で計算し，予測評価値 $\hat{r}_{u,o}$ を式 (3.22) とする．

$$\hat{r}_{u,i} = \sum_{v \in U-\{u\}} s'_{u,v} r_{v,i} \tag{3.22}$$

Algorithm 2 基本方式

入力: A の評価値 $r_{u,i}$ $i \in I_A$，B の評価値 $r_{u,i}$ $i \in I_B$

出力: 推薦値 $\hat{r}_{u,o}^{A+B}$

1. A は U の全ユーザ $v \in U - \{u\}$ について，ユーザ u との局所的な正規化類似度 $\tilde{s}_{u,v}^{A}$ を求める．

2. B は U の全ユーザ $v \in U - \{u\}$ について，ユーザ u との局所的な正規化類似度 $\tilde{s}_{u,v}^{B}$ を求める．B は次の $v \in U - \{u\}$ について，B の公開鍵による暗号文 $E[r_{1,o}], \ldots, E[r_{u-1,o}], E[r_{u+1,o}], \ldots, E[r_{n,o}]$ を計算して A に送信する．

3. A は，

$$y = E[r_{1,o}]^{\tilde{s}_{u,1}^{A}} \cdots E[r_{n,o}]^{\tilde{s}_{u,n}^{A}},$$

を計算して B に送信する．

4. B は秘密鍵を用いて y を復号して，B による正規化局所類似度 $\tilde{s}_{v,o}^{B}$ を用いて，予測推薦値

$$\hat{r}_{u,o}^{A+B} = \frac{m_B}{m} D[y] + \frac{m_A}{m} \sum_{v \in U-\{u\}} r_{v,o} \tilde{s}_{v,o}^{B} \tag{3.23}$$

を得る．

(例 3.7.1) 表 3.3 に簡単な数値例を次に示す．$*$ は予測したい値である．類似度の違いを表 3.4 に示す．

表 3.3: 基本方式の計算例データ

	A		B		
	i_1	i_2	i_3	i_4	i_5
u_1	4	3	5	1	
u_2	1	4			4
u_3	1		1	2	2
u_4	4	3	$*$	2	2

表 3.4: 組織による類似度の違い

	s_A	s_B	s_{AB}	s_{A+B}
u_1	0.839	0.294	0.750	0.512
u_2	0.076	0.117	0.100	0.101
u_3	0.083	0.588	0.150	0.386
u_4	-	-	-	-

式 (3.23) による合成類似度による予測 $\hat{r}_{u,o}^{A+B}$ は,

$$\hat{r}_{u,o}^{A+B} = \hat{r}_{u,o}^{A} \frac{m_A}{m} + \hat{r}_{u,o}^{B} \frac{m_B}{m}$$

と展開される．すなわち，局所的に予測した推薦値 $\hat{r}_{u,o}^{A}$ と $\hat{r}_{u,o}^{B}$ を m_A, m_B $(m_A + m_B = m)$ の比に応じて総和を秘匿して計算している．

3.8 事前計算方式

提案した方式 1 は $(n-1)$ 回の暗号化を必要とするため，大きな時間がかかってしまい実用的ではない．この $(n-1)$ 回の暗号化を計算しないようにするために評価値 R の事前計算を行う．一般的に評価値は $R = \{1, 2, 3, 4, 5\}$ など，離散的な小さな定義域であることが多い．そこで，予めこの評価値を $R' = \{E[1], E[2], E[3], E[4], E[5]\}$ のように暗号化しておくこととする．しかし，このままだと複数のアイテムが同じ評価値であった場合，同じ暗号文が利用されてしまう．これを防ぐために，0 の暗号文を十分な数 p だけ所有する，暗号化されたゼロの集合 $Z = \{E[0]_1, \ldots, E[0]_p\}$ を計算しておく．Paillier 暗号の暗号文は前述した式 (3.1) によって計算されるため，暗号化された $E[0]$ は $E[0]_1 \neq E[0]_2$ である．そのため，各アイテムの評価値の暗号化は次の式 (3.24) で計算する．

$$E[r] = E[r] \cdot E[0] \tag{3.24}$$

3.9 k-近傍方式

方式 1 は $(n-1)$ 個の暗号文を送信しなくてはならないため，通信量がかかる．協調フィルタリングは類似度を重みとして評価値を計算するため，重みが小さいユーザの評価値は結果にあまり影響しない．そこで，各ユーザとの局所的な類似度を求め，上位 k 人をだけを選び，選ばれたユーザのみで方式 1，または方式 2 を行う．[2)]

[2)] Angorithm4 の Step1 で行う正規化は n ではなく，k について 1 に正規化しなおす．

Algorithm 3 事前計算

入力: A の評価値 $r_{u,i}$ $i \in I_A$，B の評価値 $r_{u,i}$ $i \in I_B$

出力: 推薦値 $\hat{r}_{u,o}$

1. (事前) B は評価値 r の定義域 $\mathcal{D}$ の各値 x について $c_x = E(x)$ を計算する．十分な数 p だけ $d_1 = E(0), d_2 = E(0), \ldots$ を計算して Z とおき，安全に記憶しておく (乱数要素があるので，同一の 0 の平文でも $d_i \neq d_j$ である)．

2. (評価時) B は，評価値 x を計算するとき，任意の $d \in Z$ を選び,$E(x) = c_x \cdot d$ により x の暗号文を作成する．(以下は基本方式と同様に行う．A は基本方式と同一である)

Algorithm 4 k 近傍

入力: A の評価値 $r_{u,i}$ $i \in I_A$，B の評価値 $r_{u,i}$ $i \in I_B$

出力: 推薦値 $\hat{r}_{u,j}^{A+B_k}$

1. 基本方式の Step 1 と同一．

2. B は正規化した局所類似度 $\tilde{s}_{u,v}^{B}$ について，ユーザ集合 $U - \{u\}$ をソートする．上位 k 人のユーザ v からなる部分集合を $U(u,k)^B \subset U$ とする．$v \in U(u,k)^B$ について，ユーザ u との局所的な正規化類似度 $\tilde{s}_{u,v}^{B_k}$ を求め, 暗号化して $(v, E(r_{v,o}))$ $v \in U(u,k)^B$ を A に送信する．

3. A は，$U(u,k)^B$ の k 個の要素について，

$$y = \prod_{v \in U(u,k)^B} E(r_{v,j})^{\tilde{s}_{u,j}^{A_k}}$$

   を計算して B に送信する．

4. B は秘密鍵を用いて y を復号して，B による正規化局所類似度 $\tilde{s}_{v,o}^{B}$ を用いて，予測推薦値を式 (3.23) により求める．

第4章　非同期秘匿分散 k-means クラスタリング

4.1　非同期秘匿分散 k-means クラスタリング

また，前述していたアプローチとは異なったアプローチを提案する．先ほどの手法は協調フィルタリングのメモリベースによる予測を行った．

この，第 4 章ではモデルベースによるアプローチを利用したものを提案する．

4.2　はじめに

近年，クラウドコンピューティング技術や，ライフログの普及に伴い，多くの情報が電子化されて管理されてきている．これらのデータに基づいて，利用者の嗜好に応じた商品を推薦する情報推薦サービスなども利用が著しい．更に，特定の閉じたサービスの履歴だけではなく，インターネット利用者のウェブページの閲覧履歴をプロバイダ側で読み取り，ターゲットを絞った広告などに活用する DPI 技術 (Deep Packet Inspection) の検討も始まってきている [45, 46]．しかしながら，これらのサービスはプライバシの保護という課題がある．クラウドコンピューティング技術かプロバイダーで管理された利用者の嗜好や, 閲覧履歴が漏洩するリスクを考慮しなくてはならない．

この課題に対して，ユーザのプライバシを保護したまま，二者間でクラスタリングを行う研究が Vaidya らによって提案されている [47]．秘匿内積評価プロトコルを用いて，各々局所的に試算したクラスタの重心を漏らさずに，正しくクラスタリングを実行する試みである．

一方，Kowalczyk らは， n 台のノードが分散する P2P ネットワークにおいて各ノードが持っている情報を中央に集中させることなく平均を計算するプロトコル “Newscast” を提案している [48]．佐久間らはこの “newscast” を利用しプライベートな非同期平均計算プロトコル “Private Asynchronous Average Computation” を提案し [49]，k-means アルゴリズムを，プライバシを保護したまま実行する手法を提案している [50]．

PrivateAAC では，k-means クラスタリングを行う過程で，各クラスタの重心を秘匿して実行しているが，そこには大きな負荷がかかる．そこで，本研究ではその処理効率の向上を試みる．

本研究では，AAC に基づき，次の 2 点を改良する．(1) 各ユーザとクラスタの重心 (平均) を秘匿したまま効率良く計算するために，ユークリッド距離ではなくコサイン類似度を用いて計算を行い，各ノードのデータ，重心を秘匿したままもっとも類似度の高い重心を探索する手法について記述する．また，(2) 重心を公開，同期することなく各ノードがそれぞれクラスタリングを行う “思い込み非同期クラスタリング” について記述し，精度と効率を評価する．

4.3 基本研究

4.3.1 k-means クラスタリング

k-means クラスタリングとは，各クラスタの平均を求め，k 個のクラスタにクラスタリングをしていく方式である．まず，ノード数を n，クラスタ数を k とし，ユーザ u_i のデータを $\boldsymbol{x}_i$，クラスタ j の重心を $\boldsymbol{\mu}_j$ とする．$bx, \boldsymbol{\mu}$ は d 次元のベクトルである．初期状態でユーザ u_i はランダムに k 個のクラスタに振り分けられる．そして，各クラスタの平均値 $\boldsymbol{\mu}_j$ を計算する．各ユーザは自分の持っているデータ bx_i と各クラスタの平均値 $\boldsymbol{\mu}_j$ との距離を計算し，最も近い (類似している) クラスタに所属を変更する．これを収束するまで回繰り返すことによってクラスタリングを行う．$n = 5$，$k = 2$ の時の初期状態と 2 世代目の状態を図 4.1 に示す．

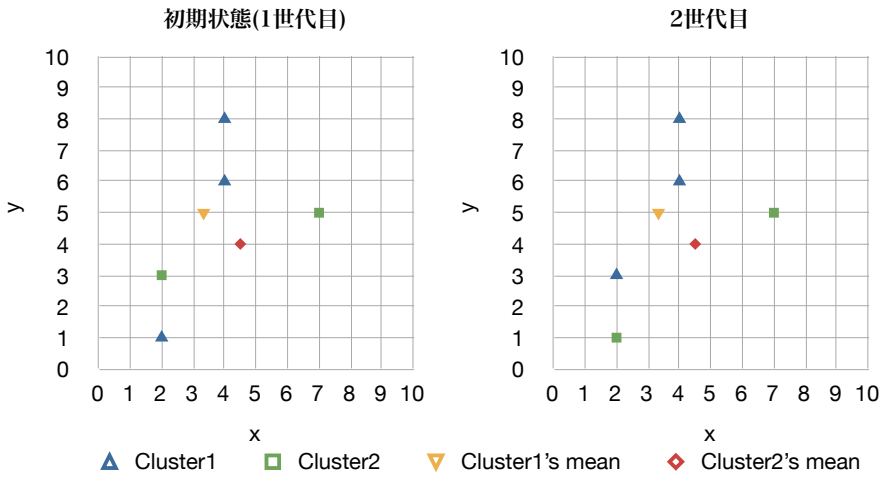

図 4.1: k-means クラスタリングの例

4.3.2 Newscast[48]

Kowalczyk らによって提案された，非同期に平均を計算するプロトコルである．P2P ネットワークにおいて各ノードが持っている情報を中央に集中することなく平均値 μ を計算する．あるノード i がある P2P ネットワーク内のノード j をランダムに選び，$\mu_i^{(t+1)} = \mu_j^{(t+1)} = \frac{\mu_i^t + \mu_j^t}{2}$ と二者間で平均を取る作業を複数回繰り返す．十分な長さの t_* サイクル後には全ノードのベクトルは $\frac{1}{n}\sum_i^n \mu$ となり，全体の平均値に近似する．t_* を収束地のサイクルとする．

$$\mu_1^{t_*} = \cdots = \mu_n^{t_*} \doteq \frac{\sum_i^n x_i}{n} \tag{4.1}$$

4.3.3 Private AAC[49]

佐久間らは，Newscast を応用して，秘匿したまま効率良く平均を求める非同期平均計算プロトコルを提案している．

準同型性を満たした暗号では，2 の割り算が実行できないので，次のようにして値を更新しる．サイクル t_i のユーザ i とサイクル t_j のユーザ j が，一般性失うことなく $t_i \geq t_j$ とすると，

$$\begin{aligned} Enc(\mu^{(t+j)}) &= Enc(\mu_i^{(t)}) \cdot Enc(\mu_j^{(t)})^{2^{t_i - t_j}} && (4.2)\\ &= Enc(\mu^t + 2^{t_i - t_j} \cdot \mu_j^t) && (4.3)\\ &= Enc(\mu_j^{(t+1)}) && (4.4) \end{aligned}$$

におり，各々の暗号文を秘匿したまま総和していく．最後に，$\mu_i^(*) = \frac{\mu_i^(t)}{2^T}$ とすることで，newscast を等価な平均を得る．

4.4 提案方式

4.4.1 概要

k-means によってクラスタリングする際には，各世代において重心を全ノードで共有する必要がある．本手法ではこの重心を全ノードで共有せず，局所的に所持している情報のみで自分の所属するクラスタを判別する．

4.4.2 提案プロトコル

ノード数を n，クラスタ数を k，ノードの持つデータ x の次元数を d とする．

各ノードを $\{u_1, \ldots, u_n\}$，サーバを S とし，d 次元のデータベクトル $\boldsymbol{x} = (x_1, \ldots, x_d)$，各クラスタの平均の集合 $M = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k\}$ を持つ．ここで，$\boldsymbol{\mu}_i$ は i 番目のクラスタの重心ベクトルである．また，サイクル数 t と世代 g の定義を図 4.2 に示す．

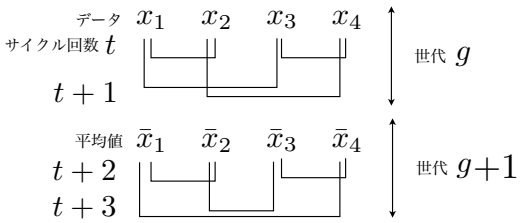

図 4.2: サイクル t と世代 g の定義

Step 1 各ノード u_i は自分の所持しているデータ $\boldsymbol{x}_i$ を *Paillier* 暗号を用いて暗号化し $Enc(x_1), \ldots, Enc(x_d)$ を取得し，自分の所属する j 番目のクラスタの重心の初期値とし，それ以外のクラスタの重心は $Enc(0)$ を要素とするベクトルとする．

Step 2 *privateAAC* を行い，各ノードは暗号化された各クラスタの平均値 $Enc(\boldsymbol{\mu}_1), \ldots, Enc(\boldsymbol{\mu}_k)$ を取得する．[1)]

Step 3 各ノード u_i は自分が所持しているデータ $\boldsymbol{x}_i$ と各クラスタの重心 $Enc(\boldsymbol{\mu}_1), \cdots, Enc(\boldsymbol{\mu}_k)$ との類似度 Sim を計算する．ノード u_i のクラスタ j への，類似度 $Sim_{i,j}$ の計算は次の式で定める．

$$
\begin{aligned}
Sim_{i,j} &= \prod_{l}^{d} Enc(\mu_{j,l})^{x_{i,j}} && (4.5)\\
&= Enc(\textstyle\sum \mu_{j,l} x_{j,l}) && (4.6)\\
&= Enc(\boldsymbol{\mu}_j \cdot \boldsymbol{x}_i) && (4.7)
\end{aligned}
$$

Step 4 ノードは k 個の類似度の中からランダムに $Sim_{i,A}, Sim_{i,B}$ の二つを選びとし，$(Sim_{i,A}/Sim_{i,B})^n$ を計算してサーバ S に送信．

Step 5 サーバは送信された値を復号し，正の値か負値かを判別し，正の場合は 1, 負の場合は 0 の値をユーザに返す．

Step 6 ノードはサーバから受け取った符号を見ることによって $Sim(A)$ と $Sim(B)$ のどちらが大きいかを判別することができる．ユーザは再び大きいと判定されたものを $Sim(A)$

[1)] ただし，今回は privateAAC の最後の段階で行う 2^{T+1} の除算は行わない．

とし，新しい $Sim(B)$ を選び，再びサーバに送信する．この *Step4-6* を このとき，ノードは A と B のクラスタ番号を記憶しておく．トーナメント形式で，最も大きい一つが残るまで $k-1$ 回繰り返すことで，最も類似度の高いクラスタを探索する．

Step 7 ノード u_i の所属を最も類似度の高いクラスタに変更する．

Step 8 *Step1-4* を T 回繰り返し，最終的に所属しているクラスタの番号を出力する．T は *k-means* を繰り返し実行する回数である．

4.5 評価

4.5.1 Newscast の性能

newscast プロトコルで算出した平均 μ^* は，真の平均 μ の近似値である．この二つの値の誤差 $\Delta\mu = |\mu - \mu^*|$ があるのかを調査した．newscast プロトコルは約 40 サイクル行うと収束するといわれている [48]．実際にノード数によってどのくらいで真の平均へ収束するのかを調査したグラフを図 4.3 に示す．

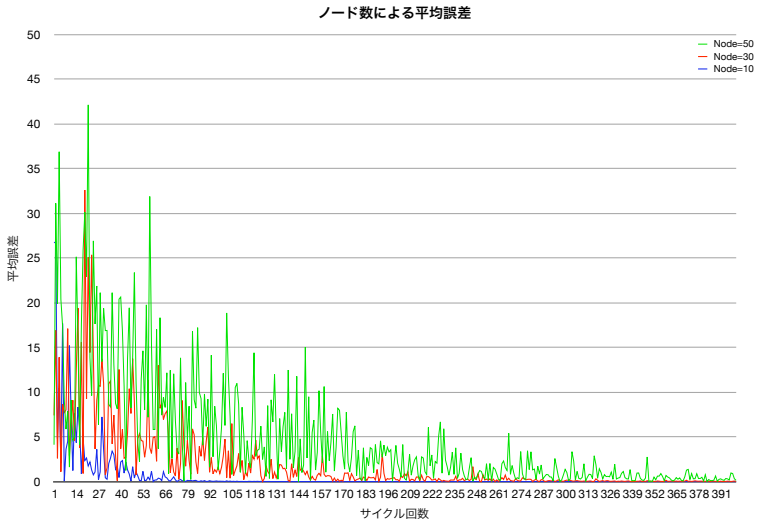

図 4.3: サイクル数による真の平均に対する誤差

newscast による平均値の分布をを図 4.5 に示す．この図は $n = 100$ で，データ x の値に 100 以下の整数を割り当てた時のものである．図に示す通り, サイクル数が高い方が平均値に収束していることがわかる．

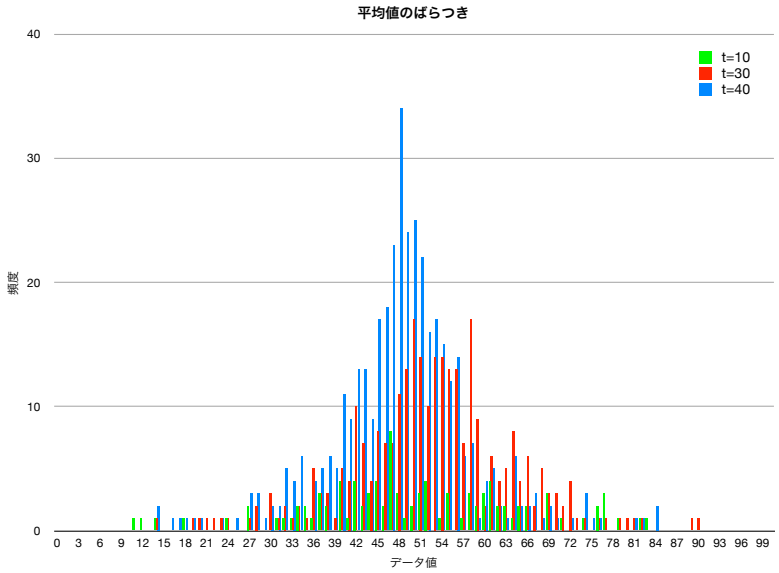

図 4.4: サイクル回数による平均値のばらつき

newscast はランダムに相手ノードを選択するため，必ずしも自ノードと相手ノードのサイクル回数が一致するとは言えない．ノード数 n で newscast を実行した時の二者間のサイクル数の誤差の分布を図 4.5 に示す．

4.5.2 コサイン類似度の精度

k-means クラスタリングを行う際に，各ノードと平均を計算する方法をユークリッド距離とコサイン類似度でそれぞれ計算した場合の精度を比較する．ユークリッド距離 $d(\boldsymbol{x}, \boldsymbol{\mu})$ と，コサイン類似度 $\cos\theta$ はそれぞれ次の式で計算される．

$$d(\boldsymbol{x}, \boldsymbol{\mu}) = \sqrt{\sum_{i=1}^{n}(x_i - \mu_i)^2} \tag{4.8}$$

$$c(\boldsymbol{x}, \boldsymbol{\mu}) = \frac{\boldsymbol{x} \times \boldsymbol{\mu}}{\|\boldsymbol{x}\| \cdot \|\boldsymbol{\mu}\|} \tag{4.9}$$

精度 E_t を次のように定義する．

$$E_t = \sum_{j}^{k} |G_{j*} - G_{jt}| \tag{4.10}$$

k-means クラスタリングにおいて，ユークリッド距離を用いて k-means クラスタリングを行った結果 G_{j*} を真値とし，コサイン類似度を用いて k-means クラスタリングを行った

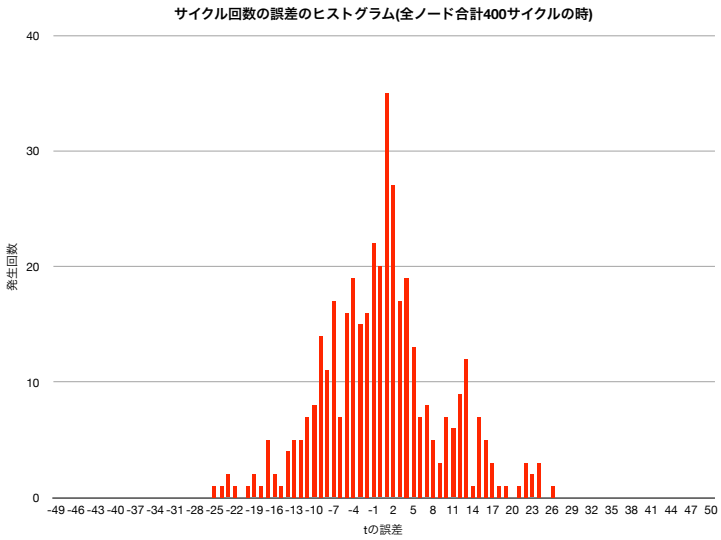

図 4.5: 誤差
の分布

結果 $G_{j,g}$ を比較し，異なったクラスタへ識別されたノードの個数を誤差とする．ユークリッド距離とコサイン類似度をそれぞれ利用した時のクラスタリング結果を図 4.6 に示す．

4.6 Web 履歴に基づくユーザクラスタリング

相羽らによる検索履歴のプライバシーを秘匿したユーザクラスタリング [51] で用いたデータを利用し，k-means クラスタリングに適用した．このデータは被験者 6 名の検索履歴を一人 15 件，合計 90 件取得し，Yahoo!!Japan の分類をもとに検索履歴を分類分けしたものを利用する．分類分けは図 4.1 の通りである．

$d = 15$,$n = 6$,$k = 3$ クラスタリングを行う前と実行後の状態を表 4.2, 表 4.3 に示す．

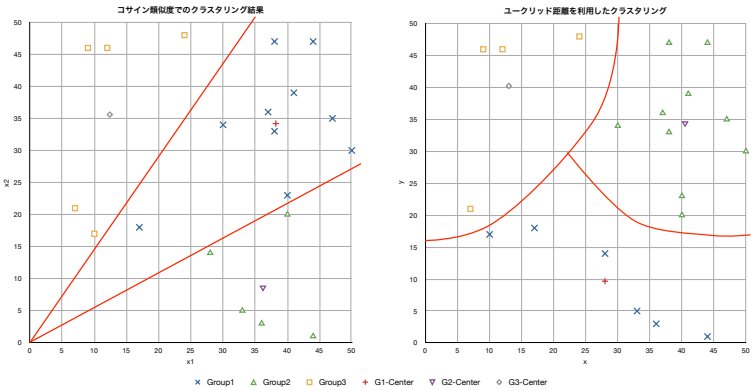

図 4.6: 左：ユークリッド距離を用いた場合，右：コサイン類似度を用いた場合

表 4.1: 各番号に対応するジャンル

番号	ジャンル名
1	エンターテイメント
2	メディアとニュース
3	趣味とスポーツ
4	ビジネスと経済
5	各種資料と情報源
6	生活と文化
7	芸術と人文
8	コンピュータとインターネット
9	健康と医学
10	教育
11	政治
12	自然か科学と技術

表 4.2: k-means を実行する前のデータとクラスタ

User	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	C1	C2	C3
A	20	0	38	3	4	13	2	7	2	2	1	2	0	1	5	0	1	0
B	8	2	50	0	7	4	0	5	10	1	0	2	3	0	8	1	0	0
C	30	6	21	13	7	1	0	20	0	1	0	0	1	0	0	1	0	0
D	0	1	2	16	23	4	38	7	8	0	0	0	0	1	0	0	1	0
E	9	19	26	7	13	1	0	4	0	0	14	3	0	4	0	0	0	1
F	8	7	18	0	20	1	6	10	11	0	4	3	2	0	10	0	0	1

表 4.3: k-means を実行後のデータとクラスタ

User	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	C1	C2	C3
A	20	0	38	3	4	13	2	7	2	2	1	2	0	1	5	1	0	0
B	8	2	50	0	7	4	0	5	10	1	0	2	3	0	8	1	0	0
C	30	6	21	13	7	1	0	20	0	1	0	0	1	0	0	1	0	0
D	0	1	2	16	23	4	38	7	8	0	0	0	0	1	0	0	1	0
E	9	19	26	7	13	1	0	4	0	0	14	3	0	4	0	0	0	1
F	8	7	18	0	20	1	6	10	11	0	4	3	2	0	10	0	0	1

第5章
実装評価

5.1 実験環境

PC(CPU:Core 2 Duo 2.26GHz,Memory:4GB,
Java version 1.6) で測定した暗号処理時間を表 5.1 に示す．[1)]

表 5.1: 試験実装仕様

仕様	値
アルゴリズム	1024 bit Paillier 暗号
暗号文長	$\ell_c = 256$ [byte]
暗号化	$t_e = 160$ [ms]
復号化	$t_d = 248$ [ms]
べき乗	$t_m = 0.093$ [ms]
積	$t_m = 0.102$ [ms]

5.2 パフォーマンス

各方式の暗号文の通信量と，計算時間を表 5.2，表 5.3 に示す．送信する暗号文は，事前計算の方式と比較しても変わらないが，k 近傍の方式で行うと $n-1$ から $k-1$ に軽減することができる．計算時間は事前計算の方式で行うことにより，暗号化の処理を 0 回にすることができる．後述するように，暗号文の生成よりも，暗号文同士の乗算の方が高速であるため，事前計算方式は有効と言える．

5.3 安全性

通信路の盗聴からは公開鍵アルゴリズムの安全性に基づいて情報は漏れない．しかし，B が不正な評価値を送ることによる秘匿性のリスクが生じる．例えば，B が A に送信する暗号文を評価値ではなく，$E(0), E(1), E(0)$ のようなマスクを送信したとする．A はプロトコル

[1)] 表 5.1 のべき乗計算の値は一桁の冪乗計算を行った際の計算時間である．

表 5.2: 通信量

方式	送信暗号文	受信暗号文
2. 提案方式	$\ell_c(n-1)$	ℓ_c
3. 事前計算	$\ell_c(n-1)$	ℓ_c
4.k 近傍	$\ell_c(k-1)$	ℓ_c
k 近傍+事前	$\ell_c(k-1)$	ℓ_c

表 5.3: 各計算処理回数

方式	$E(m)$	$E(m_1)\cdot E(m_2)$	$E(m_1)^{m_2}$
2. 提案方式	$n-1$	$n-1$	$n-1$
3. 事前計算	0	$2(n-1)$	$n-1$
4.k 近傍	$k-1$	$k-1$	$k-1$
k 近傍+事前	0	$2(k-1)$	$k-1$

通りに各暗号文にユーザの類似度をべき乗し乗算を行うと，特定のユーザの類似度が B 側に知られてしまう．本論文では，評価値の範囲を 1 から 5 の 5 段階評価と仮定しているが，この問題について検討する必要がある．

5.4 処理速度，通信コスト

基本方式と事前計算方式の処理速度，通信コストの実験を行った．実験に利用したデータは MovieLens[8] のデータセット (ユーザ数 $n = 943$，アイテム数 $m = 1682$，評価値 $100,000$ 件) を利用した．この 100,000 件のデータをアイテム ID について偶数と奇数の 2 つに分割し，それぞれ組織 A，B と設定した．

評価値 $100,000$ 件の散布図を図 5.1，図 5.2 に示す．また，組織 A，B の各分布を図 5.3，5.4 に示す．

ユーザ数 n の変化による予測時間を図 5.5 に示す．基本方式 $n = 900$ でかかった予測時間は $152s$．それに対し，事前計算方式は $4.45s$ と約 34 倍高速化することができた．ユーザ数 n の変化による通信コストを図 5.6 に示す．

5.5 相関係数

組織 A，組織 B それぞれで計算した類似度 s_A,s_B に組織の重み w_A,w_B 付きで総和した値を $\hat{s}$ とする，合計した値を合成の類似度 $\hat{s}$ とし，組織 A，組織 B のデータを合わせて計算し

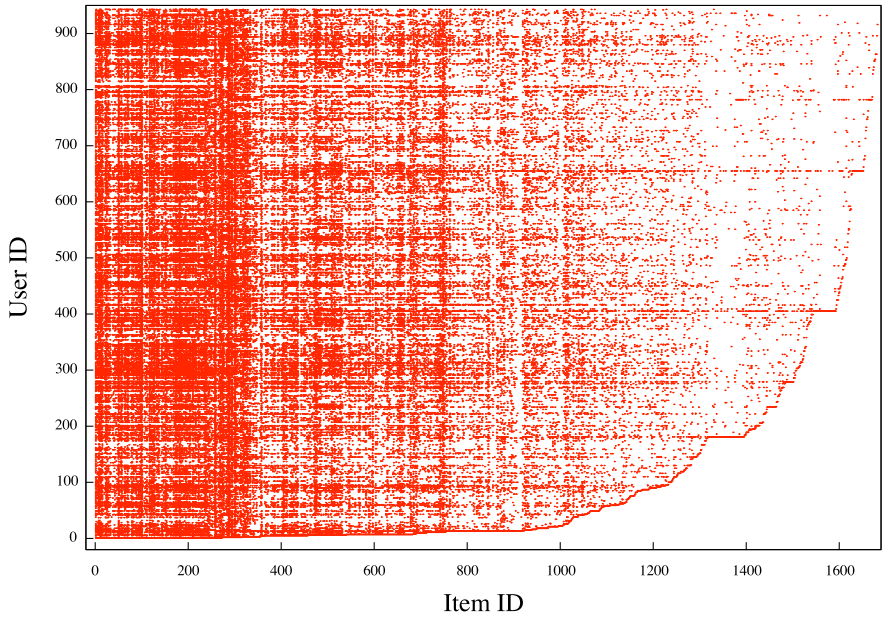

図 5.1: 実験に用いた評価値 100,000 件の散布図

た類似度を真の類似度 s とする．もし，合成の類似度と真の類似度が擬準同型性を満たしているならば，二つの類似度に相関が見られるはずである．

そこで，この二つの類似度の相関を図 5.7 の散布図に示す．相関係数は 0.673 であった．

$$\hat{s} = s_A \cdot w_A + s_B \cdot w_B \tag{5.1}$$

弱い正の相関が見受けられることから，擬準同型性を満たしている類似度といえる．

今回の実験では類似度の計算時に発生する小数点を定数倍して整数に整える演算を行っており，この値を 1000 に設定して計算した．そのため，組織 A 側の小数点 4 桁以降の類似度が切り捨てられてしまっている．これがノイズの原因のひとつではないかと考える．

5.6 精度評価

5.6.1 平均絶対誤差（Mean Absolute Error）

今回の精度評価は平均絶対誤差（Mean Absolute Error：MAE）を採用した．MAE は次の式で計算される．

$$MAE = \frac{\sum_{\ell=1}^{L} |r_\ell - p_\ell|}{L}. \tag{5.2}$$

ここで，p は予測された評価値，L はサンプル数を表す

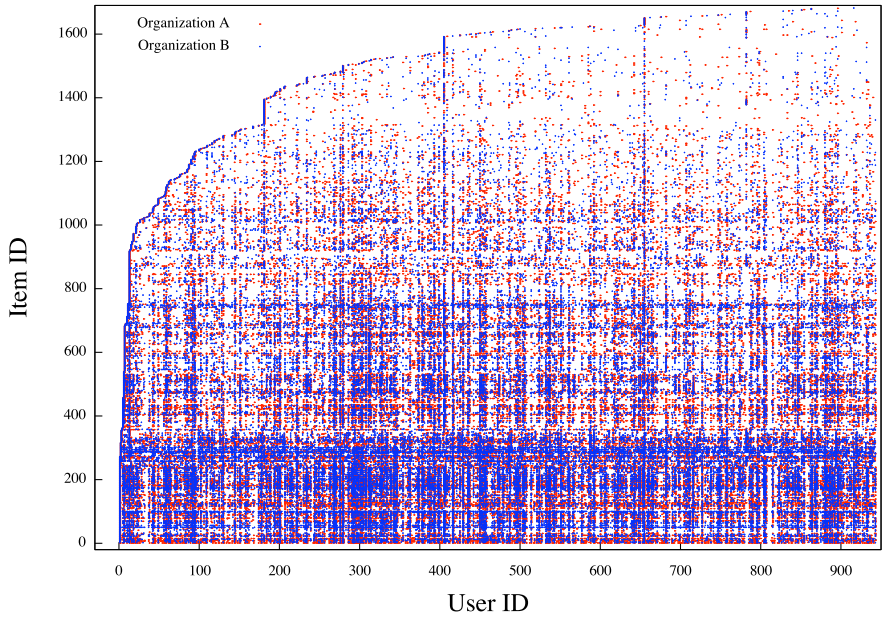

図 5.2: 実験に用いた評価値 100,000 件の組織別散布図

5.6.2 分散環境による MAE の違い

100,000 件のデータのうち 100 件をランダムに選び，次の 4 つのアルゴリズムで評価値の予測を行った．Global NN CF: 結合された DB を使った k 近傍方式，Aggregated NN CF: 分割された DB を使った k 近傍方式を示す．また，結合した DB と分割した DB それぞれのケースについて，k 近傍方式，ランダムに j 人を選ぶ方式を行った．精度を MAE(平均絶対誤差) で評価する．

実験結果を図 5.8 に示す．

結合した DB で k 近傍方式を利用し，ユーザ数 942 で予測したときの MAE が最も低く，0.9595 であった．それに対し，分割した DB で k 近傍の方式は 0.9588 となり，差は 0.007 であった．

5.6.3 類以度による MAE の違い

100,000 件のデータのうち 100 件をランダムに選び，変形ユークリッド距離，コサイン類以度，ピアソン相関係数，Somers' d 類以度の各類以度による，MAE の差を調査した．評価値の計算式は式 (3.6) を使い，ユーザ間のインターセクションが存在しない場合，または類以度を計算する時に分母が 0 になってしまった場合は，類以度を 0 とした．

また，コサイン類以度，ピアソン相関係数，Somers' d 類以度は -1 から 1 までの値を取るので，0 から 1 の値をとるように正規化した．

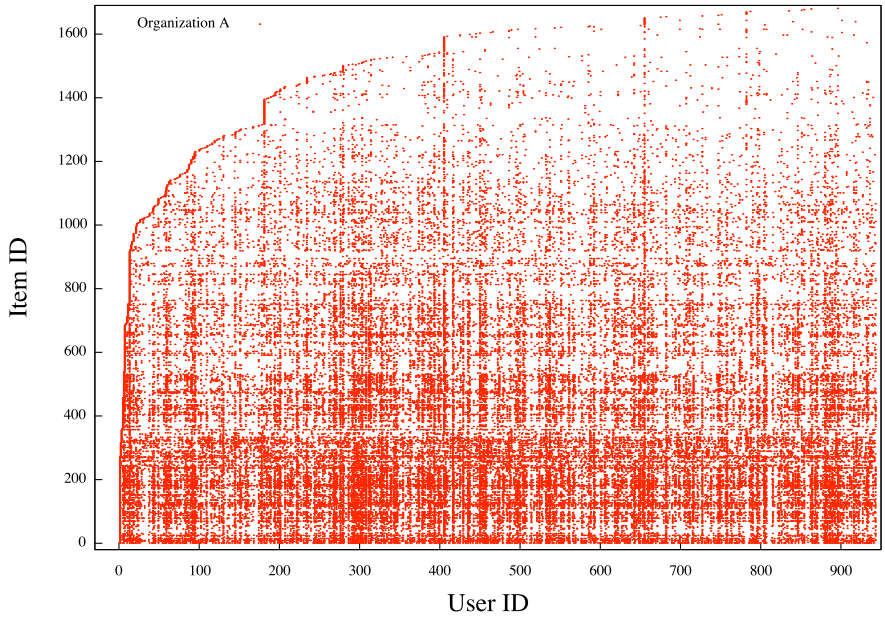

図 5.3: 組織 A の持つ評価値の散布図

実験結果を図 5.9, 表 5.4，図 5.10, 表 5.5 に示す．図 5.9 と図 5.10 の違いは，別の 100 件のサンプルに付いて行った結果である． ユーザ数 942 の時，コサイン類以度とピアソン相関係数，Somers' d 類以度の差が 0.01 ほどで，あまり差は見られなかったが，変形ユークリッド距離より精度が 0.1 向上している．

別の 100 件のサンプルで再実験を行ったところ，ユーザ数 942 の時，コサイン類以度の精度が低下していた．コサイン類以度はベクトルの向きでしか比較しないため，サンプルによって誤差が発生してしまうためであると考える．例えば，全て評価に 1 をつけたユーザ $\boldsymbol{x} = (1, 1, 1, 1, 1)$ と，全て評価に 5 をつけたユーザ $\boldsymbol{y} = (5, 5, 5, 5, 5)$ の類以度は 1 となってしまう．

5.6.4 平均を考慮した CF の式による違い

予測評価値を計算するときに，計算を単純化するため式（3.6）を用いたが，ユーザの平均を考慮した式（3.4）を適用した場合どのように変化するのかを調査した．ここで，用いる DB は分散さていない評価値 100, 000 件の DB を利用した（図 5.1）．

ユーザ間のインターセクションが存在しない場合，または類以度を計算する時に分母が 0 になってしまった場合は，類以度を 0 とした．

また，コサイン類以度，ピアソン相関係数，Somers' d 類以度は -1 から 1 までの値を取るので，同様に 0 から 1 の値をとるように正規化した．実験結果を図, 表 一番近傍ユーザが少ない $k = 200$ や $k = 300$ の時が，最も MAE が低い．これは k がおおきくなるにつれて，

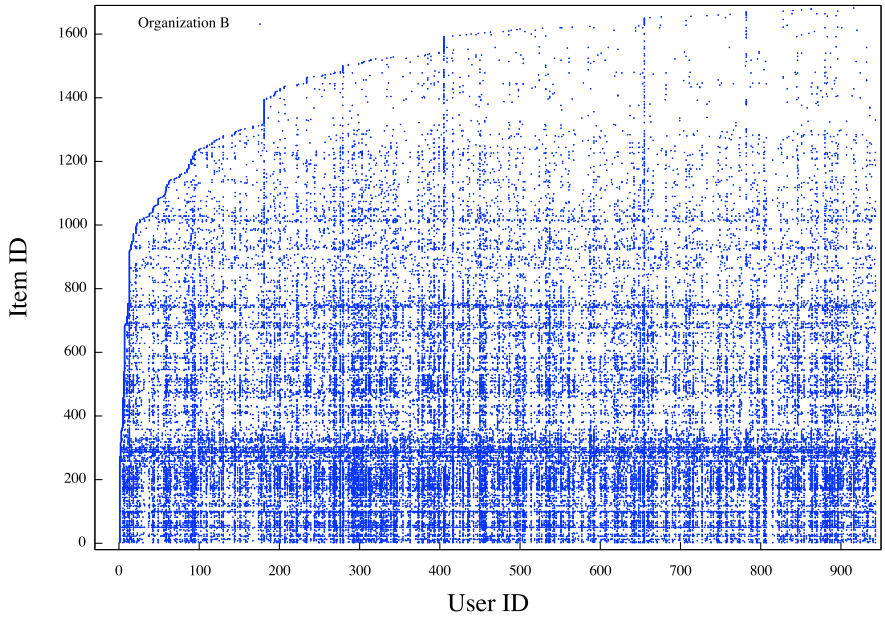

図 5.4: 組織 B の持つ評価値の散布図

ノイズが増え，精度が低下するのではないかと考えられる．最も低い MAE は，ピアソン相関係数の $k = 300$ のときに 0.7569 を示している．また，Somers' d 類以度は $k = 200$ のときに 0.759352 を示している．これらの結果から，平均値によって評価値を正規化することによってノイズが減り，精度が向上することが言える．

5.6.5 類以度と分割方法による MAE の差

極端な例として，図 5.12 に示すような環境で実験を行った．これは組織 A が 92,307 件の評価値を所持しており，組織 B が 3,541 件の評価値を所持している例である．また，組織 A に存在するユーザが組織 B に存在するが，一切評価値を所持していないときに，提案方式で推薦精度がどのようになるのかを実験した．

類以度と分割方法による MAE の差を表 5.7 に示す．

ピアソン相関係数と Somers' d 類以度は，局所 DB 分散 DB 全 DB となっており，分散した環境でも全域の DB を使った時の値に近似できている．そのため，ピアソン相関係数と Somers' d 類以度は分散計算に向いているといえる．

逆に，変形ユークリッド距離と，コサイン類以度は分散環境での計算に向いていないと言える．

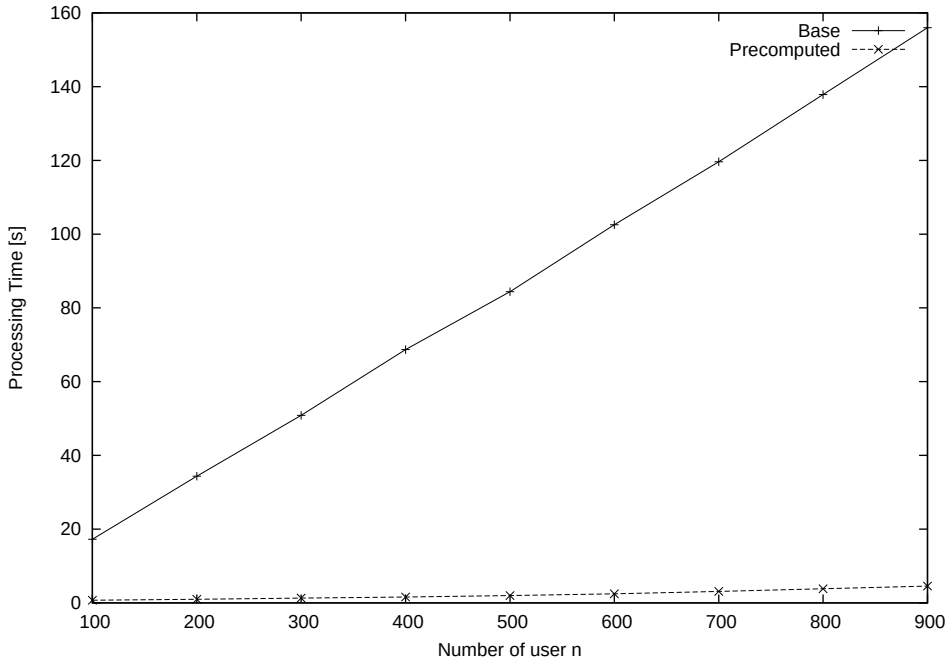

図 5.5: ユーザ数 n に対する計算処理速度（基本方式と事前計算）

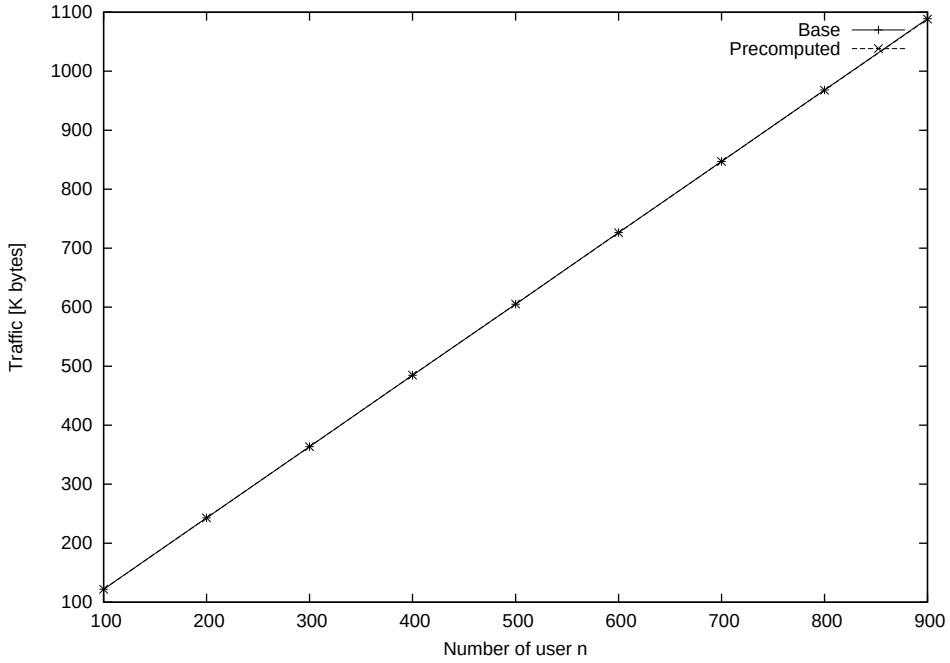

図 5.6: ユーザ数 n に対する通信コスト（基本方式と事前計算）

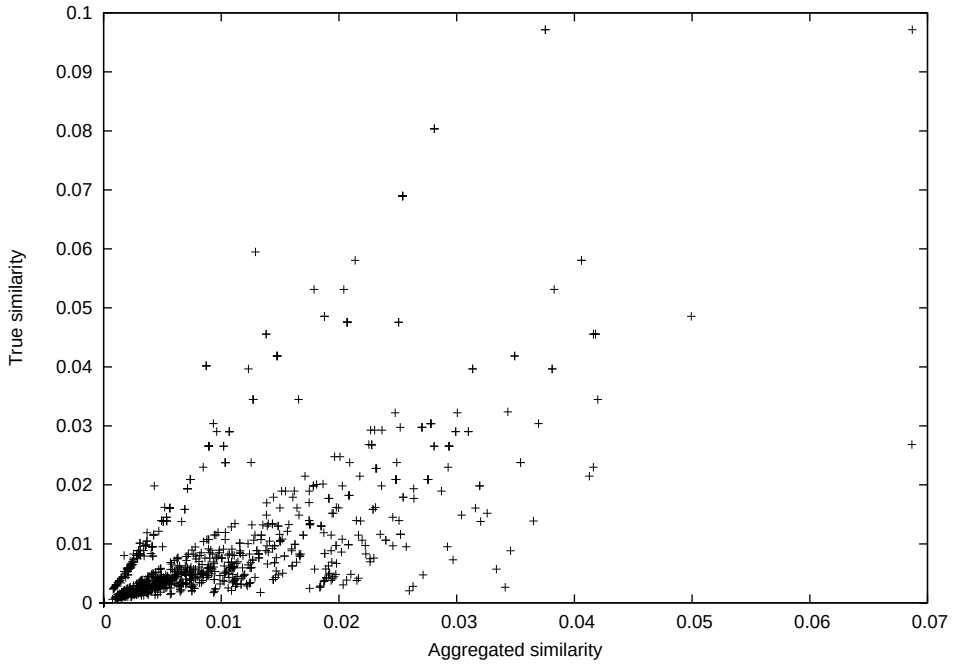

図 5.7: DB 結合時の類似度と DB 分割時の類似度の比較

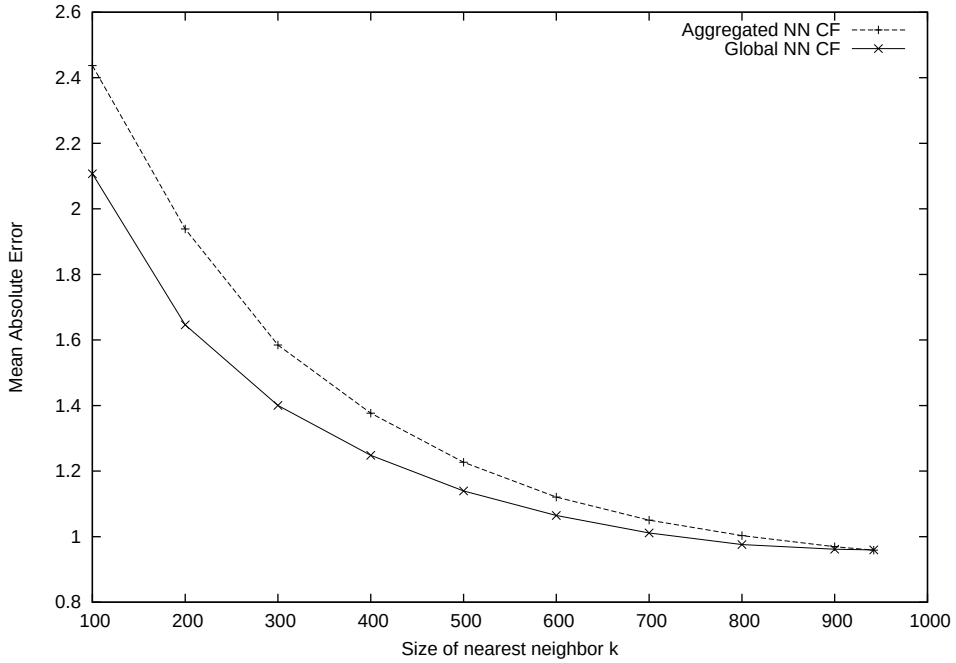

図 5.8: DB 結合時と分割時における k 近傍の精度評価

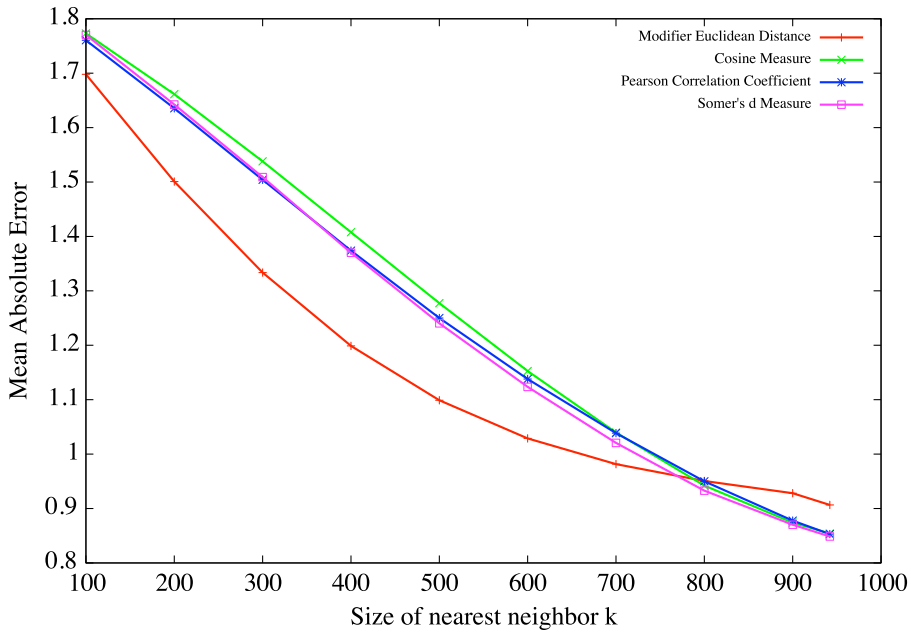

図 5.9: 類以度による MAE の差 1

表 5.4: 類以度による MAE の差 1

Number of user	Mod. Euc.	Cosine	Pearson	Somers' d
100	1.697931	1.773020	1.760508	1.770307
200	1.500814	1.661443	1.635433	1.642459
300	1.333540	1.538014	1.504127	1.508977
400	1.198898	1.407661	1.373838	1.370066
500	1.098948	1.277392	1.250035	1.240352
600	1.028873	1.152707	1.138010	1.123541
700	0.981540	1.039622	1.038514	1.020607
800	0.950567	0.942148	0.949977	0.932831
900	0.928167	0.874159	0.877762	0.870245
942	0.906570	0.854164	0.852960	0.848620

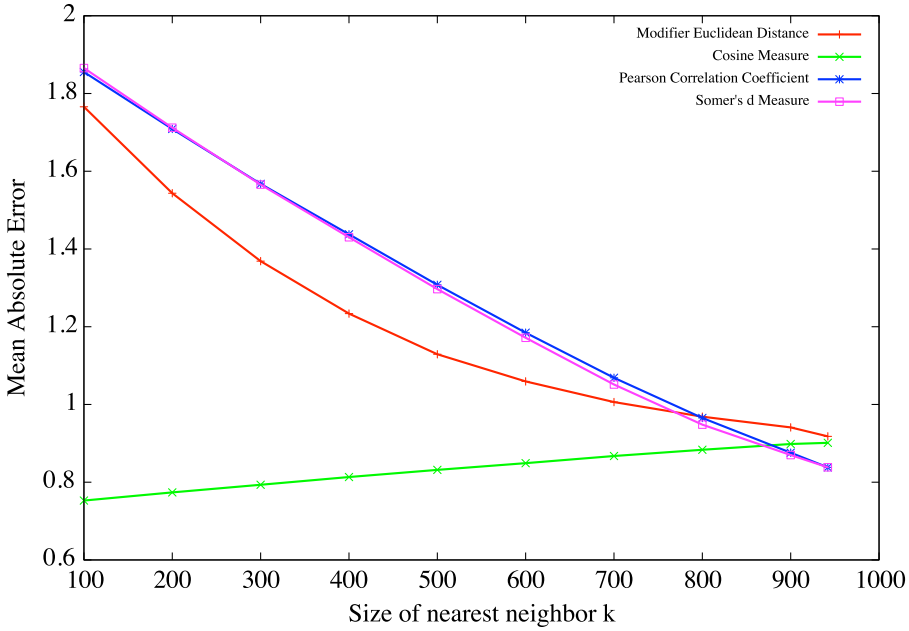

図 5.10: 類以度による MAE の差 2

表 5.5: 類以度による MAE の差 2

Number of user	Mod. Euc.	Cosine	Pearson	Somers' d
100	1.765865	0.752738	1.855328	1.865414
200	1.543438	0.773760	1.708903	1.711960
300	1.368673	0.793391	1.567504	1.565661
400	1.233753	0.813342	1.437771	1.430185
500	1.129175	0.831777	1.307699	1.296803
600	1.059245	0.849020	1.184673	1.171647
700	1.006133	0.867429	1.068605	1.051311
800	0.968622	0.883430	0.965273	0.948111
900	0.940968	0.898279	0.875911	0.869908
942	0.917980	0.901318	0.838112	0.838363

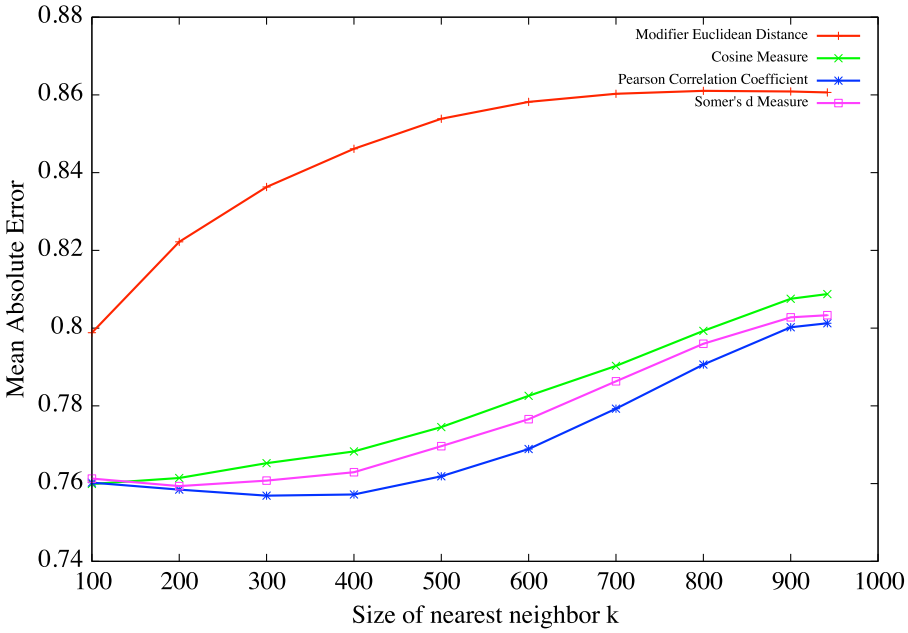

図 5.11: 結合した DB し平均を考慮した式による差

表 5.6: 結合した DB し平均を考慮した式による差

Number of user	Mod. Euc.	Cosine	Pearson	Somers' d
100	0.798816	0.759910	0.760253	0.761292
200	0.822198	0.761447	0.758452	0.759352
300	0.836312	0.765264	0.756900	0.760767
400	0.846130	0.768281	0.757202	0.762919
500	0.853870	0.774536	0.761877	0.769617
600	0.858201	0.782580	0.768892	0.776572
700	0.860290	0.790287	0.779269	0.786308
800	0.861043	0.799293	0.790629	0.795984
900	0.860881	0.807553	0.800213	0.802779
942	0.860635	0.808753	0.801250	0.803322

表 5.7: 類以度と分割方法による MAE の差

	Mod. Euc.	Cosine	Pearson	Somer's d
全 DB	1.325	1.298	0.922	1.024
分散 DB	1.517	1.490	1.086	1.226
局所 DB	1.292	1.292	1.559	1.292

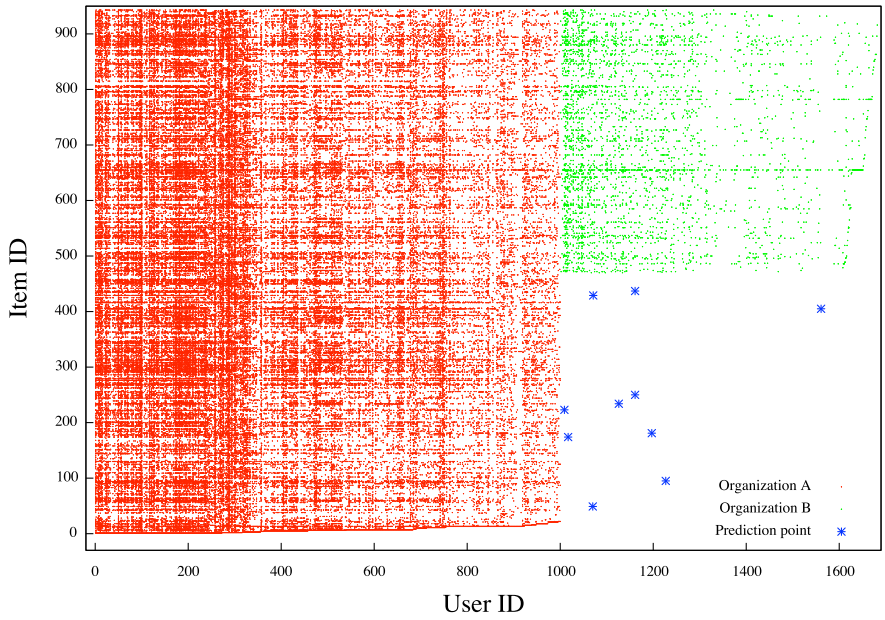

図 5.12: 極端な分割 DB の例

第6章
結論と今後の課題

6.1 結論

擬準同型性を満たした類似度を導入することにより，局所的に評価した類似度から全域的な類似度を合成できることを示し，その性質を情報推薦システムに適用した．

また，変形ユークリッド距離，コサイン類以度，ピアソン相関係数，Somers' d 類以度を本システムに適応し，精度を検証した．その結果，ピアソン相関係数と，Somers' d 類以度が分散計算に向いていることが分かった．

さらに，各組織の持つデータベースの大きさや，スパース率の違いによる実験も行った．その結果，本提案手法を導入することによって，予測評価値を二組織間の DB を結合したときの値に近似することができ，本来予測が出来ない値も予測可能になった．

しかし，今回示した分散環境のモデルは一例であり，様々なタイプの DB を用いて検証する必要がある．

6.2 課題

今後の課題として，最適な組織や類似度への重み付けや，完全準同型性を満たした類以度の発掘などが挙げられる．

6.2.1 組織への重み付け

今回利用した重みは，単純に組織が管理しているアイテムの個数としたが，この重みに，スパース率を用いる等他の手法も検討する必要がある．また，完全準同型性類以度を見つけることが出来れば，この重みを利用する必要がなくなると考えられる．

6.2.2 完全準同型性類以度

本論文で紹介した類以度はどれも完全には準同型性を満たさない．しかし，Somers' d 類以度は，第]4 章で紹介したような秘匿したまま平均を計算するプロトコルを用いることで，実現できるのではないかと考えられる．

また，単純に二つの集合の和集合と，積集合を類以度とした Jaccerd 係数を，暗号用い秘匿したまま計算することが出来れば，実現できるのではないかと考えられる．

参 考 文 献

[1] Neal Lathia, Stephen Hailes, Licia Capra, “Private Distributed Collaborative Filtering Using Estimated Concordance Measures”, In ACM 2007 Conference on Recommender Systems (RecSys). Minneapolis, Minnesota, USA. October 19-20, 2007.

[2] A. Agresti, “Analysis of Ordinal Categorical Data”, , 1984.

[3] John S. Breese, David Heckerman Carl Kadie, “Empirical Analysis of Predictive Algorithms for Collaborative Filtering”, The 14th conference on Uncertainty in Artificial Intelligence, 1998.

[4] 平山 巧馬, 小柳 滋, “協調フィルタリングにおける相関係数法の予測性能向上”, 電子情報通信学会論文誌. D, 情報・システム, Vol J90-D(2), 223-232, 2007.

[5] 木澤, 菊池, “プライバシ協調フィルタリングにおける利用者評価行列の次元削減”, コンピュータセキュリティシンポジウム 2008(CSS2008), pp. 509–514, 2008.

[6] 木澤, 磯崎, 菊池, “秘匿積集合プロトコルを利用したプライバシ協調フィルタリングの提案”, 2009 年暗号と情報セキュリティシンポジウム (SCIS2009), 2009.

[7] 多田, 菊池, “秘密分散ベースの秘匿関数計算を用いたプライバシ保護情報推薦方式”, 2011 年暗号と情報セキュリティシンポジウム (SCIS2011), 2011.

[8] Grouplens Data Sets, (`http://grouplens.org/`), 2006.

[9] Netflix prize, (`http://www.netflixprize.com/`), 2007.

[10] ★ Amazon.co.jp, (`http://amazon.co.jp`)

[11] J.Canny, “Collaborative Filltering with Privacy”, IEEE Conf. on Security and Privacy, pp. 45–47, Oakland CA, 2002.

[12] 神嶌, “推薦システムのアルゴリズム (1)”, 人工知能学会誌, Vol. 22 No. 6, pp. 826-837, 2007.

[13] 神嶌, “推薦システムのアルゴリズム (2)”, 人工知能学会誌, Vol. 23 No. 1, pp. 89-103, 2007.

[14] 神嶌, “推薦システムのアルゴリズム (3)”, 人工知能学会誌, Vol. 23 No. 2, pp. 248-263, 2008.

[15] 高島 秀佳, 山岸 英貴, 平澤 茂一, “欠損値推定による協調フィルタリング手法” 情報科学技術フォーラム一般講演論文集, Vol. 4, No. 1, pp. 15-16, 2005. 人工知能学会誌, Vol. 23 No. 2, pp. 248-263, 2008.

[16] Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, John Riedl, “An algorithmic framework for performing collaborative filtering”, SIGIR ’99 Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, 1999.

[17] Jaideep Vaidya and Chris Clifton, “Privacy preserving naive bayes classfier for vertically partitioned data”, In 2004 SIAM International Conference on Data mining, pp. 522-526, 2004.

[18] Jaideep Vaidya and Chris Clifton, “Secure set intersection cardinality with application to association rule mining”, Journal of Computer Security, Vol. 13, No. 4, pp. 593-622, 2005.

[19] R. Agrawal and R. Srikant, “Privacy-Preserving Data Mining”, ACM SIGMOD 2000, pp. 439-450, 2000.

[20] Z. Huang, W. Du and B. Chen, “Deriving Private Information from Randomized Data”, ACM SIGMOD 2005, pp. 37-48, 2005.

[21] H. Polat and W. Du, “Privacy-Preserving Collaborative Filtering using Randomized Perturbation Techniques”, ICDM 2003, pp. 1-15, 2003.

[22] 望月 安奈, 菊池 浩明, “摂動化によってプライバシーを保護した情報推薦方式”, Multimedia,Distributed, Cooperative, and Movile Symposium(DICOMO), 3G-プライバシ保護, pp. 536-541, 2011.

[23] R.L.Rivest,A.Shamir,and L.Adelman, “A Method for Obtaining Digital Signature and Public-key Cryptsystems”, Communications of the ACM 21,2, pp. 120-126, 1978.

[24] T. ElGamal, “A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms”, IEEE Trans. on Information Theory, IT-31(4), pp.469-472, 1985.

[25] P. Paillier, “Public-Key Cryptosystems Based on Composite Degree Residuosity Classes”, Proc. *EUROCRYPT’99*, LNCS 1592, pp. 223-238, 1999.

[26] Haifeng Yu, Chenwei Shi, Kaminsky, M., Gibbons, P.B., and Feng Xiao, "DSybil: Optimal Sybil-Resistance for Recommendation Systems", in IEEE Symp. on Security and Privacy, pp. 283-298, IEEE, 2009.

[27] Wenliang Du and Mikhail J. Atallah, "Privacypreserving statistical analysis", In Proceeding of the 17th Annual Computer Security Applications Conference, pp. 10-14 2001.

[28] Yehuda Lindell and Benny Pinkas, "Privacy preserving data mining", Journal of Cryptology, 15(3), pp. 177-206, 2002.

[29] Rakesh Agrawal, Alexandre Evfimievski, and Ramakrishnan Srikant, "Information shareing across private databases", in proc. of ACM SIGMOD International Conference on Management of Data, 2003.

[30] Michael J. Feedman, Kobbi Nissim, and Benny Pinkas, "Efficient private matching and set intersection", in Eurocypt 2004, IACR, 2004.

[31] G. Jagannathan and R. N. Wright, "Privacy-Preserving Distributed k-Means Clustering over Arbitrarily Partitioned Data", *ACM KDD' 05*, 2005.

[32] Andrew C. Yao, "How to generate and exchange secrets", In Proc. of the 27th IEEE Symposium on Foundations of COmputer Science, pp. 162-167, 1986.

[33] Dahlia Malkhi, Noam Nisan, Benny Pinkas, and Yaron Sella, "Fairplay – A Secure Two-Party Computation System", Usenix Security Symposium, 2004.

[34] Koji Chida, Dai Ikarashi, and Katsumi Takahashi, "Tag-Based Secure Set-Intersection Protocol and Its Application", in proc. of Computer Security Symposium (CSS 2009), IPSJ, 2009 (in Japanese).

[35] L. F. Cranor, "I Didn't Buy it for Myself, Privacy and E-Commerce Personalization", WPES 2003, Washington, DC, USA, pages 111-117, 2003.

[36] J. Canny: Collaborative Filtering with Privacy, *IEEE Conf. on Security and Privacy*, Oakland CA, May 2002.

[37] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering", In UAI, pp. 43-52, 2004.

[38] Resnick, P., Iacovou, N., Sushak, M., Bergstrom, P., and Riedl, J. , "GroupLens: An open architecture for collaborative filtering of netnews", Proceedings of the 1994 Computer Supported Collaborative Work Conference.

[39] G. Morohash, et.al, “Secure Multiparty Computation for Comparator Networks”, IEICE Trans. Fundamentals, Vol. E91-A, No. 9,2008.

[40] Katzenbeisser, S. and Petkovic, “Privacy-Preserving Recommendation Systems for Consumer Healthcare Services”, In Proceedings of the 2008 Third international Conference on Availability, Reliability and Security (ARES 2008), IEEE Computer Society, pp. 889-895, 2008.

[41] Ahmad, W. and Khokhar, “An Architecture for Privacy Preserving Collaborative Filtering on Web Portals”, In Proceedings of the Third international Symposium on information Assurance and Security, IEEE Computer Society, pp. 273-278, 2007.

[42] J. S. Breese, D. Heckrman, and C. Kadie, “Empirical analysis of predictive algorithms for collaborative filtering,” In UAI, pp.43-52, 2004.

[43] H. Kikuchi, H. Kizawa and M. Tada, “Privacy-Preserving Collaborative Filtering Schemes”, WAIS 2009, ARES 2009 federated workshop, IEEE Press, 2009.

[44] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl “Item-Based Collaborative Filtering Recommendation Algorithms,” ACM WWW10, Hong Kong, May 2001.

[45] 総務省, “クラウドコンピューティング時代のデータセンター活性化策に関する検討会”, 2010 年 5 月 28.

[46] 株式会社ドリーム・トレイン・インターネット, OpenBit.Net, “インターネット接続サービス利用規約”, 2011 年 4 月 1 日.

[47] G. Jagannathan, K. Pillaipakkamnatt, R. N. Wright, D. Umano, “Communication-Efficient Privacy-Preserving Clustering”, Transaction on Data Privacy, pp. 1-25, vol. 3, 2010.

[48] W. Kowalczyk and N. Vlassis, “Newscast EM”, Advances in Neural Information Processing Systems 17, MIT Press, 2005.

[49] 佐久間 淳, 小林 重信, “P2P ネットワークにおけるプライバシを保護した非同期平均計算プロトコル”, pp. 1-6, SCIS2007 3D4-1.

[50] 佐久間 淳, 小林 重信, “P2P ネットワークにおけるプライバシを保護した k-means クラスタリング”, pp. 1-6, SCIS2007 3D4-2.

[51] 相羽 研次, “検索履歴のプライバシーを秘匿した ユーザクラスタリング”, 東海大学情報理工学部情報メディア学科 2009 年度卒業論文, 2009.

謝辞

本論文を執筆するにあたり多くの方々から多大なる御指導と御援助を賜りました.

特に，研究に関わらず私を導いて下さった東海大学情報理工学部情報メディア学科菊池 浩明 教授に深く感謝を申し上げます.

また，本研究を推進するにあたって，御親切なる御教示ならびに御激励を賜りました東海大学情報理工学部情報科学科 中西 祥八郎 教授，東海大学情報理工学部情報科学科 内田 理 准教授，株式会社 NTT ドコモ 先進技術研究所 寺田 雅之 氏，株式会社 NTT ドコモ 先進技術研究所 石井 一彦 氏，株式会社 NTT ドコモ 先進技術研究所 関野 公彦 氏，筑波大学大学院 佐久間 淳 准教授，産業技術総合研究所 神嶌 敏弘 氏に厚く御礼申し上げます.

さらに，2 年間共に楽しみ，苦しみ，励まし合い，時には研究に対して有益な意見を与えてくれた東海大学大学院工学研究科情報理工学専攻の皆様，先生がたに感謝致します.

また，本研究にあたり，実データを提供していただいた相羽研次氏，実験に協力してくださった方々に感謝の意を述べると共に，謝辞とさせていただきます.

最後に，家族に心より感謝致します.