推薦システム

Recommender System

神嶌 敏弘

推薦システム登場の背景

社会の高度情報化 & 情報発信の低コスト化

大量のデータが常に生成されている

記憶媒体の大容量化 & 通信の高速化

膨大なデータの蓄積や流通が可能になった

↓

情報過多

Information Overload

情報洪水 (Information Overflow) ／ 情報爆発 (Information Explosion)

情報過多

[Resnick+ 94, Maes 94]

情報過多
Information Overload

欲しい情報が
埋もれている

検索しても候補がありすぎる

必要な情報を
具体化できない

適切なキーワードがわからない
官庁統計の名前が不明

情報があると分かっているのに
欲しい情報は見つけられない

推薦システムの登場

どうにかして情報を利用できないか？

他の人の意見を参考にする　　好きなものの特徴に注目

推薦システム

Recommender System

この作業をコンピュータを使って行う

推薦システム

情報過多

膨大な情報の集積

↓

情報があると分かっているのに，欲しい情報は見つけられない

欲しい情報が埋もれている or 必要な情報を具体化できない

推薦システム

利用者が必要としていると思われる情報を選び出す

内容ベース フィルタリング Content-Based Filtering	協調 フィルタリング Collaborative Filtering
アイテムの特徴を利用	他人の意見を利用

推薦システムの例と分類

推薦システム (Amazon.co.jp)

特定の顧客の嗜好に
合わせて商品を選ぶ

2007/7/26 にスクリーンショットを取得

推薦システム (Amazon.co.jp)

売れ筋/評価情報

他の利用者の評価

類似商品の推薦

2007/8/5 にスクリーンショットを取得

推薦システム (iTunes)

人気リスト

店舗の専門家(editor)の推薦

2007/7/26 にスクリーンショットを取得

推薦システム (iTunes)

他の利用者による
推薦リスト

2007/7/26 にスクリーンショットを取得

推薦システム (MovieLens)

映画を推薦する実験システム

http://www.movielens.org/

2007/7/26 にスクリーンショットを取得

推薦システム (MovieLens)

利用者の嗜好の
度合いを
★で示す

2007/7/26 にスクリーンショットを取得

キャッチコピー

[BenShafer+ 01]

If I have 3 million customers on the Web,
I should have 3 million stores on the Web

Jeff Bezos, CEO of Amazon.com

Web上の300万の顧客には，Web上の300万の店を

"the old World of Mass Production, in which standardized products, homogeneous markets, and long product life and development cycles ware the rule" ➡
"variety and customization supplant standardized products, heterogeneous and fragmented markets spring from once-homogeneous markets, and product life cycles and development cycles spiral downward"

B. Joseph Pine II, "Mass Customization"

画一的なMass Productionから
多様なMass Customizationへ

推薦システムの要素技術

ヒューマンインターフェース

- アイテムの内容や推薦に関連した情報を提示
- 推薦に必要な情報を，利用者から的確に収集

機械学習・統計的予測・情報検索

- 収集したデータに基づく推薦情報の生成
- 目的に応じた推薦情報の変換

データベース・並列計算・ネットワーク

- 推薦に必要な情報を蓄積・処理・流通させる基盤

他の技術との関連

[BenShafer+ 01]

情報フィルタリング (information filtering)

- 逐次的に入力される情報から，利用者プロファイルに適合するものを選別する技術
- 内容ベースの推薦システムと類似
  - 利用者プロファイルがクエリで表されることが多い
  - フィルタリングは，必要なものを見つけるより，不要なものの除外が主な目的

マーケティングの関連技術

- マーケティングは供給側 ⟷ 推薦システムは消費者側
- 推薦システムには，全体傾向のレポート作成などは不要
- マーケティングのように顧客層を分類することは，推薦システムの目的ではない

推薦システムの運用目的による分類

[BenShafer+ 01]

概要推薦 (broad recommendation)

- 統計情報や編集者による推薦．新規や低頻度の利用者の補助.

利用者評価 (user comments and rating)

- 他の利用者の評価を示す．コミュニティの構築.

通知サービス (notification service)

- システムを利用していないときにメールなどで通知．サイトへの再訪を促す.

関連アイテム推薦 (item-associated recommendation)

- 関連アイテムやその情報を同時に提示．cross-sellingの促進や比較判断の補助.

緊密な個人化 (deep personalization)

- 積極的に利用者の個人情報や行動履歴を利用．差別化によるシステムへのロイヤリティの構築.

推薦の個人化の度合い

[BenShafer+ 01]

- **非個人化 (non-personalized)**
  人気商品ランキング，人間の推薦者による推薦などどの利用者にも一律に行われる推薦
- **一時的個人化 (ephemeral personalization)**
  利用者の入力に基づいて推薦をする．カテゴリ検索や，類似商品の提示など
- **永続的個人化 (persistent personalization)**
  利用者の過去の行動などに基づき，利用者の同じ入力・行動に対しても異なる結果が表示される

コストと利得のバランス

[Herlocker+ 04]

推薦を利用者が受け入れたときのコストと利得

- **コスト**：利用者が支払う代金や消費する時間など
- **利得**：推薦が適切/不適切な場合に利用者が得る満足

映画などのエンターテイメント

コスト：	チケット代 + 鑑賞時間	→		**コスト少**
利得：	不適切ならちょっとがっかり	→	損失 小	**期待利得 大**
	人生が変わるかも？	→	損失 大	

低コスト
高期待利得
→
推薦の強さが弱い
低確信度

推薦システムの実行過程

[Konstan+ 03]

O-I-Pモデル (Output-Input-Process)

データ入力

入力データの種類

- **嗜好データ (preference data)**
  各種のアイテムに対して，利用者がその好き嫌いの度合いを示す
  例：アイテム1＝好き，アイテム2は，5段階評価で2
- **検索質問 (query)**
  利用者が好むアイテムについての具体的な記述
  例：価格は6000以下の和食の店
- **アイテムの特徴 (item feature)**
  推薦されるアイテムを特徴ベクトルで記述
- **デモグラフィックな特徴 (demographic feature)**
  利用者自身の年齢や住所などの特徴
- **コンテキスト(利用状況)の特徴 (context feature)**
  推薦されたアイテムを使う日時や場所，アイテムの在庫状況など

嗜好データの獲得

活動利用者

嗜好データ

どんなものが好きで
どんなものが嫌いか
という情報

推薦システム

- **明示的な獲得**：利用者に好きか嫌いか尋ねる
- **暗黙的な獲得**：利用者の行動から好き嫌いを予測

嗜好データの
明示的獲得 vs 暗黙的獲得

嗜好データの明示的な獲得

明示的な獲得：利用者に好きか嫌いか尋ねる

採点法 (scoring method)

この本は
面白かったですか？

あまり面白くなかった
５段階評価で★☆☆☆☆

格付け法 (rating method)

この服のデザインは
どうですか？

上中下でいうと中

嗜好データの暗黙的な獲得

暗黙的な獲得：利用者の行動から好き嫌いを予測

このDVD買います

買ったDVDは
好きに違いない

このWebページは
じっくり見よう

閲覧時間が長いから
とても興味があるんだろう

明示的 vs 暗黙的：データ量

利用者がアイテムについて好き嫌いを評価するのは手間

~~明示的獲得~~

これ評価して　　またぁ！面倒くさい

大量のアイテムの嗜好について
明示的に回答するのは無理

暗黙的獲得

この店おいしそう　　こっちもいいな

利用者とのインタラクション
だけで随時データ収集可能

明示的 vs 暗黙的： 評価の正確さ

嗜好パターンと強く結びついた行動を利用しても，適切に好き嫌いが表されていない場合も

明示的獲得

暗黙的獲得

明示的に好き嫌いを答えてもらえば，より直接的にアイテムへの嗜好が分かる

行動から得たデータが，利用者の嗜好を正確に表さない場合も多い

明示的 vs 暗黙的：未評価と不支持

あるアイテムに対する利用者の能動的な反応は肯定的評価と考えられるが，そうでないときは不明確

明示的獲得

こんなの嫌い　　これは嫌いと

嗜好を尋ねていないものは未評価，否定的な回答を得たアイテムは不支持と分かる

暗黙的獲得

見てないWebページ

閲覧しなかったからといって，不支持ではなく，単に知らずに未評価かもしれない

明示的 vs 暗黙的：利用者の認知

明示的に質問すれば，利用者は嗜好データをシステムに入力したことを認知できる

明示的獲得

暗黙的獲得

これなんかどう？　　あのとき　ああ答えたからね

これ好きでしょう？　　納得いかないなぁ

提示された推薦と，自分が過去に示した嗜好とが結び付くと推薦に納得しやすい

根拠が不明確になりがちで信用されなかったり，プライバシの侵害と感じることも

明示的獲得 vs 暗黙的獲得

	明示的獲得	暗黙的獲得
データ量	少ない	多い
評価の正確さ	正確	不正確
未評価と不支持	明確	不明確
利用者の認知	認知	不認知

評価値と評価インターフェース

評価スケールの影響

[Cosley+ 03]

- 5starに加え，Binary，0なしの-3〜+3，Half-Starのスケール

アイテムの再評価の一貫性

- 6週間後での評価で相関は 0.83 [Hill+ 95]
- 数ヶ月〜数年：60%がそのまま，20%が下降，20%が上昇，相関 0.70

スケールへの選好

- 5段階評価で質問：half-star=4.2，5star=3.8，no-zero=3.2，binary=2.2
- 最も細かい尺度を好む．0なしスケールは嫌われる．元の5段階尺度になじみがあるようだ

スケールの違いが評価の分布に与える影響

- binary や non-zeroでは評価は高くなる
- 全般的に元の評価値の相関は高く，分布の形状も類似
- 5star→binaryでの調査：中間評価の3は肯定的に写されやすい

評価スケールの影響

[Sparling+ 11]

実験条件

- 1値(unary)，2値(binary)，五つ星(five-star)，スライダー(100点)
- 過去の記憶に基づいて評価する映画と，その場で評価する商品レビュー
- 二次刺激：評価自体と，スケールの違いの負荷を分離する目的

得られた知見

- 細かいスケールでは評価時間がかかり，また，評価の中断も多かった
- 1値とスライダーのスケールの評価時間の相対差は映画が大きかった (62% vs 25%) が，絶対時間ではレビューの方が大きかった (2秒 vs 3.5秒)
- 評価値が中間的なときに評価時間は長くなる
- 全般的に五つ星スケールが好まれるが，レビューでは2値評価も次点

評価時に予測評価を表示する影響

[Cosley+ 03]

評価時に以前の評価を予測評価として見せたときの影響

- 見せた場合も見せなかった場合も元の評価との差は0.01
- 予測を見せた場合の方が有意に同じ評価をした

以前の値の±1の評価を予測評価としてみせたときの影響

- +1を示すと評価値は0.14増加，-1を示すと0.16減少

未評価アイテムで，予測評価を改竄して表示したときの影響

- 正確な予測とその±1いずれの提示でも評価は悪めに評価
- 正確な予測を提示したときの評価を基準にすると，+1では0.15上昇，-1では0.23下降
- 予測値がずらされていることに気づく利用者はかなりいる

評価ゆらぎとバイアス

評価のゆらぎ

- 同じアイテムを6週間後に評価すると，利用者の評価値には一部違っており，その相関は0.83 [Hill+ 95]
- 利用者の評価時に，予測評価などを提示すると，評価はそれに影響される [Cosley+ 03]

評価のバイアス

- 利用者の評価値は，一般に良い方が多くなる傾向がある

この評価値のバイアスは予測精度の評価や，評価値の予測に悪影響

MovieLens
[MovieLens]

Amazon.com
[Weigend 03]

寿司
[Kamishima 06]

採点・格付け法と順位法

[kamishima 03]

採点法・格付け法 (scoring/rating method)

5段階などの点数や「優良可」などの段階で解答

例：被験者が 対象A を好きならば，尺度の「好き」を選ぶ

順位法 (ranking method)

計測する度合いの強さの順に対象を整列

例：被験者は 対象A が最も好きで，対象B が最も嫌い

嗜好の予測

嗜好の予測

考慮すべき性質

- **データ：**利用者数，アイテム数，嗜好データの数‥‥
- **予測評価値：**予測誤差，セレンディピティ，被覆率‥

※ 機械学習を使わずに人間が与えたルールで予測するものもある

内容ベースフィルタリング
VS
協調フィルタリング

内容ベースフィルタリングと協調フィルタリング

推薦システム

推薦に使う手がかりが違う二つの方法

内容ベース フィルタリング
Content-Based Filtering

協調 フィルタリング
Collaborative Filtering

アイテムの特徴を利用

他人の意見を利用

※アイテム：推薦する商品や情報

これらを組み合わせたハイブリッド法が主流に

内容ベースフィルタリング

アイテムの特徴を利用

特徴：アイテムの内容を表す情報

新聞記事 ⇒ 題名，日付，本文‥‥

映画 ⇒ 題名，監督，ジャンル，俳優，受賞‥‥

ラーメン ⇒ スープのもと，味付け，麺の太さ‥‥

内容ベースフィルタリング

直接指定型　**どんなものが好きかを具体的に入力**

とんこつの太麺のラーメンが食べたいな～

間接指定型　**利用者の過去の嗜好データを使う**

うま～

とんこつ
ちじれ麺
ナルトあり
チャーシュ薄
替え玉あり

まず～

醤油
細麺
ナルトなし
チャーシュ厚
替え玉なし

うま～

とんこつ
細麺
ナルトあり
チャーシュ厚
替え玉なし

多分，こういうのが好き

とんこつ
・
ナルトあり
・
・

利用者プロファイル

利用者がどんな「特徴」をもったアイテムを好むのかという情報

内容ベースフィルタリング
直接指定型

内容ベースフィルタリング
間接指定型

協調フィルタリング

他人の意見を利用した「口コミ」による推薦

嗜好が似ている人が好きなものを推薦する

ラーメンの嗜好が似ている人たち

協調フィルタリング

活動利用者
嗜好データ
比較
標本利用者
似ている
オススメ
中 間
オススメ
似てない
オススメ
利用者DB
要約
推薦システム
オススメ

協調 vs 内容ベース：セレンディピティ

セレンディピティ(Serendipity) ＝
単に知らないだけでなく，意外性もある推薦

意外性のある発見は楽しい！

協調フィルタリング

自分の知らないことを，他人は知っていることも

内容ベースフィルタリング

利用者プロファイルは自分の知っていることだけ

目新しさとセレンディピティ

[Herlocker+ 04]

目新しさ
Novelty

興味＋新規性

白沢監督の
最新作映画だよ

長い間
待ってたよ

セレンディピティ
Serendipity

興味＋新規性＋意外性

この新人監督の映画
もきっと気に入るよ

そんなのもあるんだ
面白そうだね

※ 関係した概念：突出性(salience)は推薦の思い切りのよさ

協調 vs 内容ベース：ドメイン知識

ドメイン知識：推薦対象の特徴など

- ドメイン知識を管理するDBの維持は高コスト
- 利用者が選択規準にする特徴の情報がなければ推薦は不可能
- どの特徴を採用するかが，利用者の意志決定に影響を与える [Häubl 01]

協調フィルタリング

そもそも内容記述は不要

内容ベースフィルタリング

アイテムの特徴

タイトル

ジャンル

サスペンスなら何でも
いいってわけじゃ・・・・
あの俳優が出てなくちゃ

アイテムを表す特徴によっては
よい推薦はできない

協調 vs 内容ベース：start-up問題

start-up問題 (cold-start問題) ＝
使い始めの利用者への推薦や新アイテムの推薦

嗜好データのない利用者や,
アイテムは推薦できない

アイテムの特徴でどうにかなる
直接指定型なら新規利用者でもＯＫ

協調フィルタリングと利用者数

協調フィルタリングは利用者数が少ないと稼働しない

協調フィルタリングシステムの運用を始めると…

start-up問題：負のフィードバック・ループ

- データベース中にはあっても，誰にも評価されていないアイテムは推薦されない
- 嗜好パターンが少数派の利用者には嗜好パターンが類似している標本利用者がいない

協調 vs 内容ベース：システムの利用者数

協調フィルタリングは，嗜好の予測精度が利用者数に依存しているので，システムの最大利用者数が限られた小さなコミュニティでは運用は難しい

協調フィルタリング

だれか教えて

し〜ん‥‥

システムの利用者数が少ないと良い推薦はできない

内容ベースフィルタリング

麺を味わうなら

やっぱり塩

それなら‥‥

アイテムの特徴があれば，他の利用者は関係ない

協調 vs 内容ベース：被覆率

$$\text{被覆率(Coverage)} = \frac{\text{推薦候補にできるアイテム数}}{\text{とりそろえた全アイテム数}}$$

とりそろえたアイテム全てを推薦できるわけではない

~~協調フィルタリング~~

たこ焼き味アイス

嗜好が似た人がだれも試していないアイテムは無理

内容ベースフィルタリング

たこ焼き味アイス

ソース味が好きな人にはお薦めだね

アイテムの特徴を手がかりにすればどのアイテムでもOK

協調 vs 内容ベース：類似アイテム

アイテムの類似性

アイテムの特徴に基づいた類似性の判定ができる.

~~協調フィルタリング~~

色違いばかり
見せられても…

たとえ色違いでも，アイテムIDが違えば違うもの

内容ベースフィルタリング

この色が
お気に入り

アイテムのいろいろな特徴に基づいて推薦ができる

協調 vs 内容ベース：少数派の利用者

少数派の嗜好の人でも適切な推薦を受けられるか？

似た嗜好の人がいない利用者は，適切な推薦をいつまでも受けられない

利用者プロファイルさえ獲得できれば，適切な推薦を受けられる

協調 vs 内容ベース：まとめ

[Balabanović+ 97, Burke 02]

	協調	内容ベース
セレンディビティ	😊	☹
ドメイン知識	😊	☹
start-up問題	☹	😋
システムの利用者数	☹	😊
被覆率	☹	😊
類似アイテム	☹	😊
少数派の利用者	☹	😊

協調フィルタリングの
アルゴリズム

問題設定：明示的評価

利用者から，アイテムへの嗜好データを明示的に集める場合

どんぶり店の例

顧客が食べたことのあるどんぶりに対して，好き嫌いを評価

評価値行列 $\mathbf{S}$

	1.親子丼	2.牛丼	3.海鮮丼	4.カツ丼
1.山田	1	3	?	3
2.田中	?	1	3	?
3.佐藤	2	1	3	1
4.鈴木	1	3	2	?

利用者 i がアイテム j を3段階の採点法で評価

	嫌い	中間	好き	不明
S_{ij} =	1	2	3	?

例：利用者1の山田さんはアイテム2の牛丼が好き：$s_{12} = 3$

評価値が不明の欄を他の評価値を使って予測

問題設定：暗黙的評価

利用者から，アイテムへの嗜好データを暗黙的に集める場合

レストラン案内サイト例：利用者が閲覧したことのあるページを記録

	1.トップ	2.和食	3.中華	4.フレンチ	5.エスニック
1.山田	○		○	○	○
2.田中	○	○		○	
3.佐藤	○	○	○		○
4.鈴木	○	○		○	○

利用者 i がアイテム j を肯定的に評価したとき $S_{ij}=1$ 未評価のとき $S_{ij}=0$

例：利用者3の佐藤さんはアイテム2の和食ページを閲覧：$s_{32} = 1$

未評価セルは必ずしも否定的とは限らないが，未評価セルを否定的と仮定し，未評価の部分が肯定／否定を予測

推薦タスクの分類

[Herlocker+ 04, Gunawardana+ 09]

適合アイテム発見 (find good items)

- 利用者の嗜好に適合するアイテムを，何か見つけ出す
  - **明示的評価**：予測評価値が高いアイテムを推薦
  - **暗黙的評価**：肯定的と予測されたアイテムを推薦
- 利用者は積極的な意志決定の動機を持っていると想定

  例：今から食べに行くレストランを決める

評価値予測 (predicting ratings)

- アイテムに利用者が付けるであろう評価値を予測する
  - **明示的評価**：予測された評価値をそのまま提示
  - **暗黙的評価**：肯定的な予測の確からしさを評価値とする
- 明確な意志決定はなく，リストを閲覧する状況を想定

  例：レストランの紹介Webサイトを眺めて楽しむ

推薦タスクの分類

[Herlocker+ 04, Gunawardana+ 09]

- 適合アイテム列挙 (find all good items)
  - 自分の嗜好に適合するものを全て列挙する
  - 意思決定にあたって候補を網羅的に検討したい状況を想定
    例：会社の法務部門での特許検索
- 効用最適化 (optimizing utility)
  - cross-sell の増加や，究極的には利用者の満足などの効用を最大化
  - 適合アイテム発見タスクの適合度を効用とみなすと，その一般化と考えられるタスク
  - 例：小売店での長期的な売上を効用とし，それを最大化

大規模データ＆高速性

大規模データ

利用者数：数十万〜数百万
アイテム数：数十〜数百万

高速性

応答時間：数百ミリ秒
同時要求数：10〜1000/秒

↓

多くの計算時間
が必要

↓

大量のデータは
処理できない

相反する要求に応える必要

疎なデータ

好き/嫌いをシステムに対して答えるのは面倒

アイテムは多数

嗜好データはごく一部

疎なデータ (Sparse)

ほとんどの対象についてデータは不明

手がかりが少ないので，機械学習アルゴリズムが規則性を見いだすのが困難になる

協調フィルタリング手法の分類

メモリベース法 (memory-based method)

蓄積したデータを推薦時に直接用いて予測

- **利用者間型**
- **アイテム間型**

モデルベース法 (model-based method)

事前に調べておいたデータの規則性を使って予測

- **クラスタモデル**
- **関数モデル**
- **確率モデル** (履歴条件型，共起型)
- **時系列モデル** (マルコフ過程，マルコフ決定過程)

メモリベース法とモデルベース法

[Breese+ 98]

	メモリベース法	モデルベース法
推薦システムが利用される前	データ	データ
	↓ 何もしない	↓ データの規則性(モデル)を見つけておく
推薦システムの利用時	データ	モデル
	↓ データ自体から推薦候補を予測	↓ モデルを使って推薦候補を予測
	予測評価値	予測評価値

メモリベース法 vs モデルベース法

	メモリベース法	モデルベース法
推薦時間	推薦時にデータを毎回，調べるので時間がかかる	事前にデータの規則性は調べてあるので素早い推薦
適応性	利用者やアイテムが変わってもモデルを作り直しは不要	利用者やアイテムが変わったらモデルを作り直す

メモリベース法

蓄積したデータを推薦時に直接用いて予測

機械学習の観点からは *k*-近隣法 とみなせる

利用者間型

user-user type

活動利用者と嗜好パターンが類似している標本利用者が好むアイテムを推薦

アイテム間型

item-item type

活動利用者が好むアイテムと，嗜好パターンの観点から類似しているアイテムを推薦

利用者間型メモリベース法

利用者間型メモリベース法

活動利用者と嗜好が似ている人が好きなアイテムを推薦

1 いろいろなアイテムへの好き嫌いの傾向が類似している利用者を見つける

2 類似した標本利用者が好むものは，活動利用者も好むと予測

GroupLensの方法

[Resnick+ 1994]

GroupLens

ミネソタ大のJohn　Riedlを中心としたグループが開発した，協調フィルタリングシステムのパイオニア

現在は映画の推薦システム MovieLens へ改良

http://www.movielens.org/

メモリベース法の代表的なアルゴリズム
利用者間の類似度を使う方法

- **Step1**：嗜好パターンが似ている利用者を見つける
- **Step2**：似ている嗜好から活動利用者の嗜好を予測

利用者の類似度

- **Step1**：嗜好パターンが似ている利用者を見つける

利用者 i と利用者 j の類似度

好きなもの ＆ 嫌いなものが同じ ➡ 類似度 大

好きなもの ＆ 嫌いなものが違う ➡ 類似度 小

相関係数 (Correlation Coefficient)

$$\frac{\sum_k (s_{ik} - \bar{s}_i)(s_{jk} - \bar{s}_j)}{\sqrt{\sum_k (s_{ik} - \bar{s}_i)^2}\sqrt{\sum_k (s_{jk} - \bar{s}_j)^2}}$$

$\bar{s}_i$ は利用者 i の平均評価値

相関係数は最大値が 1 で，最小値が -1

評価値の計算

- **Step2**：似ている嗜好から活動利用者の嗜好を予測

利用者 i のアイテム k についての推定評価値

利用者 i が評価した全てのアイテムについての平均評価

利用者 i と利用者 j の類似度

アイテム k への標本利用者 j の評価の平均からの偏差

$$\hat{s}_{ik} = \tilde{s}_i + \frac{\sum_{j,j\neq i} r_{ij}(s_{jk} - \bar{s}_j)}{\sum_{j,j\neq i} |r_{ij}|}$$

全標本利用者についての和

評価アイテム数の違いに関する補正

GroupLensの方法

詳細版

GroupLensの方法

Step1：嗜好パターンが似ている利用者を見つける

「嗜好パターンが似ている」とは？

似ている度合い（=**類似度**）を数値で表し，類似度が大きいほど嗜好パターンが似ている

「類似度」に必要な条件とは？

好きなものが同じ
嫌いなものが同じ → **類似度 大**

好きなものが違う
嫌いなものが違う → **類似度 小**

GroupLensの方法

- Step2：似ている嗜好を参考に，活動利用者の嗜好を予測

	推薦の判断の対象		
	好き	中間	嫌い
嗜好パターンが似ている人の嗜好	推薦	保留	却下
嗜好パターンが違っている人の嗜好	却下	保留	推薦

例：どんぶり専門店「丼兵衛」

計算するとき扱い易いように番号を割り当てる

利用者 4人の常連客

1. 山田　2. 田中　3. 佐藤　4. 鈴木

アイテム 4種類のどんぶり

1. 親子丼　2. 牛丼　3. 海鮮丼　4. カツ丼

利用者の評価 3段階の数値で評価してもらう

利用者 i の
アイテム j の評価

$S_{ij} =$

嫌い	中間	好き	不明
1	2	3	?

例:　利用者1の山田さんはアイテム2の牛丼が好き

$$s_{12} = 3$$

例：どんぶり専門店「丼兵衛」

	1.親子丼	2.牛丼	3.海鮮丼	4.カツ丼
1.山田	1	3	?	3
2.田中	?	1	3	?
3.佐藤	2	1	3	1
4.鈴木	1	3	2	?

利用者の類似度

「類似度」に必要な条件とは？

好きなもの ＆ 嫌いなものが同じ → 類似度 大

好きなもの ＆ 嫌いなものが違う → 類似度 小

利用者 i の平均評価値を $\overline{s}_i$ で表す

		嫌い	中間	好き
$s_{ik} - \overline{s}_i$	$=$	負	0	正

利用者の類似度

利用者 i と利用者 j の類似度について考える

好き嫌いが同じ ➡ $(s_{ik}-\bar{s}_i)(s_{jk}-\bar{s}_j) > 0$

好き嫌いが違う ➡ $(s_{ik}-\bar{s}_i)(s_{jk}-\bar{s}_j) < 0$

相関係数 (Correlation Coefficient)

これを全てのアイテム(=どんぶり)について和をとると

$$\frac{\sum_k (s_{ik} - \bar{s}_i)(s_{jk} - \bar{s}_j)}{\sqrt{\sum_k (s_{ik} - \bar{s}_i)^2}\sqrt{\sum_k (s_{jk} - \bar{s}_j)^2}}$$

最大値が 1 で，最小値が -1 になるようにする工夫

利用者の類似度

例：利用者2の田中と利用者3の佐藤の類似度を考える

	1.親子丼	2.牛丼	3.海鮮丼	4.カツ丼
2.田中	?	1	3	?
3.佐藤	2	1	3	1

注意 利用者が共に評価しているものだけで計算

➡ 牛丼と海鮮丼の評価値は計算に含めるが，親子丼とカツ丼の評価値は無視する

田中の平均評価 $\bar{s}_2 = \frac{1}{2}(s_{22} + s_{23}) = \frac{1}{2}(1+3) = 2$

佐藤の平均評価 $\bar{s}_3 = \frac{1}{2}(s_{32} + s_{33}) = \frac{1}{2}(1+3) = 2$

利用者の類似度

	1.親子丼	2.牛丼	3.海鮮丼	4.カツ丼
2.田中	?	1	3	?
3.佐藤	2	1	3	1

注意 利用者が共に評価しているものだけで総和を計算

分子

$$\sum_{k=2,3}(s_{2k}-\bar{s}_2)(s_{3k}-\bar{s}_3) = (1-2)(1-2) + (3-2)(3-2) = 2$$

分母

$$\sum_{k=2,3}(s_{2k}-\bar{s}_2)^2 = (1-2)^2 + (3-2)^2 = 2$$

$$\sum_{k=2,3}(s_{3k}-\bar{s}_3)^2 = (1-2)^2 + (3-2)^2 = 2$$

利用者の類似度

	1.親子丼	2.牛丼	3.海鮮丼	4.カツ丼
2.田中	?	1	3	?
3.佐藤	2	1	3	1

相関係数は1で，田中と佐藤の嗜好パターンは完全に一致

$$\frac{\sum_{k=2,3}(s_{2k}-\bar{s}_2)(s_{3k}-\bar{s}_3)}{\sqrt{\sum_{k=2,3}(s_{2k}-\bar{s}_2)^2}\sqrt{\sum_{k=2,3}(s_{3k}-\bar{s}_3)^2}}=\frac{2}{\sqrt{2}\sqrt{2}}=1$$

特殊な場合：二人が共通に評価しているものが1個以下なら，相関係数は 0 とみなす

評価値の計算

相関係数の性質

利用者 i と利用者 j の評価値の相関係数を r_{ij} で表す

二人の嗜好パターンが似ている	➡	$r_{ij} > 0$
二人の嗜好パターンが違っている	➡	$r_{ij} < 0$

利用者 j の嗜好をもとに，アイテム k を利用者 i に推薦すべきかを判断する

嗜好パターン	利用者 j の嗜好	推薦する?
r_{ij}	$s_{jk} - \bar{s}_j$	$r_{ij}(s_{jk} - \bar{s}_j)$
類似(+)	好き(+)	推薦(+)
相違(-)	好き(+)	却下(-)
類似(+)	嫌い(-)	却下(-)
相違(-)	嫌い(-)	推薦(+)

評価値の計算

アイテム k の利用者 i への推薦を,
利用者 i 以外の全ての利用者の嗜好を基に計算

$$\boxed{\sum_{j,j\neq i} r_{ij}(s_{jk}-\bar{s}_j)}$$

問題点

アイテム k を未評価の利用者の嗜好は無視するので，多くの利用者に評価されたものほど推薦されやすい

↓

正規化

$$\boxed{\frac{\sum_{j,j\neq i} r_{ij}(s_{jk}-\bar{s}_j)}{\sum_{j,j\neq i} |r_{ij}|}}$$

評価値の計算

$$\frac{\sum_{j,j\neq i} r_{ij}(s_{jk}-\bar{s}_j)}{\sum_{j,j\neq i} |r_{ij}|}$$

－ → 却下　　0 → 中立　　＋ → 推薦

中立のとき利用者 i の平均スコアになるように補正

利用者 i のアイテム k についての推定評価値

$$\tilde{s}_i + \frac{\sum_{j,j\neq i} r_{ij}(s_{jk}-\bar{s}_j)}{\sum_{j,j\neq i} |r_{ij}|}$$

$\tilde{s}_i$ は利用者 i が評価した全てのアイテムについての平均評価

評価値の計算

例：利用者2の田中のアイテム1の親子丼の予測評価値

平均評価値と相関係数の計算

田中と山田の相関係数

共通して評価しているアイテム数が1個だけなので相関係数は0．よって無視する

田中と佐藤の相関係数

$$\bar{s}_3 = 2 \qquad r_{23} = 1$$

田中と鈴木の相関係数

$$\bar{s}_4 = \frac{5}{2} \qquad r_{24} = -1$$

評価値の計算

総和は相関を計算できなかった利用者1(山田)を除いた利用者3(佐藤)と利用者4(鈴木)について考える

評価値

$$\sum_{j=3,4} r_{ij}(s_{jk} - \bar{s}_j)$$
$$= r_{23}(s_{31} - \bar{s}_3) + r_{24}(s_{41} - \bar{s}_4)$$
$$= 1(2-2) + (-1)(1 - \frac{5}{2}) = \frac{3}{2}$$

正規化係数

$$\sum_{j=3,4} |r_{ij}| = |r_{23}| + |r_{24}|$$
$$= |1| + |-1| = 2$$

評価値の計算

利用者2の平均評価値

$$\tilde{s}_2 = \frac{1}{2}(s_{22} + s_{23}) = \frac{1}{2}(1+3) = 2$$

利用者2のアイテム1の予測評価値

$$\tilde{s}_2 + \frac{\sum_{j=3,4} r_{2j}(s_{j1} - \bar{s}_j)}{\sum_{j=3,4} |r_{2j}|}$$

$$= 2 + \frac{3/2}{2} = \boxed{2.75}$$

予測評価値はかなり大きいので，利用者2番の田中はアイテム1番の親子丼が好きと予測できる

まとめ

GroupLensの方法

ミネソタ大のJohn　Riedlを中心としたグループが開発した，協調フィルタリングシステムのパイオニア

メモリベース法の代表的なアルゴリズム
利用者間の類似度を使う方法

- **Step1**：嗜好パターンが似ている利用者を見つける
- **Step2**：似ている嗜好から活動利用者の嗜好を予測

まとめ

- **Step1**：嗜好パターンが似ている利用者を見つける

相関係数 (Correlation Coefficient)

利用者 i と利用者 j の間の相関係数

$$r_{ij} = \frac{\sum_k (s_{ik} - \bar{s}_i)(s_{jk} - \bar{s}_j)}{\sqrt{\sum_k (s_{ik} - \bar{s}_i)^2}\sqrt{\sum_k (s_{jk} - \bar{s}_j)^2}}$$

まとめ

- **Step2**：似ている嗜好から活動利用者の嗜好を予測

利用者 i のアイテム k についての推定評価値

$$\hat{s}_{ik} = \tilde{s}_i + \frac{\sum_{j, j \neq i} r_{ij} (s_{jk} - \bar{s}_j)}{\sum_{j, j \neq i} |r_{ij}|}$$

アイテム間型メモリベース法

アイテム間型メモリベース法

活動利用者が好きなアイテムと類似したアイテムを推薦

1

いろいろな標本利用者からの好き嫌いの傾向が類似しているアイテムを見つける

2

活動利用者が好むアイテムと類似したアイテムを好むと予測

アイテム間型メモリベース法

類似アイテムを推薦する方法

アイテムベクトル：あるアイテムへのいろいろな利用者の評価を表す

‖

アイテム j に対するアイテムベクトルは評価値行列 S の第 j 列ベクトル

アイテムの類似度 ＝ アイテムベクトルの類似度

ベクトルの類似度：コサイン，相関係数，ユークリッド距離など

注目中のアイテム

- 利用者が高評価
- 買い物かごに入っている
- 商品ページを閲覧中

→ 注目中のアイテムと類似しているアイテムを推薦

- **Amazon.com** [Linden+ 04] や **TiVo** [Ali+ 04] で利用される
- 実装は容易で，アイテム間の類似度だけで推薦が可能
- 閲覧や購入といった利用者の行動に対応した推薦に便利

アイテム間型メモリベース法

[Sarwar+ 01]

Sarwar の方法

- 評価するアイテムと類似したアイテムの類似度で重み付けした，類似アイテムへの活動利用者の評価の平均

活動利用者のアイテムjへの予測評価スコア

アイテム k への活動利用者の評価

$$\hat{s}_{aj} = \frac{\sum_k r_{kj} s_{ak}}{\sum_k |r_{kj}|}$$

アイテム j と似ているアイテムについての和

アイテム k とアイテム j の類似度

評価ゆらぎの軽減

評価値には「ゆらぎ」➡バイアス項を除去　非常に有効

平均値による標準化

[Herlocker+ 99]

利用者の評価値の，利用者ごとの平均が 0 となるように事前に標準化

$$s'_{ij} = s_{ij} - \frac{1}{|\mathcal{U}_i|} \sum_{j \in \mathcal{U}_i} s_{ij}$$

利用者 i が評価した
アイテムの集合

回帰を使った補正

[Bell+ 07]

回帰モデル

$$s_{ij} = \theta_i x_{ij} + N(0, \epsilon)$$

評価値　パラメータ　説明変数　ノイズ

最小二乗で解くと…

$$\hat{\theta}_i = \sum_i s_{ij} x_{ij} / \sum_i {x_{ij}}^2$$

※ 疎データ対策で実際
には縮小推定する

i が j を評価したとき x_{ij} は1，それ以外で0ならば利用者の平均に相当

スコアの更新式：

$$s'_{ij} = s_{ij} - \hat{\theta}_i x_{ij}$$

s'_{ij} について，アイテムへの評価など様々な説明変数での回帰を適用

メモリベース法の改良

- **近傍の利用**

  全ての利用者だけでなく，類似度の大きな利用者だけを用いて予測スコアを計算すると，予測精度を下げずに高速化ができる．

- **平均値**

  ステップ2の第2項の $\tilde{s}_i$ だが，利用者 j の全アイテムへの評価の平均 $\bar{s}_i$ と置き換えても精度にあまり影響はない．

- **デフォルト投票**

  類似度を計算するとき，二人が同時に評価しているアイテムが少ないと，類似度を高精度で計算できない．よって，一方だけが評価している場合，もう一方の利用者の評価値として，そのアイテムへの全利用者の平均値を使う． [Breese+ 98]

- **類似度関数の学習**

  Pearson相関や順位相関が利用者間の類似度としてよく用いられている．

  さらに改良するために，利用者 i と j の類似度 r_{ij} をパラメータと考え，経験損失を最小化するように類似度行列を学習する [Bell+ 07]

メモリベース法：まとめ

- 利用者間型では，　類似度関数の学習がなければ，　データベースの更新にすぐに対応できる．推薦時の計算量の問題は，活動利用者の近隣の利用者の嗜好データのみを使うことでほぼ解消できる
- アイテム間型は，事前にアイテム間の類似度行列を計算できていれば，推薦時も高速．同じアイテムが推薦されやすくセレンディピティは低いといわれる(portfolio効果) [Ziegler+ 05] アイテムの類似度行列だけで推薦可能なので，TiVo のようなローカル端末に有利.
- 類似度関数を学習を導入すればかなり向上するが，　予測精度についてはモデルベース法に対してやや不利 [Koren 09]
- サクラ攻撃については，モデルベースより不利との報告がある [Mehta+ 08, Mobasher+ 07] 利用者間型は平均攻撃 [Lam+ 04] に，アイテム間型はセグメント攻撃 [Burke+ 05] に弱いの報告.

モデルベース法

事前に調べておいたデータの規則性を使って予測

- **クラスタモデル**
  嗜好が類似した利用者のグループごとに推薦をする
- **関数モデル**
  利用者の嗜好パターンから，アイテムの評価値を予測する関数
- **確率モデル**
  - **履歴条件型**：過去の嗜好パターンが与えられたときに，あるアイテムを好む条件付き確率
  - **共起型**：ある利用者があるアイテムを好む確率
- **時系列モデル**
  - **マルコフ過程**：アイテムを評価した時間的順序も考慮
  - **マルコフ決定過程**：加えて，利用者の行動もモデル化

クラスタモデル

[Breese+ 98, George+ 05]

類似度を相関行列などで測り，嗜好が類似した利用者グループを生成

個人化の度合いは弱く，大きくははずさない推薦

gray sheep問題：複数の嗜好パターンが混在する利用者に弱い[Burke 02]

関数モデル

関数モデル

推薦候補のアイテムの特徴と活動利用者の特徴を入力とし，その活動利用者のそのアイテムへの予測評価値を出力する関数を利用

$$\hat{s}_{ij} = f(\mathbf{x}_i, \mathbf{y}_j)$$

予測評価値　利用者 i の特徴　アイテム j の特徴

入力値

- **利用者の特徴：**その利用者の他のアイテムへの評価など
- **アイテムの特徴：**そのアイテムへの他の利用者の評価など

出力値

- **回帰問題：**より好ましいときに，より大きな値を出力する関数
- **分類問題：**{好き, 嫌い} の二値カテゴリ値を出力する関数
- **順序回帰問題：**{上, 中, 下} といった順序カテゴリ値を出力する関数

既存の多種多様な予測手法が使えて，いろいろな拡張が可能

回帰問題：行列分解の概念

潜在因子：嗜好のパターン
※ラベルは説明用のもの

予測評価値：$\hat{s}_{ij} = u_{i1}v_{j1} + u_{i2}v_{j2} + \cdots + u_{ik}v_{jk}$

利用者とアイテムが共に同じ潜在因子と関連が大きいと高評価

回帰問題：行列分解

[Canny 02, Koren 08, Koren 09, Mehta+ 08, Weimer+ 08]

- 評価値行列 S を，利用者行列 U とアイテム行列 V に分解し，低ランク近似で，データが疎である問題に対処
- 利用者 i のアイテム j への評価は，U の第 i 行ベクトルの転置 $\mathbf{u}_i$ と V の第 j 列ベクトル $\mathbf{v}_j$ の積

$$\boxed{s_{ij} = \mathbf{u}_i^\top \mathbf{v}_j}$$

- 行列分解手法：特異値分解，因子分析，NMFなど
- 欠損値補完手法：平均値で補完，EMアルゴリズム，欠損値を無視

行列分解：問題の定式化

直接的な方法

欠損値補完手法 平均値で補完 ＋ 行列分解手法 特異値分解

↓

補完した値でデータが歪められる + 計算量が多い O(m n k)

欠損値を無視して定式化 次の損失関数を最小化する

評価済みの利用者とアイテムの対の集合　　正則化パラメータ

$$f(U,V) = \underbrace{\frac{1}{2}\sum_{(i,j)\in\mathcal{K}} \left(s_{ij} - \mathbf{u}_i^\top \mathbf{v}_j\right)^2}_{\text{予測値と観測値の残差の二乗和}} + \underbrace{\frac{\lambda}{2}\left(\|U\|_F^2 + \|V\|_F^2\right)}_{\text{正則化項}}$$

- データが疎で $|\mathcal{K}| \ll mn$ であるため，行列全体を特異値分解するのと比較して効率的に計算可能
- 不支持と未評価が区別可能な明示的評価が前提

行列分解：勾配降下法

$$f(U,V) = \frac{1}{2}\sum_{(i,j)\in\mathcal{K}} \left(s_{ij} - \mathbf{u}_i^\top \mathbf{v}_j\right)^2 + \frac{\lambda}{2}(\|U\|_F^2 + \|V\|_F^2)$$

損失関数は下に凸なので勾配降下法で最小化

勾配の計算

$$\frac{\partial f}{\partial \mathbf{u}_g} = -\sum_{\substack{(i,j)\in\mathcal{K} \\ \text{s.t. } i=g}} (s_{gj} - \mathbf{u}_g^\top \mathbf{v}_j)\mathbf{v}_j + \lambda \mathbf{u}_g \qquad \frac{\partial f}{\partial \mathbf{v}_h} = -\sum_{\substack{(i,j)\in\mathcal{K} \\ \text{s.t. } j=h}} (s_{ih} - \mathbf{u}_i^\top \mathbf{v}_h)\mathbf{u}_i + \lambda \mathbf{v}_h$$

更新式

適当な値で分解後の行列 S と V を初期化し，各反復で両行列の行と列を次式で，収束するまで更新する

$$\mathbf{u}_g \leftarrow \mathbf{u}_g - \eta \frac{\partial f}{\partial \mathbf{u}_g} \qquad \mathbf{v}_h \leftarrow \mathbf{v}_h - \eta \frac{\partial f}{\partial \mathbf{v}_h}$$

行列分解：確率的勾配降下法

評価値データ集合 $\mathcal{K}$ が非常に大きい場合

データを主記憶上に保持できず，勾配の計算が非効率的

確率的勾配降下法：データ全体ではなく，データ一個ずつの勾配を求め，全てのデータについて更新を実行する

各 $(i,j) \in \mathcal{K}$ について，パラメータを次式で更新

$$\mathbf{u}_i \leftarrow \mathbf{u}_i - \eta\{-(s_{ij} - \mathbf{u}_i^\top \mathbf{v}_j)\mathbf{v}_j + \lambda \mathbf{u}_i\}$$
$$\mathbf{v}_j \leftarrow \mathbf{v}_j - \eta\{-(s_{ij} - \mathbf{u}_i^\top \mathbf{v}_j)\mathbf{u}_j + \lambda \mathbf{v}_j\}$$

- 学習率(ステップ幅) η は，反復数 t と，パラメータ η_Δ と t_0 について次式で更新：$\eta = 1/\{\eta_\Delta(t + t_0)\}$
- データ数が少ない場合は，同じデータを何度か走査する
- データを主記憶上に保持しないので大規模データに向くが，逆に小規模データでは収束しないこともある
- 多くの改良が近年行われている [Bottou]

行列分解：二値評価と評価ゆらぎ

二値評価

[Rendle+ 09]

評価が 0/1 の二値評価だと直線ではあてはめが難しい

シグモイド関数 $\sigma(x) = 1/(1+\exp(-x))$ 導入し，残差を次式で評価

$$s_{ij} - \sigma(\mathbf{u}_i^\top \mathbf{v}_j)$$

評価のバイアス

[Koren+ 09]

利用者やアイテムに固有の評価値のバイアスを同時に補正

全体のバイアス　利用者 i のバイアス　アイテム j のバイアス

$$s_{ij} - (\mu + a_i + b_j + \mathbf{u}_i^\top \mathbf{v}_j)$$

$\mathbf{u}_i$ や $\mathbf{v}_j$ と同時に，バイアスパラメータ μ, a_i, b_j も同時に推定

これらのバイアスの補正は非常に有効

- その他のバイアス：利用者が評価対象として選ぶアイテム群に依存したバイアス，利用者やアイテムごとの，評価時刻に依存したバイアス

行列分解：評価情報とハイブリッド化

利用者が評価したアイテムの情報

[Koren+ 08, Paterek 07]

SVD++：利用者が，どのアイテムを評価したかという情報は重要

利用者 i に依存した スカラーの係数

利用者 i が評価した アイテムの集合

アイテム l の効果 k 次元ベクトル

$$s_{ij} - (\mu + a_i + b_j + [\mathbf{u}_i + c_i \textstyle\sum_{l \in \mathcal{R}_i} \mathbf{w}_l]^\top \mathbf{v}_j)$$

- [Koren+ 08] では c_i を $1/\sqrt{|\mathcal{R}_i|}$ にするヒューリスティックを導入

ハイブリッド化

アイテムや利用者の特徴も考慮可能

$$s_{ij} - (\mu + a_i + b_j + \mathbf{u}_i^\top \mathbf{v}_j + \mathbf{d}_j^\top \mathbf{x}_i + \mathbf{e}_i^\top \mathbf{y}_j)$$

- $\mathbf{x}_i$：利用者 i のデモグラフィックやそのコンテキストの特徴ベクトル
- $\mathbf{y}_j$：アイテム j 自身やそのコンテキストの特徴ベクトル
- $\mathbf{d}_j$， $\mathbf{e}_j$：学習する重みベクトル

テンソル分解

[Rendle+ 09]

協調タグ付け(collaborative tagging)：アイテム i に利用者 u がタグ t を付けるかどうかを予測．行列分解の推薦システムでは好き／嫌いが1次元だが，ここではいろいろなタグに嗜好を多次元化

テンソル分解による低ランク近似

評価値テンソル: S　コアテンソル: C　利用者: U　アイテム: I　タグ: T

- **評価値テンソル**：利用者 u がアイテム i にタグ t を付けた・付けないを (u, i, t)=1/0 で表す．欠損値もある．

$$(C, U, I, T) = \arg\min \sum_{u'} \sum_{i'} \sum_{t'} \Big(S(u, i, t) - C(u', i', t') U(u', u) I(i', i) T(t', t) \Big)^2$$

行列分解：まとめ

- 不支持と未評価の区別が明確な明示的な評価の場合に適用可能な方法
- 全体・利用者・アイテムのバイアス項の効果は非常に大きい
- 利用者やアイテムごと評価の時間変化などの補正も必要に応じて可能
- 確率的勾配降下法との組み合わせで，大規模データに対して最も効率的な方法
- 利用者の嗜好プロファイルを目的関数に含めることでハイブリッド化も可能
- Netflixのデータ： 評価値数 100万，利用者数 48万，映画数 17,770 で次元数 k は 50 程度 [Koren+ 09]
- 現在，知られているサクラ攻撃に対して頑健 [Mehta+ 08]
- 秘密関数計算と組み合わせてプライバシ保護型の拡張も可能

関数モデル：分類問題

[Nakamura+ 98]

評価値 {1,2,3,4,5} などをクラスとみなしてクラス分類問題として解く

Cross-G-Leearn-Relation: 重み付き多数決を使ったオンライン学習

s_{ij} が r であることの得票値

$$\sum_{i':s_{i'j}=r} v_{ii'} + \sum_{j':s_{ij'}=r} w_{jj'}$$

$v_{ii'}$: 利用者 i と i' の間の重み

$w_{jj'}$: アイテム j と j' の間の重み

$r \in \{1,2,3,4,5\}$ 中で得票値が最大になる r を予測評価値とする

学習： s_{ij} の評価値が r であるとき，同じ列で評価値が r であるアイテムの重み $w_{jj'}$ は増やし，それ以外の重みは減らす．$v_{ii'}$ も同様.

確率モデル

確率モデル

確率：偶然に生じる事象を扱う枠組み

確率分布：事象が生じる頻度や確からしさを表す

アイテム△の好き/嫌いについて

Pr[△ → 好き]=0.8　　Pr[△ → 嫌い]=0.2

仮に利用者を適当に一人呼んでくる

↓

アイテム△を「好き」か「嫌い」か尋ねる

↓

10回のうち，8回ぐらいは好き，2回ぐらいは嫌い

確率モデル

条件付き確率：他の情報に基づいて変わる確率

Pr[△→ 好き | ○→ 好き]

対象　　条件

Pr[△→ 好き | ○→ 好き]=0.8

Pr[△→ 好き | ○→ 嫌い]=0.3

アイテム○が好きな利用者10人のうち△も8人ぐらいは好きだが，
嫌いな利用者なら10人のうち△が好きなのは3人ぐらい

条件付き確率でモデルがあれば，他のアイテムの好き嫌いから，あるアイテムを好む確率を予測できる

例：　もし ○ が好きなら，きっと △ も好き

条件付き確率

		アイテム ○	
		好き	嫌い
アイテム △	好き	☺ ☺ ☺ ☺	☺ ☺
	嫌い	☺	☺ ☺

Pr[△ → 好き | ○ → 好き] =

$$\frac{\triangle \rightarrow \text{好き かつ } \bigcirc \rightarrow \text{好き の人数}}{\bigcirc \rightarrow \text{好き の人数}} = \frac{4}{5}$$

確率モデル：履歴条件型

利用者の過去の嗜好データが与えられたときに
あるアイテムを好む条件付き確率でモデル化

Pr[△→ 好き | ○→ 好き , ◇→ 嫌い , □→ 好き , ・・・・]=0.8

アイテム数は膨大	⇔	評価値数は少ない
複雑なモデル		疎な訓練データ

↓

モデルの学習は困難

↓

ベイジアンネットなどのグラフィカルモデルを用いて簡潔に

確率という明確な理論的背景
アイテム数が少なく，利用者数が多いときに有利なモデル化

履歴条件型：ベイジアンネット

データが疎なので，複雑な条件付き確率分布の計算は無理

↓

潜在変数：条件と結論の中間の隠れた要因

ベイジアンネット：依存性にループがない場合．信念伝播(belief propagation)で解ける

確率モデル：共起型

ある一回の評価付けで，どの利用者がどのアイテムを評価したかという同時確率でモデル化

利用者 がアイテム を購入したとする

Pr[利用者 → , アイテム →]=0.02

利用者 がアイテム を5段階で 4 と評価

Pr[利用者 → , アイテム → , 評価値 → 4]=0.02

潜在変数を導入してモデルを簡潔にする

未評価と否定的評価が区別できなくても学習可能
アイテム数や利用者数について線形に複雑なので，破綻しにくい

pLSA：概念

嗜好パターン：典型的な好みのパターン　※ ラベルは説明用のもの

同時分布：
$$\Pr[x,y] = \sum_{z \in Z} \Pr[x|z]\Pr[y|z]\Pr[z]$$

共起型確率モデル：pLSA

pLSA (probabilistic Latent Semantic Analysis)

自然言語処理の次元削減法 [Hoffmann 99] を協調フィルタリングにも適用 [Hoffman+ 99]． aspectモデルともいう．

三つの多値カテゴリ変数の共起を考える

利用者： $x \in \{1, \ldots, n\}$

アイテム： $y \in \{1, \ldots, m\}$

潜在変数： $z \in \{1, \ldots, l\}$ (典型的な好みのパターン)

アイテム j_k を利用者 i_k が購入/閲覧していたら
対 (i_k, j_k) を訓練データ集合に加える

↓

不支持評価のデータが不要なので，未評価と不支持が不明確な暗黙的評価で得たデータを扱える

pLSA：定式化

訓練データ $\mathcal{D} = \{(x = i_k, y = j_k)\}_{k=1}^{N}$

同時分布 **生成モデル：**z が与えられたとき x や y は条件付独立

$$\Pr[x, y] = \sum_{z \in Z} \Pr[x|z] \Pr[y|z] \Pr[z]$$

$\Pr[x \mid z]$, $\Pr[y \mid z]$, $\Pr[z]$ はそれぞれ多項分布に従う

- 分布 $\Pr[x \mid z]$ は，各 $z = 1 \cdots l$ について，$x = 1 \cdots n$ の値をとりうる多項分布でモデル化する

  パラメータ $\Pr[x \mid z]$ の総数は $l \times n$ 個となる

- 同様に分布 $\Pr[x \mid z]$ と $\Pr[z]$ のパラメータ数は $l \times m$ 個と l 個となる

対数尤度関数： $\mathcal{L}(\mathcal{D}; \theta) = \sum_{(i,j) \in \mathcal{D}} \ln \Pr[x = i, y = j; \theta]$

pLSA：学習

pLSAのEMアルゴリズムによる最尤推定
対数尤度関数を最大化して同時確率 $\Pr[x, y]$ を推定

推定すべきパラメータ： $\theta = (\{\Pr[x|z]\}, \{\Pr[y|z]\}, \{\Pr[z]\})$

Eステップ 現在のパラメータが与えられたときの潜在変数の事後確率

$$\Pr[z|x,y] = \frac{\Pr[z]\Pr[x|z]\Pr[y|z]}{\sum_{z'}\Pr[z']\Pr[x|z']\Pr[y|z']}$$

Mステップ 現在の潜在変数の分布の下でのパラメータの値

$$\Pr[x|z] = \frac{\sum_y \#(x,y)\Pr[z|x,y]}{\sum_{x',y}\#(x',y)\Pr[z|x',y]}$$

$$\Pr[y|z] = \frac{\sum_x \#(x,y)\Pr[z|x,y]}{\sum_{x,y'}\#(x,y')\Pr[z|x,y']}$$

$$\Pr[z] = \frac{\sum_{x,y}\#(x,y)\Pr[z|x,y]}{N}$$

訓練データ中である利用者 x が，アイテム y を購入・閲覧する回数

x は1〜n，y は 1〜m の範囲で計算

尤度関数が収束するまで，EステップとMステップを交互に反復

pLSA：推薦

推薦：$x = i$ のときの事後確率を最大化するアイテム

$$y^* = \arg \max_{y \in \{1,\ldots,m\}} \Pr[y|x = i]$$

事後確率分布関数で $x=i$ を代入

同時分布：

$$\Pr[x, y] = \sum_{z \in Z} \Pr[x|z] \Pr[y|z] \Pr[z]$$

学習したパラメータ

事後確率分布：

$$\Pr[y|x] = \frac{\Pr[x, y]}{\sum_y \Pr[x, y]}$$

\#
\#

pLSA：並列分散計算

[Das+ 07]

利用者のデータを二つのサイトで分割して保持

サイト1	サイト2
$\mathcal{X}_1 = \{1, \ldots, n_1\}$	$\mathcal{X}_2 = \{n_1 + 1, \ldots, n\}$

サイト1での値のMステップの計算

$$\Pr[x|z] = \frac{\sum_y \#(x,y) \Pr[z|x,y]}{\sum_{x',y} \#(x',y) \Pr[z|x',y]}$$

$x' \in \mathcal{X}_2$ の利用者については $\#(x', y)$ の値は未知

$$\Pr[x|z] = \frac{\sum_y n(x,y) \Pr[z|x,y]}{\sum_{x' \in \mathcal{X}_1, y} \#(x',y) \Pr[z|x',y] + \sum_{x'1\#\mathcal{X}_2, y} \#(x',y) \Pr[z|x',y]}$$

サイト2からサイト1に送信

pLSA：拡張

状況に合わせて，入出力や内部の潜在変数を容易に拡張可能

aspectモデル

[Hoffmann+ 99]

評価値を導入したモデル

[Hoffmann+ 99]

評価ゆらぎを考慮 decoupledモデル

[Jin+ 03]

評価値の順序関係を考慮 preferred orderingモデル

[Jin+ 03]

ベイズ推定を導入した latent Dilichlet allocation (LDA) なども利用可

[Blei+ 03]

pLSA：まとめ

- 利用者と，その利用者が支持したアイテムの共起情報のみがあれば，不支持評価のデータが不要なので，未評価と不支持が不明確な暗黙的評価で得たデータを扱える
- 明示的評価の場合，順序評価のある場合，ハイブリッドな場合など非常に多様な拡張が可能
- 学習すべきパラメータ数さが，利用者数やアイテム数に対して線形にしか増加しないので，アイテム数が多いときには履歴条件型より有利
- pLSAの解法であるEMアルゴリズムは，総和をとる部分を複数のサイトに分割して計算させれば容易に分散並列化が可能
- GoogleNewsでは，ユーザ・アイテム数が100万規模，潜在変数の定義域の大きさは1000程度 [Das+ 07]
- 現状のサクラ攻撃に対しては頑健といわれている [Mobasher+ 07]

時系列モデル

時系列モデル

利用者の過去嗜好パターンで，評価した時間的順序も考慮する

モデルは複雑になるが，時間的な要素を考慮したいときに有効

マルコフモデル

マルコフモデル：直近の一定の個数のアイテムへの嗜好だけで，次の嗜好が決まると考える時系列モデル

店舗での購入記録

直近の2回に購入したアイテムで，次に購入するアイテムが決まる

条件付き確率分布を使ってモデル化

最大エントロピー原理

[Pavlov+ 03]

アイテム数は多いので3次程度のマルコフモデルでも困難

↓

素性関数を導入して最大エントロピー原理でパラメータを推定

(自然言語処理で言語モデルの学習に利用されている)

次のアイテム　　過去の購買履歴

$$\Pr[y^{(t+1)}|Y^{(t)}] = \frac{1}{Z}\exp[\sum_{k=1}^{K}\lambda_k f_k(y^{(t+1)}, Y^{(t)})]$$

素性の重みパラメータ　　素性関数

素性関数：履歴にある性質があれば1，でなければ0になる関数

- trigger：購買履歴にあるアイテムがあれば1をとる素性関数

マルコフ決定過程

[Shani 02, Shani 05]

マルコフモデルに加えて，推薦に対する利用者の行動も考慮する

強化学習：マルコフ決定過程で，一定期間の総購入額などの報酬の最大化をするように，行動方針(政策)を学習する枠組み

ハイブリッド法

ハイブリッド法

協調フィルタリングと内容ベースのハイブリッド

- **混合 (mixed)**
  両方の結果を同時に提示
- **切り替え (switching)**
  何らかの規準に基づいて推薦手法を切り替える
  例：初期利用者にはstart-up問題に強い内容ベース，常連利用者には協調フィルタリング
- **メタ推薦 (meta-recommendation)**
  メタ検索エンジンのように，独立に計算された推薦リストを統合して表示する
- **縦続 (cascade)**
  第1段階で候補を選別し，第2段階で順位付けをする
  例：ルールを用いた内容ベースで候補を選別し，協調フィルタリングでそれらを順位付けする

ハイブリッド法

- **特徴拡張 (feature augmentation)**
  一方の手法が出力する評価値や分類結果を，次の手法の入力に利用する
  例：協調フィルタリングで推薦される関連アイテムを，アイテムの特徴とみなして内容ベースフィルタリングを実行

- **抽象情報 (abstraction)**
  嗜好データやアイテムの特徴を，一方の手法で，ベクトルや確率分布のような抽象情報に変換して，もう一方に入力する
  例：協調フィルタリングで，嗜好データそのものの類似度ではなく，内容ベースで生成した利用者プロファイルの類似度を利用

- **完全結合 (total integration)**
  二つの手法を同等のレベルでモデル化

ハイブリッド法：完全結合

[Popescul+ 01, Ono+05, 麻生+ 06]

pLSAの拡張

次の共起を考える

- 利用者ID：$x=10$
- アイテムID：$y=3$
- アイテムの特徴：
  f=(赤, 100, No,…)

同時確率は

$$\Pr[x, y, \mathbf{f}] = \sum_{z \in Z} \Pr[z|x] \Pr[y|z] \Pr[\mathbf{f}|z] \Pr[z]$$

Pr[**f** | z] のモデルかは，特徴が数値ならガウス分布，カテゴリ値なら多項分布を使う

pLSAのさらなる拡張

ハイブリッド法：完全結合

[Basilico 04]

関数モデル：カーネルトリックの利用

利用者 x のアイテム y への好みの度合い　　重み

$$F(x,y) = \sum_{(x',y')\in(X,Y)} \alpha(x',y')K((x,y),(x',y'))$$

利用者とアイテムの全ての対についての和　　利用者-アイテム対の間のカーネル

カーネル：利用者カーネルとアイテムカーネルに分解

$$K((x,y),(x',y')) = K_X(x,x')K_Y(y,y')$$

利用者=アイテム対のカーネルは，利用者カーネルとアイテムカーネルの積

- 利用者カーネル：利用者 x と x' の間の相関係数など
- アイテムカーネル：アイテム y と y' の特徴の一致の度合いなど

推薦の提示

推薦の提示

適切な形式で推薦候補を利用者に提示

タスクに応じた推薦結果の表示

[Herlocker+ 04, Gunawardana+ 09]

適合アイテム発見

利用者の目的に適合すると予測した順にアイテムを提示

評価値予測

表示しているアイテムに予測評価値を参考出来るように表示

推薦の配送

[Ben Shafer+ 01]

推薦結果を利用者に配送する手続きの分類

Push：利用者がシステムを直接には使っていない場合に，利用者に推薦を届ける

- 電子メールやWebのバナー広告

Pull：利用者が明示的に要求したときに推薦リストを生成

- 最近では利用されなくなり，Passive が使われるように

Passive：利用者がシステムを利用中に推薦を添付

- 商品の検索結果に予測評価値を添付
- 閲覧中の商品に同時購入の可能性のある商品を推薦
- 積極的な推薦はできず，主導権は利用者にある

推薦アイテムの選別

予測評価の高いアイテムでも推薦すべきでないアイテム
ルールベースのフィルタリングにより必ず除外する

利用者に依存した条件

- **利用者が既知**：すでに提示したものや，過去に購入したもの
- **デモグラフィックな情報**：男性・女性向け，海外居住者には発送不可，年齢制限のあるアイテム
- **利用者が提示**：特定のカテゴリなど利用者が指定した条件でフィルタリング ← 利用者の満足に大きく影響 [Swearingen+ 01]

利用者に依存しない条件

- **アイテムに依存**：商品の在庫状況，色違いなど類似品の集約，違法アイテムの除外
- **コンテキストに依存**：季節商品，利用者の現在位置を考慮

推薦システムの評価

推薦システムの評価

[Herlocker+ 04]

オフラインテスト (Offline Test)

- 過去に蓄積した評価値・行動ログを用いたテスト
- 低コストなので大量のデータを評価できる
- 利用者は介在しないので，ユーザーインターフェースの評価は不可

実験室テスト (Laboratory Test)

- ユーザーインターフェースを備えたシステムを用いるが，被験者を対象とした評価
- 評価条件を統制できるので，仮説の検証ができる

実地テスト（Field Test; ABテスト）

- システムの実運用中に，実際の利用者を対象とした評価
- 実運用での評価に最も近い評価が可能
- 実際の利用者に悪影響を残す場合
  例：検索エンジンでの遅延の影響を調べたところ，元の状態に戻しても利用者の行動は完全には回復しなかった [Brutlag 09]

推薦システムの評価

[Herlocker+ 04]

	オフライン	実験室	実地
実験規模	😊	☹	☹
評価コスト	😊	☹	☹
実験条件の自由度	☹	😊	☹
実利用者への影響	😊	😊	☹
ユーザインターフェースの評価	☹	😊	😊
実運用での評価	☹	😅	😊

システムの運用

コスト・実験規模・利用者への影響から，運用までのテスト順序は

オフライン→実験室→実地

予測誤差

予測誤差：予測した嗜好と実際の嗜好のずれ

- 教師ありの機械学習手法では最も重視される
- 利用者の満足にとって最も重要な性能指標 [Swearingen+ 01]

低い予測誤差 ⟷ 計算が遅い

トレードオフ

予測精度は高い方が良いが，そのために問題が生じる
機械学習はこうした得失を考慮して設計

過学習と汎化誤差

コップとポヤングの嗜好はほぼ一致するが，**例外がある**

“コップヌードル” と “キチンラーメン” への嗜好から“やきそばポヤング” への嗜好を予測

過学習と汎化誤差

ルール A 例外的なデータにも細かく合わせた規則

コ＝😊 → ポ＝😊　　コ＝☹ → ポ＝☹

ただし　コ＝😊　かつ　キ＝☹ → ポ＝☹

ルール B 例外的なデータは無視した簡潔な規則

コ＝😊 → ポ＝😊　　コ＝☹ → ポ＝☹

学習に用いたデータでは → ルール A が良い

未知のテストデータでは → ルール B が良い

どちらのルールが良い？

過学習と汎化誤差

訓練用データ → 機械学習アルゴリズム → データの規則性 → 規則性を用いた予測 → 未知のデータ

未知のデータについて正確に予測できることが目標

↓

例外は無視して，訓練用データに合わせすぎない

- **過学習**：訓練用データに合わせすぎた状態
- **汎化誤差**：未知のデータでの，予測のずれ

過学習していない，汎化誤差の小さな規則を学習

交差確認

交差確認 (cross-validation)
学習に用いなかったテスト用データで，誤差をもとめ，汎化誤差の目安にする

データの分割方法

推薦システムでも利用される機械学習一般用のデータ分割法

k分割交差確認 (k-fold cross validation)

- データを k 個のブロックに分ける
- 最初は，第2〜第 k ブロックをまとめて学習用に，　第1ブロックをテスト用に利用し第1ブロックのデータに対して評価値を計算
- 以後，テスト用ブロックを第2〜第 k ブロックまで変えながら同様の処理を繰り返す
- 各ブロックごとに求めた評価値を集め，それらの平均値で評価

leave-one-out (一個抜き法)

- k 分割交差確認の k が，データ数に等しく，テスト用データが常に1個だけの場合

データの分割方法

[Breese+ 98]

推薦システム専用の，学習用とテスト用のデータ分割法

各利用者ごとに，その利用者が評価したアイテムを集める

- **Given x**：各利用者ごとに，x 個のアイテムを学習用，残りをテスト用に用いる．評価済みデータが少ない場合の性能を評価する場合．
- **AllButOne**：各利用者ごとに，1個をテスト用，残りを学習用に用いる．評価済みデータが多い場合の性能を評価する場合．

誤差の評価尺度：適合アイテム発見

データは対 (利用者, アイテム) や三つ組 (利用者, アイテム, 評価値)

正解率 (Accuracy)

$$\frac{\text{予測出来た適合データ数}}{\text{データの総数}}$$

- 適合した場合だけをうまく予測できたかどうかを評価
- 機械学習の予測分類問題でよく利用される

適合率/精度 (Precisioon)

$$\frac{\text{予測出来た適合データ数}}{\text{適合と予測したデータの総数}}$$

再現率 (Recall)

$$\frac{\text{予測出来た適合データ数}}{\text{適合データの総数}}$$

- 適合率は適合と予測したもののうち，正解であるものの割合
- 再現率は，適合されているもののうち，予測に成功したものの割合
- 情報検索で利用されている
- **F値**：適合率と再現率の調和平均

誤差の評価尺度：評価値予測

平均絶対誤差 (Mean Absolute Error; MAE)

$$\frac{1}{\text{データ総数}} \sum_i |\text{予測評価値}_i - \text{正解評価値}_i|$$

平均二乗誤差平方根 (Root Mean Average Error; RMSE)

$$\left[\frac{\sum_i (\text{予測評価値}_i - \text{正解評価値}_i)^2}{\text{データ総数}} \right]^{\frac{1}{2}}$$

- どちらも正解予測値と予測評価値が平均的にどれくらい離れているかを評価する
- MAEの方が，RMSEより大きくかけ離れた事例の影響を少なめに評価する
- 統計的予測・機械学習の回帰問題でもよく利用される

その他の問題や視点

プライバシ問題

推薦システムとプライバシ

[Ben Shafer+ 01]

プライバシー問題：利用者のプロファイルや嗜好データは個人情報

↓

良い推薦には，利用者の信頼を得て，これらの個人情報を集める必要

プライバシ問題への対策

プライバシーポリシー：個人情報の扱いを明示する

本当に遵守しているか? ➡ TRUSTeなどの監査会社

P3P(Platform for Privacy Preference)：

- 開示してもよい情報を知らせるプロトコル
- 利用者が制限していない情報のみ個人端末から送信される
- あまり普及していない

プライバシー問題

プライバシー問題：当初から問題は意識されてきた

利用者のプロファイルや嗜好データは利用者が
公開したくない個人情報

ジレンマ

高精度の推薦には
大量の嗜好データのデータが必要

- 推薦システムという語が広まるもとになった最初の文献ですでに指摘されている [Resnick+ 97]
- 推薦システムと利用者に関してアンケート調査し，利用者はプライバシを気にせずに入力してくれたとの報告も [Swearingen+ 01]

個人情報の種類

[Ben Schafer+ 01]

電子商取引サイトでの推薦システムが扱う個人情報

- **顧客が明示的に示す趣向の情報**
  商品へのレートやコメント，興味のある商品属性
- **顧客が暗黙的に示す趣向の情報**
  閲覧した商品や情報，閲覧時間，検索操作，カートに入れた商品，サイトの訪問
- **商品購入時の対話で得る情報**
  支払い方法，口座番号，発送先，発送先が購入した商品
- **明示的に示される識別情報**
  名前，住所，メールアドレス，電話番号
- **暗黙的に示される識別情報**
  閲覧しているコンピュータのIPアドレスやドメイン

プライバシポリシー

[Ben Schafer+ 01]

プライバシポリシー：

取引中に収集している情報とその使い道の説明の一覧

- 顧客の許可なく情報を販売しない条項は殆どのポリシーに含まれる

↓

- 他の企業との情報共有を妨げるので，推薦システムにとって重要
- ポリシーが厳しく，情報の収集が全くできなければ，推薦は不可能

↓

問題点

- 法律用語で書かれていて分かりにくく，作為的は変更の余地を残す
- 実際にポリシーが守られているかどうかわからない

↓

TRUSTeやBBBOnlineなどの監査会社によりサイトの信用を示すブランドを取得する

- TRUSTeを取得したReal.comが顧客情報をサーバに送信していた問題

P3P

P3P (Platform for Privacy Preference)

- WWWコンソーシアム(W3C)制定：http://www.w3.org/P3P/
- 開示してもよい情報を知らせるプロトコル
- エージェントとサイトの間で交渉し，エージェントは利用者に開示される情報を示すとともに，企業サイトも利用者の意向が分かる
- 何度も個人情報を入力する手間が省ける
- 利用者を識別する Pairise Unique IDentifier (PUID) はサイトごとに違い，サイト間での情報共有は困難となり，推薦システムには不利

↓

問題点

- 2006.11.13 の Final P3P 1.1 Working Group Note から更新なし
- 複雑で利用者はすぐには理解できない？
- 条件なしのオプトイン型だと，利用者は条件を厳しくしがち

外部データとの関連付けによる漏洩

[Frankowski+ 06]

推薦システムでの映画評価 ←対応付け→ 映画掲示板での感想の書き込み

同じ一群の映画について言及・評価している人は同一である可能性が高い

- あまり言及・評価されていない映画の重みを増やす工夫
- 一人が評価した映画は少ないので識別されやすい

- 一意に特定された利用者は全体の31%，5人のうちの一人として特定される利用者は44%
- 一意に識別される可能性をなくすには，88%ほどの映画に対する評価を消す必要

TiVoのプライバシ対策

TiVo：セットトップボックスでテレビの番組を推薦するシステム
ケーブルテレビでチャンネル数が多いアメリカでは普及

アイテム間型メモリベース法

推薦するときに必要な情報は？

自分の評価情報 → 自分のマシン内にあって公開不要

アイテムの類似性 → プライバシを保護して計算

プライバシ対策

- 送信されてきた評価データを永続的には保存しない
- 通信終了後には，タイムスタンプやIPアドレスなどの，どのクライアントからの送信だったかの情報を消去

プライバシ保護協調フィルタリング

大前提

個人の個々の嗜好データは個人情報

全体を集めた嗜好モデルは個人情報ではない

↓

個々の嗜好データを秘密にしたまま，全体の嗜好モデルが獲得できれば，プライバシを保護したまな協調フィルタリングが実現できる

暗号化を使う方法

- 個人の嗜好データを暗号化し，暗号化したままモデルを計算する

ランダム化を使う方法

- 個人の嗜好データにランダムなノイズを付加することで，元のデータを隠蔽する

敵対者モデル

semi-honest：プロトコルの実行中に自身が得た情報や，通信路上の情報から，他の参加者の秘密情報を知ろうとするが，プロトコル自体を逸脱することはない

- malicious な場合よりずっと効率的なプロトコルの実行が可能

malicious：他の参加者の秘密情報を知るために，プロトコルを逸脱したメッセージを送ることもありうる

- semi-honestを前提としたプロトコルは，ゼロ知識証明との組み合わせで，maliciousなモデルにも対応させることが可能

rational：効用関数に基づいて、自分の利益になる場合には semi-honest に振る舞うが、そうでなければ malicious に振る舞う

- 現実的な場面を想定した，新しい敵対者モデル
- 大幅な計算量・通信量の削減が可能

プライバシ保護協調フィルタリング
暗号化による方法

[Canny 02, Canny 02]

嗜好データは暗号化してから外部に出す

MPCと秘密関数計算

[佐久間+ 09]

YaoのMPC (Secure Multi-Party Computation)

[Yao 86]

複数の参加者が保持する秘密の入力に対する多項式時間の任意の計算を第三者の補助なしに，安全に実行できるようにする

↓

原理的には，個人情報を保護して嗜好モデルを獲得可能

↓

計算量が非常に大きく非実用的

準同型性暗号による秘密関数計算 (Secure Function Evaluation)

暗号化関数 Enc(・) と平文 m_1 や m_2 が次式を満たす [Aggarwal+ 08]

- **加法準同型性**：暗号文同士の加算 … 修正ElGamal，Pallier，楕円

  $\mathrm{Enc}(m_1)\cdot\mathrm{Enc}(m_2) = \mathrm{Enc}(m_1 + m_2)$，$\mathrm{Enc}(m_1)^{m_2}=\mathrm{Enc}(m_1\ m_2)$

- **乗法準同型性**：暗号文と平文の積 … RSA，ElGamal

  $\mathrm{Enc}(m_1)\cdot\mathrm{Enc}(m_2) = \mathrm{Enc}(m_1\ m_2)$

秘密総和計算 (Secure Sum)

[Aggarwal+ 08]

A B C の3人が，それぞれ，秘密の値 S_A S_B S_C を保持
S_A S_B S_C を他人に明かさず総和 $S = S_A + S_B + S_C$ を計算

① A は鍵を作って公開鍵を配布

※ 加法準同型性のある暗号を使う

② 暗号化してAに集める

$$Dec_A(Enc_A(S_A)\cdot Enc_A(S_B)\cdot Enc_A(S_C)) = Dec_A(Enc_A(S_A+S_B+S_C)) = S$$

③ A は総和を計算して復号し全員に総和を配布

※ この方法は通信を傍受可能な場合やプロトコル違反がある場合には安全ではない

厳密な値が計算できるが，共謀やプロトコル逸脱には脆弱

準同型暗号を使った行列分解

行列分解での学習はだいたい次の最適化問題になる

$$f = \|S - UV\|_F \rightarrow \min$$

正解のスコア　利用者行列　アイテム行列

※ U の行ベクトル $\mathbf{u}_i$ と，V の列ベクトル $\mathbf{v}_j$

簡単に勾配降下で最適化・現在の $\mathbf{u}_i$ と $\mathbf{v}_j$ は利用者間で共有している

$$\frac{\partial f}{\partial \mathbf{u}_i} = -\sum_j (s_{ij} - \mathbf{u}_i \mathbf{v}_j)\mathbf{v}_j$$

$$\frac{\partial f}{\partial \mathbf{v}_j} = -\sum_i (s_{ij} - \mathbf{u}_i \mathbf{v}_j)\mathbf{u}_i$$

利用者 i の情報だけがあれば，自分の $\mathbf{u}_i$ を更新可能

全利用者の総和が必要

加算準同型暗号を利用

勾配を秘密に計算できる！

プライバシ保護協調フィルタリング
ランダム化による方法

[Polat+ 03, McSherry+ 09]

嗜好データは乱数を加えてから外部に出す

ランダム化

A B C の3人が，それぞれ，秘密の値 S_A S_B S_C を保持
S_A S_B S_C を他人に明かさず総和 $S = S_A + S_B + S_C$ を計算

$N(0, \sigma)$ は，平均が 0 で，標準偏差が σ の乱数

$R_A \sim N(0, \sigma)$
$V_A = S_A + R_A$

$R_A \sim N(0, \sigma)$
$V_A = S_A + R_A$

$R_A \sim N(0, \sigma)$
$V_A = S_A + R_A$

$$V_A + V_B + V_C = (S_A + S_B + S_C) + (R_A + R_B + R_C) \rightarrow (S_A + S_B + S_C) = S$$

漸近的に0

共謀やプロトコル逸脱には強いが，近似計算しかできない

加算ノイズの利用

[Polat+ 03]

加算ノイズによるプライバシ保護

真値 a_i に 0 平均で同分布に従うノイズ ε を加える $a'_i = a_i + \varepsilon$

内積 $\displaystyle\sum_{i=1}^{n} a'_i b'_i \xrightarrow{n\to\infty} \sum_{i=1}^{n} a_i b_i$

総和 $\displaystyle\sum_{i=1}^{n} a'_i \xrightarrow{n\to\infty} \sum_{i=1}^{n} a_i$

ノイズを加えた値しか与えらなくても内積や総和は近似的に計算可能

利用者 i はアイテム j の評価値を
標準化しノイズを加えて公開

$$z_{ij} = \frac{s_{ij} - \bar{s}_i}{\sigma_i} + \epsilon$$

利用者間型メモリベース法

類似した利用者の嗜好を反映した評価

利用者間の類似性

真値に収束

$$\hat{s}_{aj} = \bar{s}_a + \sigma_a \frac{\sum_i^n \left(\sum_k z_{ak} z_{ik}\right) z_{ij}}{\sum_i^n \left(\sum_k z_{ak} z_{ik}\right)} = \bar{s}_a + \sigma_a \frac{\sum_k z_{ak} \left(\sum_i^n z_{ik} z_{ij}\right)}{\sum_k z_{ak} \left(\sum_i^n z_{ik}\right)}$$

予測評価値

標準化の逆変換

評価アイテム数の違いを補正

自分の値

差分プライバシ

[Aggarwal+ 08]

差分プライバシ (Differential Privacy)

ある評価値が一つあってもなくても，ランダム関数によるノイズ付加後に観測される値の分布に見られる変化の差は制限されている

ランダム関数 $M(\cdot)$ に対し，ある評価値が一つだけ異なる集合の対 A と B および，M の値域の任意の部分集合 S について次式が成立

$$\Pr[M(A) \in S] \leq \exp(\epsilon) \Pr[M(B) \in S]$$

出力集合 S から，A と B の差を推定するとき，P[A]/P[B] を制限

$$\frac{\Pr[A|S]}{\Pr[B|S]} = \frac{\Pr[A]}{\Pr[B]} \frac{\Pr[S|A]}{\Pr[S|B]}$$

exp(ε)以下

※ 出力の性質ではなく，こうした計算手続きの性質が差分プライバシを満たすという

近似差分プライバシ (Approximate Differential Privacy)

制限を次式のように緩和する

$$\Pr[M(A) \in S] \leq \exp(\epsilon) \Pr[M(B) \in S] + \delta$$

差分プライバシの実現

$A \approx B$ ：集合 A と B は要素が一つだけ異なる
$f(X)$ ：d 次元実数を出力する関数

次の条件を満たす $M(X)$ は ε -差分プライバシを満たす

ランダム化関数：$M(X) = f(X) + (\mathrm{Laplace}(0, \sigma))^d$

$$\sigma \geq \frac{1}{\epsilon} \max_{A \approx B} \|f(A) - f(B)\|_1$$

次の条件を満たす $M(X)$ は(ε , δ)-差分プライバシを満たす

ランダム化関数：$M(X) = f(X) + (\mathcal{N}(0, \sigma^2))^d$

$$\sigma \geq \frac{1}{\epsilon} \sqrt{2 \ln(2/\delta)} \max_{A \approx B} \|f(A) - f(B)\|_2$$

※例えば，要素の上限が∞である場合には，関数の差を有界にできないので，総和について差分プライバシを満たす手続きは不可能

差分プライバシを実現する協調フィルタリング

[McSherry+ 09]

アイテムごとや利用者ごとの平均，アイテム間の共分散が推薦には必要

個人の嗜好の値が 0/1 だとすると…

スコアの度数：平均の計算などに必要

$f(A)$：集合 A の要素を d 個のビンに分けた度数

$$\max_{A \approx B} \|f(A) - f(B)\| = 1 \quad \text{が成立}$$

総和 一つのレコードの総和が高々 R ならば

$$\max_{A \approx B} \|f(A) - f(B)\| \leq R \quad \text{が成立}$$

アイテム間共分散

利用者 u の評価値が $r_u^{(a)}$ から $r_u^{(b)}$ に変化した場合

変化したとき $\|r_u^{(b)} - r_u^{(b)}\| = 1$ を満たすなら

$$\|\mathrm{Cov}^{(a)} - \mathrm{Cov}^{(b)}\| \leq \|r_u^{(b)}\| + \|r_u^{(b)}\|$$

サクラ攻撃

サクラ攻撃

[Mehta+ 08, Lam+ 04]

推薦システムの導入で…

利用者

欲しいものが
見つかった

システム運営者

利用者が満足
サイトが人気

コンテンツ提供者
製品のメーカー

不満

ライバル商品を推薦
自分の商品が推薦されない

サクラ攻撃 (shilling attack)

推薦結果に介入する意図でシステムに干渉

サクラ攻撃の実例

[Lam+ 04]

- 2001年6月にソニーピクチャーズが，実在しない批評家による批評を，新作映画の宣伝に使った
  http://news.bbc.co.uk/1/hi/entertainment/film/1368666.stm

- Amazon.comで発見された作為的なアドバイス
  http://www.wired.com/news/ebiz/0,1272,53634,00.html
  http://news.com.com/2100-1023-976435.html

- eBayにて出品者への評価の売買
  http://www.auctionbytes.com/cab/abn/y03/m09/i17/s01

- digg票の売買とブログ贈賄が始まった
  http://longtailworld.blogspot.com/2006/10/diggdiggs-vote-buying-and-payperpost.html

サクラ攻撃の基本

- **サクラボット(shilling bot)**
  偽評価を書き込むソフトウェアエージェント
- **標的アイテム (target item)**
  評価を操作したいアイテム
- **攻撃目的の分類**
  - **販促攻撃 (push attack)**
    標的アイテムに良い評価を与えて，より推薦されるようにする
  - **排除攻撃 (nuke attack)**
    標的アイテムに悪い評価を与えて，推薦されないようにする
  - **攪乱攻撃**
    推薦の質を下げて，システムへの信頼を低下させる

サクラ攻撃の方法

協調フィルタリングでは**類似した利用者**の評価を利用

※ 排除攻撃の場合の例

標的アイテムだけに悪い評価をしても，通常利用者のプロファイルと似ていないので無効

標的アイテム以外のアイテムも評価して，通常利用者のプロファイルと類似させる

サクラ攻撃の戦略

詰め物アイテムへの評価値による戦略の違い

- **ランダム攻撃** (random attack) [Lam+ 04]
  ランダムな評価値．あまり効かず，発覚しやすい．
- **人気商品攻撃** (bandwagon attack) [Burke+ 06]
  ランダム攻撃の改良型．人気商品には良い評価を，その他は平均攻撃と同じ
- **平均攻撃** (average attack) [Lam+ 04]
  そのアイテムの平均評価を中心とした評価値を与える．利用者間型に有効
- **セグメント攻撃** (segment attack) [Burke+ 05]
  標的アイテムの類似商品に良い評価を与える．アイテム間型に有効．
- **愛憎攻撃** (love/hate attack) [Burke+ 06]
  排除攻撃専用．標的アイテムには最低の，詰め物には最高の評価を与える

サクラ攻撃への対策

利用者の評価値は，表示されている予測評価値に影響される

[Cosley+ 03]

↓

サクラ攻撃は検出して排除しなくてはならない

検定や異常検出技術の利用

- 予測される評価値と与えられた評価値の差
- アイテムへの平均評価からの乖離
- 攻撃は多数のアイテムに行われることが多いので，利用者の評価数

発生頻度が少ない，これらの統計量が検出されたら，攻撃とみなす

攻撃は時間的にまとめて実行されやすい

[Zhang+ 06]

↓

評価値が異常な時間窓を検出

Webコンテンツ最適化

Webコンテンツ最適化/オンライン推薦

[Abe+ 99, Agarwal+ 09, Li+ 10]

Webコンテンツ最適化 (Web contents optimization)

Webコンテンツ
(バナー広告やニュース記事など)
利用者が関心を持ち，頻繁にアクセスされるほど利用者満足や収益は向上

候補コンテンツから，アクセスされる確率が最大のコンテンツ群を選び出す

- **スケーラビリティ**：表示の遅延なしに，大量のリクエストを処理
- **動的変化**：利用者の嗜好の変化や，候補コンテンツの入れ替え
- **契約上の制約**：バナー広告には最低表示回数などの契約条件
- **フィードバック遅延**：利用者がコンテンツを見るまでの時間遅延

多腕バンディット

多腕バンディット問題

(multi-armed bandit problem)

バンディット（スロットマシーン）が何台かある．T回チャンスがあるとき，一番儲かるようにするには，どれをどの順で引けばよい？

当たり確率が最大のものにかけ続ければよい

↓

当たり確率は不明なので，推定する必要

活用-探索のジレンマ

- **活用 (exploitation)**：当たる可能性を最大化するには，現在のところ一番当たり確率の高いマシンに賭けた方がよい
- **探索 (exploration)**：現在のところ当たり確率が高いマシンは，偶然当たりが続いただけかもしれないので，他のマシンも調べる必要

UCB1

[Auer+ 02]

リグレット：一番当たる確率が高いスロットマシーンを引き続けた場合に得られる金額から，実際やってみた場合に取りそびれた金額

Upper Confidence Bound 1 (UCB1)

ある決まった回数　T　だけスロットマシーンを引き続けたとき，リグレットを小さくするようにするアルゴリズム
マシンを引く前に，全てのマシンについて次のスコアを計算して，一番スコアの高いマシン j を引くと良い

j番目のマシンが当たった回数

j番目のマシンを引いた回数

$$\frac{n_j^+}{n_j} + \sqrt{\frac{\boxed{2\ln n}}{n_j}}$$

実用的にはこの定数は調整して性能を向上

全マシンを引いた回数

確率 ε で非最適なマシンを引く ε-Greedy のリグレットは O(T)
それに対して，UCB1 は O(ln T) のリグレットを達成できる

ロングテール
人気バイアスとバンドワゴン効果

推薦システムとロングテール

ロングテール

販売数量の少ない分布の裾(tail)の部分の商品でも，多品種を販売したときの売上げは，販売数量の多い商品群に対するそれに匹敵する

推薦システムはロングテールを加速する？

YES

推薦システムや閲覧ツールによって，顧客はニッチな商品により容易にアクセスできるようになり，ロングテール戦略に有利 [Brynjolfsson 06]

NO

顧客が事前には特定の商品を好むが，それを推薦で変更する可能性が非常に高い場合以外はメインストリームの寡占を推薦システムは強化するとモデルで分析 [Fleder+ 07]

人気バイアスとバンドワゴン効果

人気商品は，推薦システムの介在でさらに人気化しやすい

人気バイアス (popularity bias)

[Celma+ 08]

推薦システムには人気のあるアイテムを，より推薦しやすい傾向

- メモリベース法では，他のアイテムの近傍や，近傍にいる利用者が評価している可能性が高い
- モデルベース法では，訓練データ中に多数あるアイテムの誤差を重視

バンドワゴン効果 (bandwagon effect)

[Sunder+ 08]

多くの人が良いと思ったものを良いと思う傾向が人間にはある

- 編集者，自分自身，多数の人の選択，コンピュータが選んだニュース
- 同じ記事であっても，多数の人が選んだものであると言われたときは，編集者や自分自身が選択したときより良い記事と感じた
- ニュースとしての価値も，自身が選んだものよりあると感じた

ロングテールの活性化は可能か？

[Levy+ 10]

Last.fm の実験サービスを使ったロングテール部分の喚起

ユーザの全体傾向の分析

- Last.fm ユーザと非ユーザでは，人気音楽への好みの有意差はない
- ライトとヘビーユーザでは，後者が若干ロングテール音楽を好む傾向

ロングテール強制推薦

- インディーズでさえないバンドの音楽しか推薦しないことで，ロングテールへの注意を喚起
- グラフが左の方が人気音楽重視
- ロングテール推薦により，ロングテールを若干だが注目させることは可能

フィルターバブル問題

フィルターバブル問題

[TED Talk by Eli Pariser]

フィルターバブル問題

個人化技術の普及により，ネット利用者が多様な情報源や視点に接する機会が減らされているとの Eli Pariser による主張

http://www.thefilterbubble.com/

リベラル派の友人を増やしていると，私に断りなく
FaceBook の友人推薦リストから保守派が消されてしまった

フィルターバブル問題

[TED Talk by Eli Pariser]

- Google の個人化された検索結果をいろいろな人に頼んで送ってもらった
- Scott と Daniel の結果は明らかに異なる

- Egypt で検索したところ，民主化デモなどがあった時期にも関わらず，一方の検索結果には通常の観光情報が主であった

フィルターバブルの議論

[RecSys 2011 Panel on the Filter Bubble]

フィルタバブルは実在しているか？

- 個人化の分野では，経験範囲を狭めうることは1990中期から指摘
- 全ての可能性の網羅は本質的に不可能で，これはトレードオフ問題

フィルターバブル問題はどれほど重大か？

- 唯一のビッグブラザーが特定の情報を選択しているのではない
- 人によって異なる情報を受け取るのは，社会の多様性には良いこと
- 既に信じていることを強化する懸念 → 結果はまちまち

この問題に対する対処は？

- どういう基準で選択しているのかという透明性・説明を確保
- 単純な精度ではなく，長期的視点・推薦リスト全体の精度を追求
- その人が求めた殻であるなら，それに干渉すべきではないのでは？

http://acmrecsys.wordpress.com/2011/10/25/panel-on-the-filter-bubble/

推薦理由

推薦の透明性

[Sinha+ 02]

推薦そのものだけでなく推薦の理由も提示すると，利用者が推薦を信用し受け入れやすくする

実験条件

- 五つの音楽推薦サイトで実験
- 被験者は日常的に音楽推薦サイトを利用し，被験者数は12人
- 各サイトごとに被験者は10回の推薦について好み，透明性を評価
- 透明性は「なぜシステムがこのアイテムをあなたへ推薦したかを理解していますか？」という質問

↓

- 透明性がある場合の方が，アイテムを好む割合は有意に多かった
- 知らなかった，以前から好き，以前から嫌いの三つの場合で，知らなかったり，以前から好きな場合には，透明性がある場合に推薦されたアイテムを好む割合が有意に増えた

推薦理由の提示：ホワイトボックス

ホワイトボックス：推薦のプロセス自体を見せる

2007/7/26 にスクリーンショットを取得

例1：アイテム間型近傍法で，過去に購入したアイテムと類似していることを明示する

[Linden+ 03]

例2：利用者間型メモリベース法で，嗜好が似ている人の評価を示す

[Herlocker+ 00]

推薦理由の提示：ブラックボックス

[Herlocker+ 00]

ブラックボックス：推薦のプロセスは利用者に意識させない

推薦の強さと共に，推薦の確かさを提示

簡潔だが，推薦の透明性を高めるために有効

- **推薦の強さ：**利用者がどれくらい強くアイテムを好むと推薦システムは予測しているか
  - **例**：5 段階評価で 4 ぐらい
- **推薦の確かさ：**利用者に示した推薦を，システムはどれくらい確かだと考えているか
  - **例**：過去の同様の推薦では，利用者の実際の嗜好と一致した割合は80%

推薦の多様性

PRPの呪縛

PRP (Probability Ranking Principle) [Robertson 77, Belew 00]

情報検索分野での古典的な原理

クエリへの適合確率で成立した回答リスト → あらゆる回答の中で利用者にとって最も効果的

↓

これは推薦システムでも真理？？？

利用者の情報要求は不明確，もしくは存在しない
情報検索とは違った，意志決定支援としての側面

- **cherry picking**：容易に分かるアイテムのみを推薦して予測精度を「かせぐ」のは安易 [Herlocker+ 04]
- **収穫逓減の法則 (law of diminishing marginal returns)**：類似したものを続けて与えられると効用は低下 [Ziegler+ 05]

推薦の多様性

アイテムごとに好き嫌いを判断し
それらを集めてリストにする

↓

推薦はリスト全体をまとめて考慮すべき [McNee+ 06]

リスト中に含めるアイテムの選択によって

正解率 ⟷ 多様性

バランス

例：いろいろな利用者に対して一連の推薦をする場合を想定．予測嗜好スコアがしきい値の以上アイテムから，できるだけ異なるアイテムが推薦されるようにリストを作る

個別多様性

個別多様性 (individual diversity)
一度の推薦で同時に提示するアイテムリスト内の多様性

話題の多様化

話題の多様化 (Topic Diversification)

[Ziegler+ 05]

予測精度の低下と引き替えに，より多様なタイプのアイテムが推薦アイテムリストの中に含まれるようにする

類似度関数 c：アイテムの内容に基づいてアイテム間の類似度を測る
アイテムの階層的な分類があるとき，その構造木中でのホップ数の逆数などで測る

z-1 位まで決まった推薦リストに z 番目を追加する方法

1. 候補アイテムを，利用者の嗜好スコアの順に整列したリスト➡ P_w
2. 候補アイテムを，z-1位までの推薦リストへの類似度 c の平均の逆順に整列したリスト➡P_c
3. 両リスト中の候補アイテム b の順位を $P_w(b)$ と $P_c(b)$ として加重順位 $(1-\theta)\,P_w(b) + \theta\,P_c(b)$ が最小のものを推薦リストに追加する

ソーシャル推薦

ソーシャル推薦

利用者間メモリベース法

嗜好が似ている他の利用者が好きなものを推薦

↓

嗜好共起性で繋がった利用者のネットワークも利用できる

- Web上での共起性に基づくネットワーク [Kautz 97]
- 過去の推薦での有用性に基づくネットワーク [O'Donovan 05]
- 同じカテゴリの商品を購入しているネットワーク [Cho 07]

↓

明示的ソーシャルネットワークの登場

- レビューサイトで，他人の評価を評価したネットワーク (epinion)
- 狭義の SNS の友人関係ネットワークの利用 (mixi, FaceBook)

友人や信頼できる人の嗜好をより重視した推薦

ソーシャル推薦

[Ma+ 08]

確率モデルの拡張

評価値と同時に，ソーシャルネットの関係の予測誤差も考慮

グラフィカルモデル

目的関数

アイテム評価値

$$\frac{1}{2}\sum_{(x,y)}(r_{xy} - \sigma(\mathbf{u}_x^\top \mathbf{v}_y))^2$$

ソーシャル関係

$$+\frac{\lambda_{\mathbf{Q}}}{2}\sum_{(x,i)}(q_{xi} - \sigma(\mathbf{u}_x^\top \mathbf{w}_i))^2$$

シグモイド関数

$$+\frac{\lambda}{2}\text{sq-reg-term}$$

正則化項

パラメータ $\mathbf{u}_x$，$\mathbf{v}_y$，$\mathbf{w}_i$ を最適化

入出力の多様化

入出力の多様化

内容ベースフィルタリングの弱点

アイテムのデータベース構築は高コスト

- DBが整備されているのは書籍・CDなどの限られた品目
- アイテムの情報を提示すると意志決定にとって重要 [Swearingen 01]
- 高コストのDB整備には，それに見合う規模の店舗できない
- DBにない商品はないも同然 → DBの寡占が流通の寡占に

↓

推薦のためのデータベースを抵コストで構築する技術

- **入力情報の整備**：アイテム情報の Linked Data や Webサービス による共有，嗜好のゆらぎの補正，評価基準の多様化
- **入力情報の高度化**：自然言語・画像・音声・音楽情報処理で抽出

データベース情報の収集

集中的に大規模データベースを作る

製品情報の掲載や修正にタイムラグ
情報の寡占による流通の支配

↓

分散管理と凝集サービス

メーカー・サービス事業者

- 自社の製品・サービスの情報を linked data や Webサービスとして提供

データ加工・提供

- 各種のデータベースを収集して，Webサービスで効率的に提供するサービス
- 目的に合わせて語彙の変更や同義語の凝集などを行う

自然言語・画像・音楽情報処理

自然言語処理

- 評判情報分析 (sentiment分析)
  - 利用者のアイテムへの嗜好，アイテムaspectの相対的優位性
- 固有表現抽出，情報抽出
  - アイテムの特徴データベースをテキスト情報から構築

画像処理

- 画像認識技術
  - 利用者の画像から性別や服飾の好みの情報を抽出
  - 画像から屋内外・天気などの利用状況の情報を抽出
  - 色調・大きさなどのアイテム特徴の抽出

音楽情報処理

音楽の曲調などの特徴を，波長データから抽出

入出力の多様化

入力デバイスの多様化

画像処理で環境や対象の識別に使われれている特徴量を推薦の生成に利用

自然言語処理と画像処理との協調

高さ・奥行き・幅との対応付け

出力手法の高度化

推薦の提示：★の数やアイテムの順位付けリスト

魅力的な提案ではない

↓

魅力的な推薦の提示は意志決定に影響 [Swearingen+ 01, Cooke+ 02, Senecal+ 04, Sinha+ 02]

- 利用者からのフィードバックをより活用したARや電子掲示板
- 小売店だけでなく，メーカーも直接顧客にアクセス

→ 流通の流れが変わるかも？

グループ推薦

[O'Connor 01, Baltrunas 10, Berkovsky 10, Seko 11]

グループ推薦 (group recommendation)
個人ではなく，グループに対する推薦．メンバー間で嗜好の一致・不一致があるためそのバランスをどうするかが課題

映画・TV番組など，旅行プラン，レストラン・料理レシピなどが対象

スコアに基づく方法：各アイテムのメンバー個人へのスコアから，グループ全体への効用を決める関数を定義し，その値の良いものを推薦

- メンバーごとの重要性で重み付けしたスコア平均値
- グループ内で一番スコアが低いメンバーのスコアを最大にする
- 前回の行動で，我慢したメンバーを重視する
- メンバーの好むジャンルを考慮する

推薦リストに基づく方法：各メンバーごとに推薦リストを作り，そのリストをまとめる方法

- 情報検索で用いられる順位凝集 (rank aggregation) の技術を利用

マルチ評価基準

[Adomavicius 07]

マルチ評価基準 (multi-criteria)
アイテムの評価基準が全体としての嗜好ではなく，そのアイテムのいろいろな側面についての評価が与えられる場合

Zagatのレストランガイドは，食事，内装，サービスなどの複数の規準で評価

全評価基準を考慮した評価スコア：全ての評価基準のスコアを入力として全体としての嗜好を予測するスコア関数を定義し，その全体スコアに基づいて予測

- 利用者間のメモリベース法で，全ての評価基準を考慮した利用者間の類似性尺度を定義

各評価基準ごとに求めたスコアをまとめる：各評価基準を用いて個別にスコアや推薦リストを作り，その後，個々のスコアや推薦リストをまとめる

コンテキスト配慮型推薦

コンテキスト配慮型 (context-aware recommendation)
推薦システムが利用される状況に応じて，推薦内容を適応させる推薦

- 映画：一人で見に行く，家族で見に行く，カップルで見に行く
- レストラン：毎日の食事と記念日の食事，地理的な近さ

各コンテキストごとに推薦モデルを構築

↓

疎な嗜好データがさらに疎になるため難しい

↓

コンテキストも，アイテムや利用者の特徴と同様に特徴量ととらえ，
内容ベース型やハイブリッド型の推薦アルゴリズムを利用する

価格個人化推薦システム

[神嶌 10, kamishima 11]

価格差別の個人化

価格意識がある ＋ アイテムへの嗜好

↓

もし割引があったなら購入する顧客にのみ割引価格を提示

利用者は割引価格で購入でき，提供者は追加売上げ

バンディットを利用した割引対象の予測

割引提示による追加利益の確保 vs 新規割引対象顧客の探索

- 価格意識があるかどうかの予測，適切な割引率の設定
- 全体の損益を考慮した安全性モデル，独禁法への対処
- 転売目的で購入する顧客の予測
- 価格より価値・納期に意識のある利用者にはそうした提示

セット推薦

共起性のあるアイテム群

協調的なアイテム群

デジカメと充電池

対立的なアイテム群

デジカメAとデジカメB

アイテムのセット推薦

- 協調的なアイテム群であるかの予測
- セット販売のみの割引き価格の設定

共起性以外の観点からのアイテムのセット化

- 嗜好パターンが類似している利用者の好むアイテム群と，アイテム自体の特徴を考慮して(シャツとジャケット，柄+柄はダメ)，衣料品のコーディネート
- 旅行先でのレストラン・レジャーのセット

対話型システム

フィードバックを導入し対話的にする

アイテムレベルのフィードバック

推薦後に，推薦リスト中やシステムが指定したアイテムについて，その評価値を利用者が返す

- より予測精度を向上させるのに役立つアイテムをシステムが選択して，利用者に評価させる能動学習 [Kohrs+ 01,Boutilier+ 03, Rubens+ 07]
- セッション中での評価を短期プロファイルとする．これを重視することで，コンテキストに依存した推薦をする [Rafter+ 05]
- 前回利用者が選んだアイテムを推薦リストに混ぜて，それが選ばれたら推薦の改善方向が誤っていたと考える [Smyth+ 03]

特徴レベルのフィードバック

クエリを使った内容ベースフィルタリングで用いる
特徴の値について「より大きく」といった制約を入力させる

[Smyth+ 04, McSherry 05]

技術課題

推薦システムの歴史

- **1980年代**：情報フィルタリング研究やUSの特許 [芳賀 08]
  - System and Method for Predicting Subjective Reactions(1987) US Patent 4870579
  - System and Method for Recommending Items(1989) US Patent 4996642
- **1992**：協調フィルタリング (collaborative filtering)の用語 [Goldberg+ 92]．他の利用者による人手による推薦.
- **1994**：協調フィルタリングの自動化．GroupLens [Resnick+ 94] や Ringo [Shardanand+ 95]．現在の推薦システムの基礎.
- **1990年代中期**：情報フィルタリングやメモリベース推論の，内容ベースフィルタリング (content-based filtering) への変化.
- **1996**：専門ワークショップの開催
- **1997**：ACM Communications誌の特集．推薦システム (recommender system) の呼称の定着 [Resnick+ 97].
- **1990年代後期**：NetPerceptions や Firefly などによる商業化
- **現在**：Web上のサービスやセットトップボックスでの利用

推薦システムの研究動向

技術課題

- **データ入力：** **入力情報の整備**（アイテム情報のデータベース・Webサービス化，嗜好のゆらぎの補正，評価基準の多様化）**入力情報の高度化**（自然言語・画像・音声・音楽情報処理で抽出）

- **嗜好の予測：** **start-up問題**，**利用者・アイテムの変化への対処**（オンライン学習）**セレンディピティなどへの配慮**，**推薦リスト全体の最適化**，**スケーラビリティ**（オンライン学習，データベース参照の効率化，並列分散処理）**マルチ評価基準**，**グループ推薦**，**ソーシャル推薦**

- **推薦の提示：** **推薦の提示環境**（街頭の案内板・掲示板，拡張現実の利用）**対話型のインターフェース**（能動学習，learning to rank)

- **その他：プライバシー保護**（プライバシ保護データマイニング，プライバシ=効用のトレードオフ，法整備）**サクラ攻撃**（提供者側から見たサクラ攻撃検出，利用者によるサクラ攻撃の検出）

推薦システムは役に立つのか？

利用者にとって推薦システムは役立つ？

↓

もちろん!

ただし，次の条件が必要！

推薦システム側		利用者側
利用者が欲しいと予測したもの	＝	利用者が欲しいもの

推薦システムと利用者要求の一致

推薦システム側		利用者側
利用者が欲しいと予測したもの	=	利用者が欲しいもの

利用者が推薦システムに合わせる

- **利用者主導の推薦**：推薦の方向付けを利用者が行う
- **推薦結果の選択**：システムが開示した情報から推薦を取捨選択
- **推薦状況の限定**：利用者が情報要求が自明な状況で利用

推薦システムが利用者に合わせる

- **入出力の多様化**：評価値以外の情報をもっと活用
- **推薦内容の多様化**：正解率以外の視点も考慮，より積極的な提案

利用者が推薦システムに合わせる

- **利用者主導の推薦**
  利用者が望む推薦の方向付けを事前に入力
  - 自身の関心をプロファイルに事前登録して推薦を制約する機能は利用者の満足に大きく影響 [Swearingen+ 01]
- **推薦結果の選択**
  システムが提供する情報に基づいて，利用者が推薦を取捨選択
  - 推薦の根拠に関する説明を，利用者に提示し，利用者はそれに基づいて，推薦を受理するかどうかを判断する [Herlocker+ 00, Sinha+ 02]
  - 統計的予測や店員の推薦といった推薦プロセスを明示 [Senecal+ 04]
- **利用状況の限定**
  利用者の情報要求が自明な状況で利用してもらう
  - 駅の案内板，店頭の商品案内など利用者の要求が限定されている状況で利用されるようにする

推薦システムが利用者に合わせる

入出力の多様化

- デモグラフィック・アイテムの特徴・利用状況なども考慮したハイブリッド推薦アルゴリズム
  - 画像・ 音楽・ テキスト処理など高度なアイテム特徴抽出技法
  - 利用時間・場所，アイテムの在庫状況なども考慮
  - DB中のアイテムデータだけでなく，入力された情報や画像などと組み合わせた魅力的な提案

推薦内容の多様化

[McNee+ 06, McNee+ 06]

- 予測精度は最も重視すべきだが，それだけでは不十分
  - アイテムの特性や利用者の目的に応じて，予測正解率，被覆率，セレンディピティなどの推薦における重点指標を変更
  - 利用者の反応や，フィードバックに基づいて推薦リストを変更

予測精度とのトレードオフ

セレンディピティ：意外性のある推薦

だれもが確実に好むアイテムを推薦すると正解率は向上

↓

推薦がありきたりになって推薦システムを使う意義がない

被覆率：推薦の対象にできるアイテムの割合

予測が簡単なアイテムについてのみ予測をすると正解率は向上

↓

利用者が本当に好むアイテムが推薦の対象にならない

セレンディピティや被覆率は，予測精度とトレードオフの関係

セレンディピティと予測精度

有用性の影響

[Adomavicius+ 05]

当たらない推薦が無意味なのは当然だが
「分かりきった」推薦では利用者には有用ではない

例：スーパーマーケットの客に，牛乳や卵などの日常的に購入するものを推薦するのは有用な推薦とはいえない

予測精度が高くても有用性は向上するとは限らない

アイテムの性質

[Senecal+ 04]

- **searchアイテム**：スペックなどから洞察で意志決定できる
  白物家電など ➡ **スペックから判断でき，推薦には依存しない**
- **experienceアイテム**：試さないと良いかどうかわからない
  嗜好品など ➡ **推薦に依存し，セレンディピティが重要**

推薦対象の性質によってセレンディピティの重要性は変わる

セレンディピティと予測精度

[Swearingen 01]

アイテムへの知識量に基づく利用目的の分類

備忘録

- 既知のアイテムを思い出させる

類似品

- 既知のアイテムに類似したアイテムを，比較などのために探す

新規アイテム

- 自分が確実に好むであろう，新製品などを探す．ジャンルなどは，既知のアイテムなどと共通．

視野を広げる

- 他のジャンルにも自分の関心を広げるため，やや興味の薄いアイテムも含める

上の方ほど正解率を重視，下の方ほどセレンディピティ重視

被覆率と予測精度

被覆率（推薦の対象にできるアイテムの割合）のトレードオフ

利用者間型メモリベース法 近隣の利用者が未評価のアイテムは対象外

非常に類似した標本利用者の評価のみに基づく
高予測精度
低被覆率

⟷

比較的に低類似度の標本利用者の評価も参考にする
低予測精度
高被覆率

推薦タスクの影響

[Herlocker+ 04]

適合アイテム発見 ｜ 評価値予測 ｜ 適合アイテム列挙

正解率重視 ⟷ **被覆率重視**

利用目的によって被覆率の重要性は変わる

提供者と利用者のメリット

現状の多くの推薦

- **利用者側**：情報過多の解消，興味の幅を広げる
- **提供者側**：顧客のロイヤリティの獲得，cross-sellingの期待

マルコフ決定過程による推薦

[Shani 02, Shani 05]

利用者がライフタイムで支払う金額の最大化 ≠ 利用者の満足の最大化

↓

提供者側だけの一方的なメリット

↓

推薦システムへの利用者の信頼の喪失

提供者と利用者の双方のメリットを拡大する

運用目的に応じた手法の選定

- **概要推薦：新規訪問者向け推薦**
  非個人化推薦である人気リストや，直接指定型の内容ベース
- **利用者評価：利用者間の評価情報の交換**
  システム側は基本的に場を提供するだけ
- **関連アイテム推薦：関連アイテムを補助的に推薦**
  協調フィルタリングや，アイテムの特徴を用いた関連アイテム
- **通知サービス，緊密な個人化：積極的に個人化した推薦**
  積極的に利用者の個人情報や行動履歴を利用，協調フィルタリングや間接指定型内容ベース，およびそのハイブリッドを利用
- **能動的な検索：積極的に個人化した推薦**
  直接指定型の内容ベースを利用し，順位付けに協調フィルタリングや間接指定型の内容ベースを利用

まとめ

推薦システム：情報過多への対抗技術

- **内容ベースフィルタリング：**アイテムの特徴を利用
- **協調フィルタリング：**口コミ過程の自動化

推薦システムの過程 (O-I-Pモデル)

1. **データの入力：**明示的 vs 暗黙的
2. **嗜好の予測：**状況に応じて適宜選択
   - **メモリベース法：**データから直接予測
   - **モデルベース法：**事前に嗜好パターンを獲得
3. **推薦の提示：**利用者の目的に応じた提示

その他：プライバシー保護協調フィルタリング，サクラ攻撃，推薦リスト全体の良さを考慮，予測精度以外の要因を考慮した予測

参考文献

参考文献については以下のファイルを参照して下さい

http://www.kamishima.net/archive/recsysdoc.pdf