

KDDI研究所における レコメンデーション技術への取り組み

株式会社KDDI研究所

小野 智弘

2008.9.2@行動計量学会チュートリアル

1

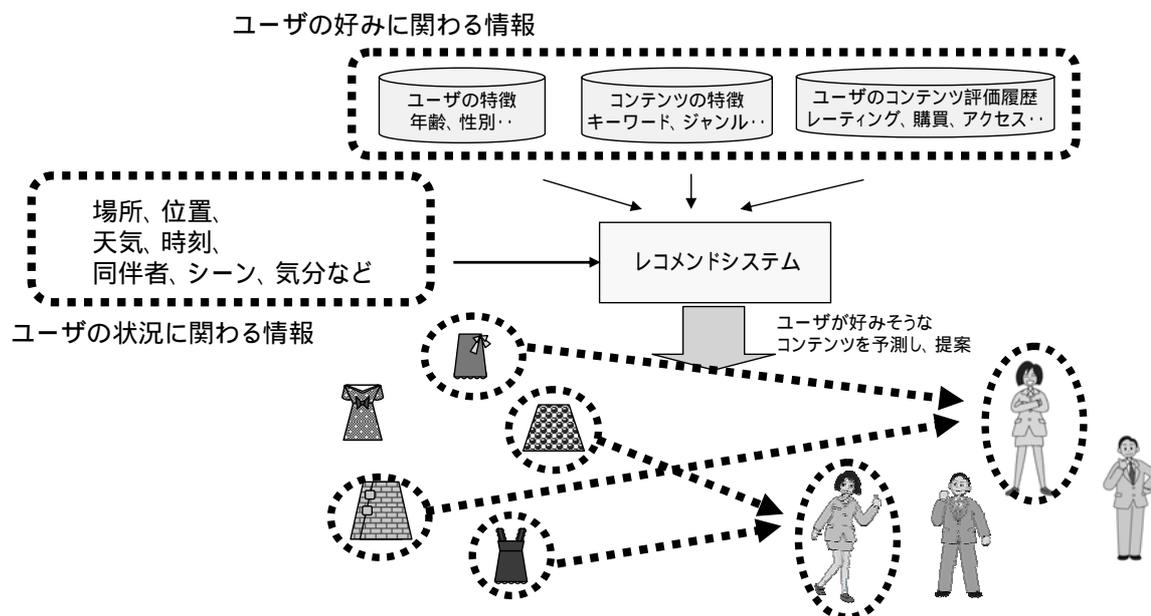
Outline

1. イントロ
 - 情報系・工学系の視点から
2. 事例1:コンテンツベースの事例(音楽)
3. 事例2:コンテンツベースの事例(ブログ)
4. 事例3:履歴ベースの事例
 - クロスメディアレコメンデーション
5. 事例4:属性ベースの事例
 - シーン別映画推薦
6. まとめ

2

“パーソナライズ”されたレコメンデーション

= ユーザの好みに関わる情報を手がかりに、個々のユーザに最適な商品やサービスを個別に推奨すること



国際学術会議

1. ACM International Conference on Recommender Systems[1], 2007年より毎年開催
2. 他に人工知能分野(AAAI, ECAI)、データ工学等の会議の併設としてレコメンデーションに関するワークショップ多数開催

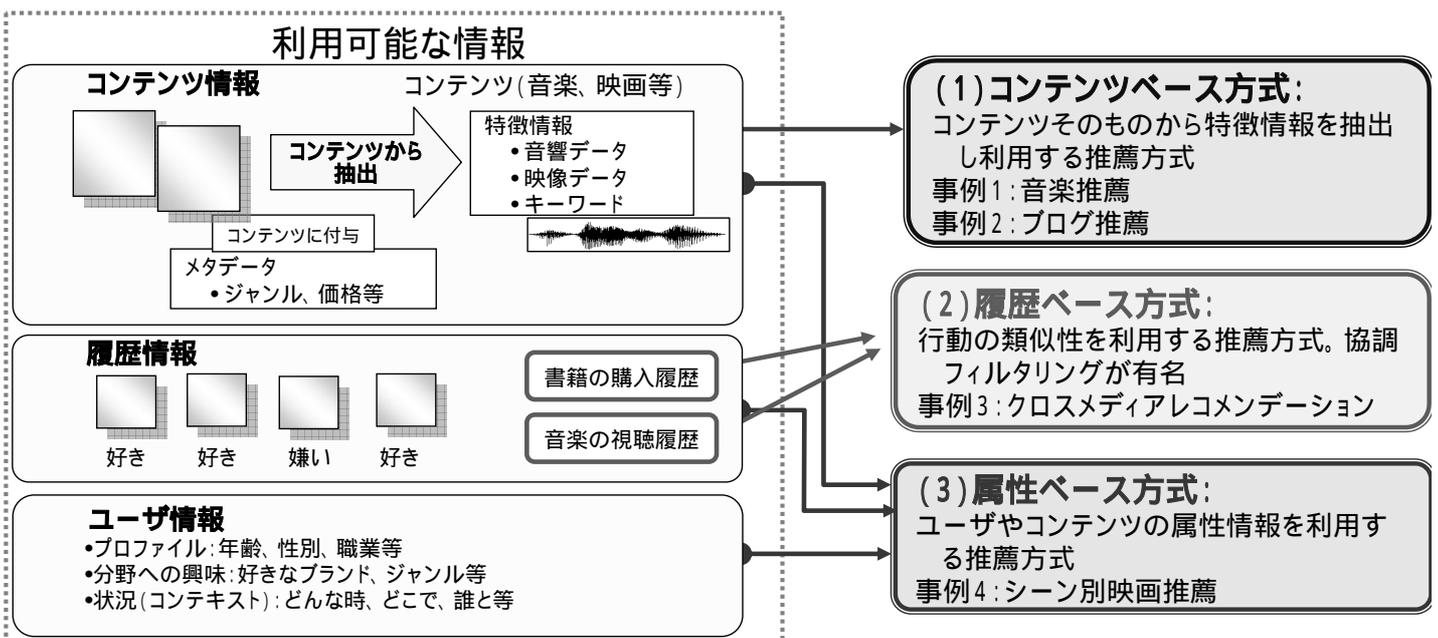
～ レコメンデーション技術に関する国際セミナーRECOMMENDERS06 より ～

7つの将来課題

1. データの少なさ・偏りの克服
2. 超大規模データの扱い
3. 暗黙的な評価履歴の扱い
4. 複数の観点(たとえばレストランではサービス、味、雰囲気)の扱い
5. ユーザの状況の扱い
6. 不正評価の扱い
7. プライバシーの扱い

5～7は比較的最近の話題

1. イントロ
レコメンデーション方式の分類とKDDI研究所の研究事例



特長

- (1) 音楽(事例1)やブログ(事例2)の特徴情報を詳細解析し解析結果に基づき推薦
- (2) 音楽視聴履歴とゲーム購買履歴など、特性の異なる複数の履歴を利用してコンテンツを推薦(事例3)
- (3) シーンや気分を含むさまざまなユーザ情報を活用して、コンテンツを推薦(事例4)

1. イントロ
 - 情報系・工学系の視点から
2. 事例1:コンテンツベースの事例(音楽)
3. 事例2:コンテンツベースの事例(ブログ)
4. 事例3:履歴ベースの事例
 - クロスメディアレコメンデーション
5. 事例4:属性ベースの事例
 - シーン別映画推薦
6. まとめ

2. コンテンツベースの事例(音楽)
一般的なコンテンツベース方式の概要

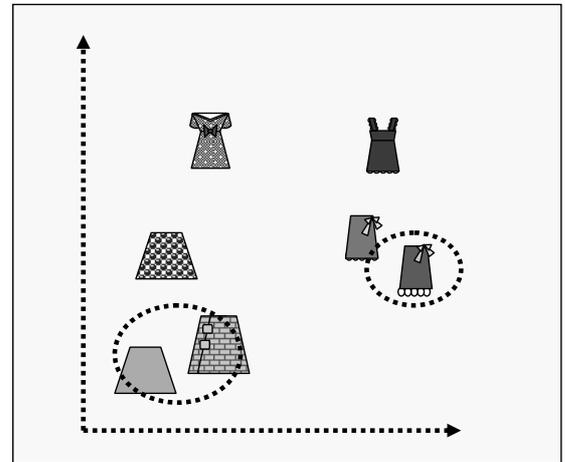
「ユーザが好きなコンテンツの特徴は類似しているだろう」



対象ユーザが好むコンテンツに類似した特徴を持つコンテンツを推薦



1. ユーザが指定したコンテンツ群から特徴抽出
2. コンテンツ特徴データベースの検索
好きなコンテンツ群の特徴に類似し、
嫌いなコンテンツ群の特徴に類似していないもの
3. 推薦コンテンツの選定

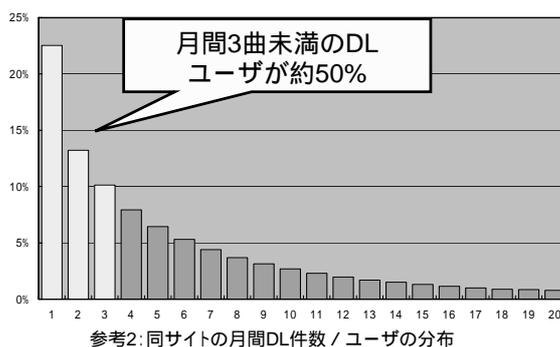
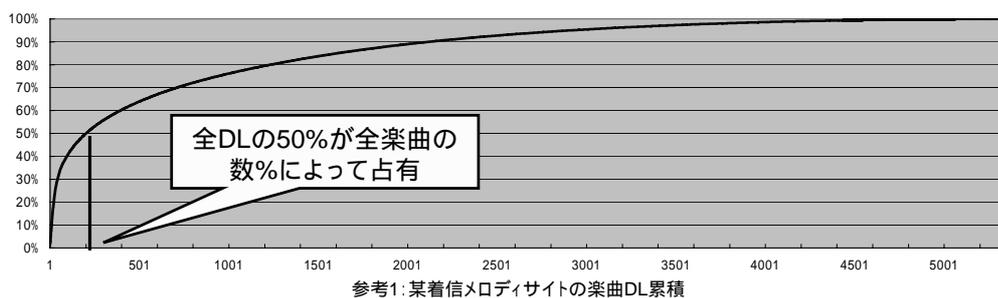


コンテンツの特徴空間

課題 = コンテンツそのものからの特徴方式

1. 事例1: 音楽推薦
2. 事例2: ブログ推薦

• ユーザ購入楽曲の偏重

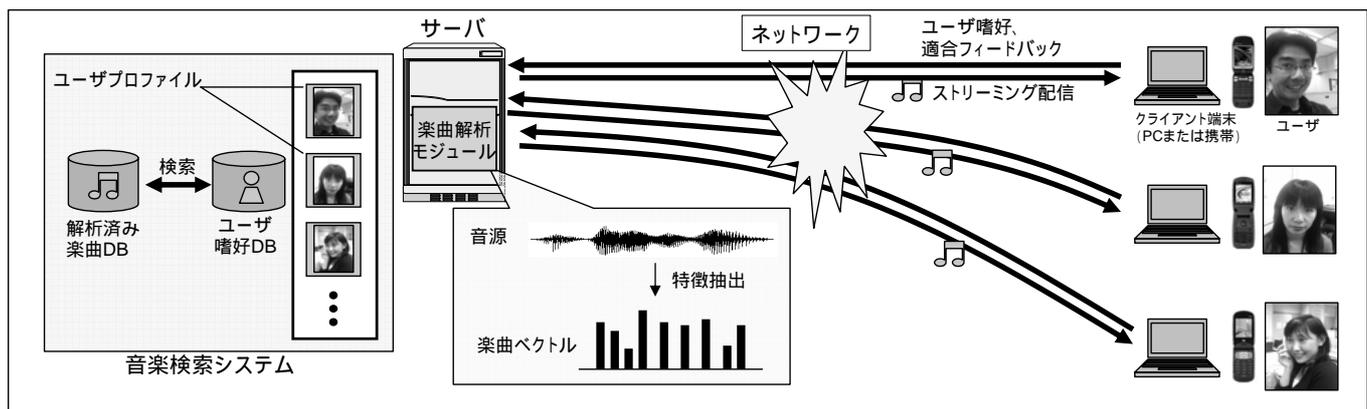
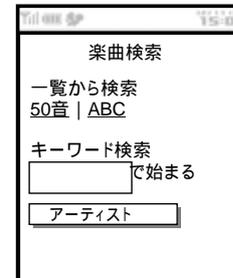


- ごく一部の人気楽曲(ヒット曲、人気アーティストetc)に購入が集中
- 多くのユーザが人気楽曲のみを購入

↓

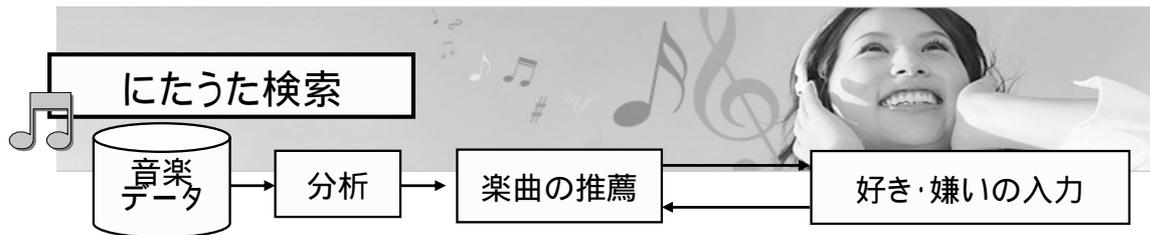
• 客単価が低い
• 多くの楽曲がユーザに発見されることなく埋没

- 電子化された音楽情報の普及
 - au LISTEN MOBILE SERVICE (LISMO)、iTunes Store などのネット上の音楽配信サービスの流行
 - 配信対象楽曲数の急増
音楽情報検索の必要性高まる
- 既存の「音楽情報検索」技術
 - 音楽のメタ情報に基づく検索
 - アーティスト・タイトル・ジャンルなど
 - Content-based検索
 - 「鼻歌」入力(query-by-humming)による検索
 - 「聴かせて検索」
前提: 既知の音楽データの検索
- 本研究の目的
 - ユーザの音楽嗜好を反映したcontent-based検索
 - 未知の「イイ曲」の発見が可能!



処理の流れ

1. 楽曲データの登録
 1. 新規楽曲を自動解析して特徴抽出を行い、楽曲データベースに登録
2. 楽曲の推薦
 1. ユーザが指定、またはユーザー嗜好DBに登録済みの楽曲群に特徴が類似した楽曲を推薦
 2. 推薦楽曲に対するユーザーのフィードバックを取得し、推薦結果を修正して再度推薦



知らない曲も推薦してくれる

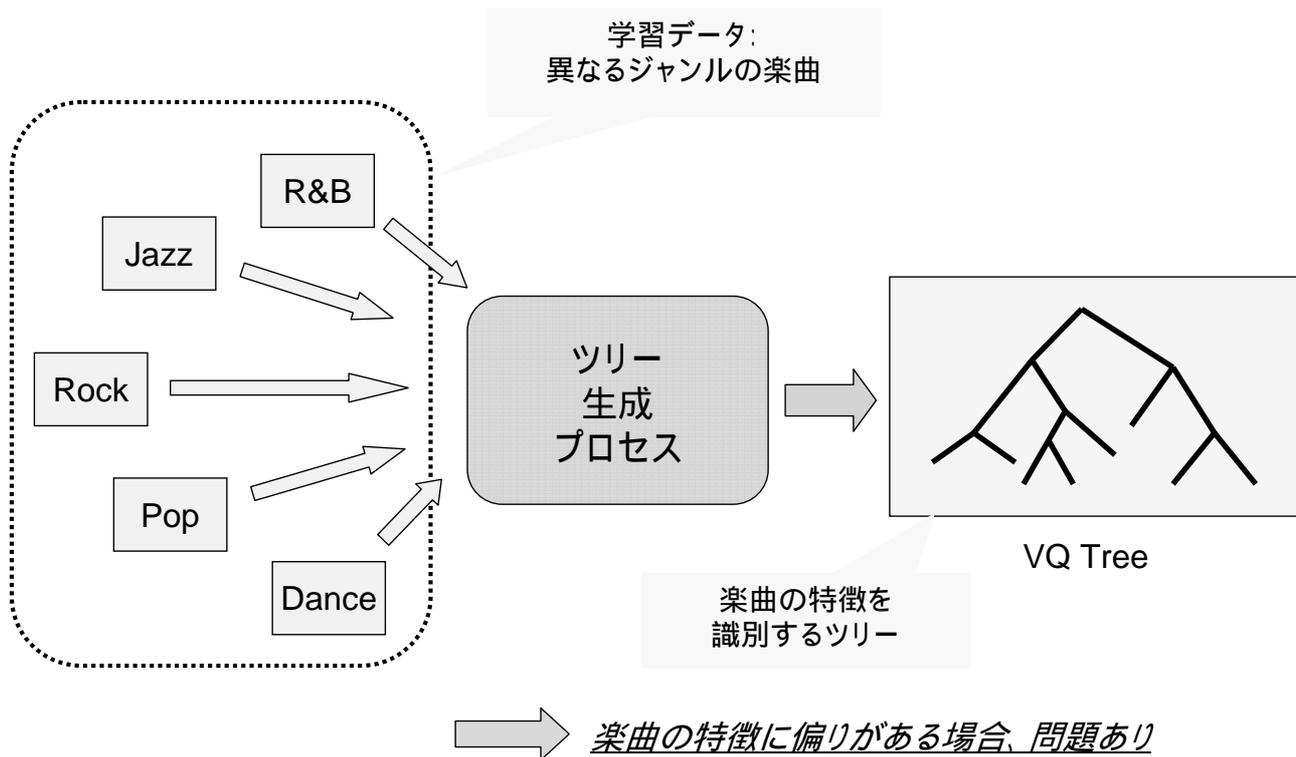
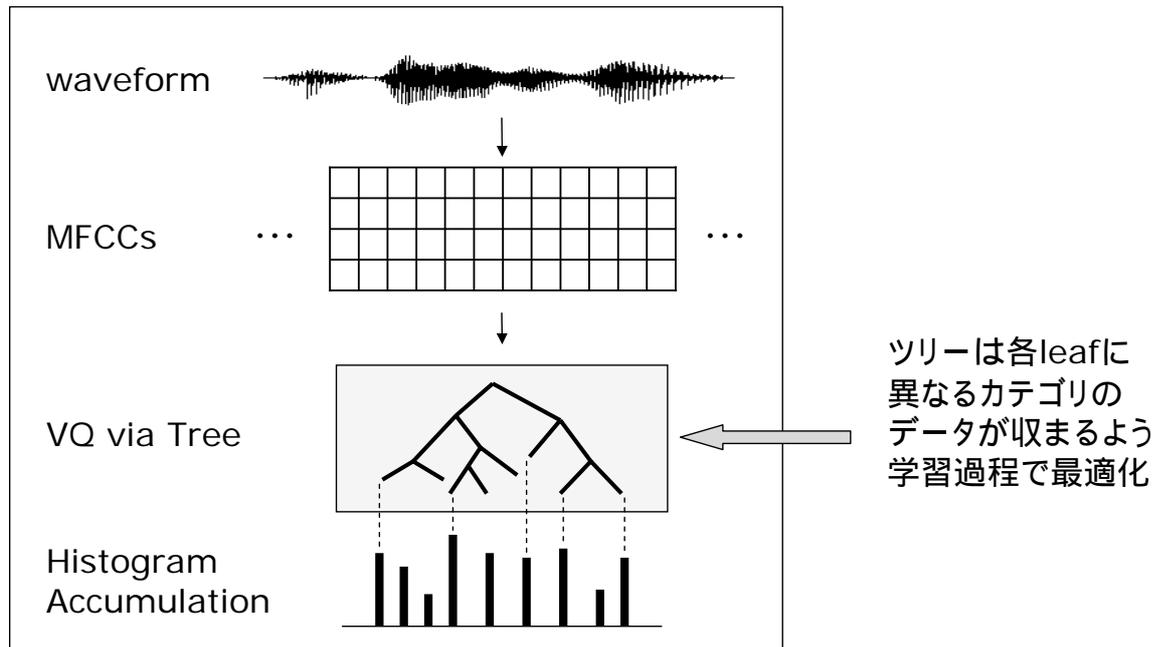
人手による情報整備が不要

埋もれたいい曲を発見し推薦してくれる(ロングテール対応)

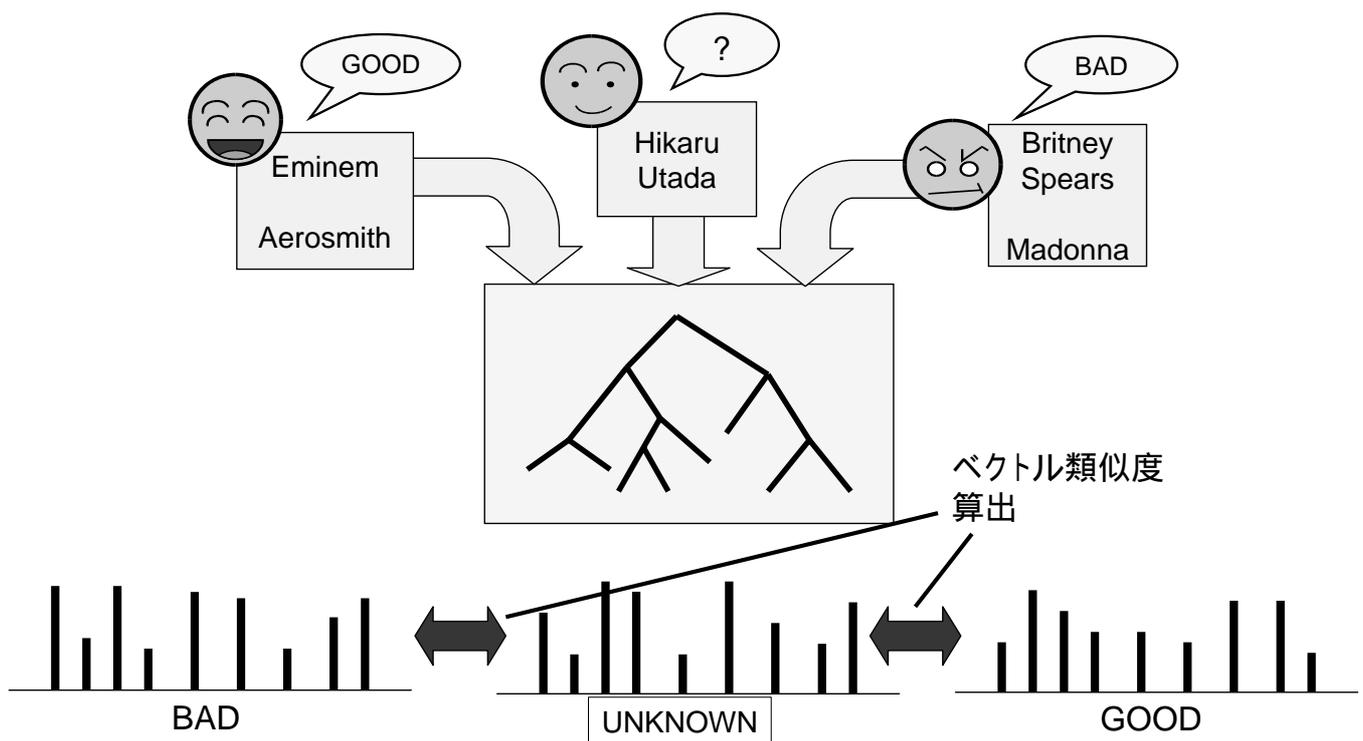
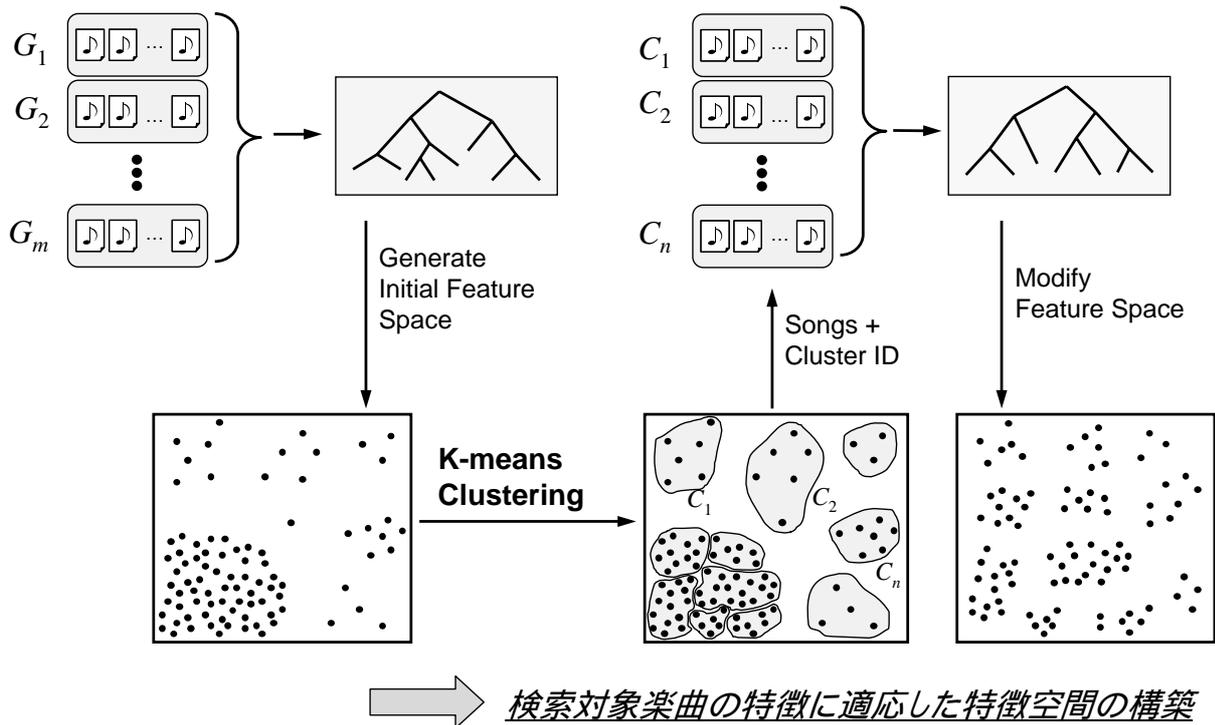
大量の履歴データが不要であるのでショップ立ち上げ時に対応可能(コールドスタート対応)

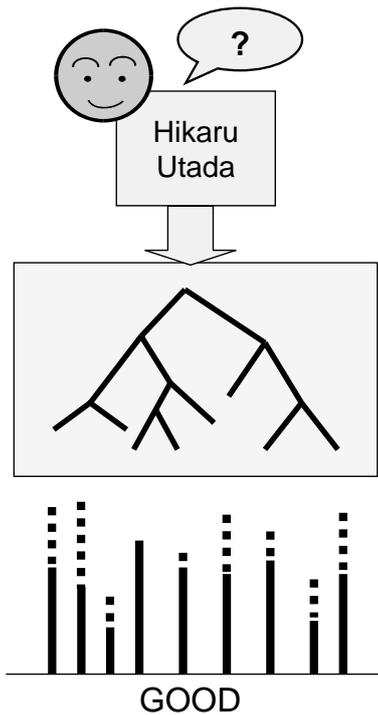
推薦結果に対する評価(適合フィードバック)を利用して精度を自動改善

- 楽曲解析モジュール = 楽曲からの音響的特徴抽出技術
 - Tree-based vector quantization (TreeQ)[Foote,99]に基づく音楽からの特徴抽出
 - 検索対象楽曲に適応した特徴空間の構築
- ユーザの音楽嗜好変化への対応
 - 検索結果に対するユーザからの適合フィードバックを利用し、ユーザプロフィールを最適化



楽曲クラスタリングを利用した特徴空間再構築





- 目的:
 - ユーザからのFB情報を利用し、検索精度向上を図る。
- 手法:
 - 好きな楽曲の情報を初期プロフィールに蓄積
 - 更新されたプロフィールを利用し、再検索
検索精度向上を実現

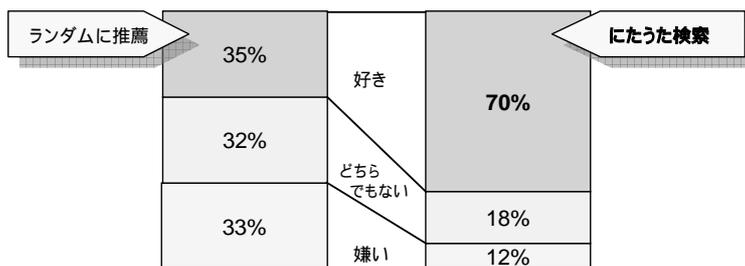


Fig. ランダム推薦とにたうた検索結果の精度比較

- 実験データ:
 - 検索対象楽曲約7000曲に対し、被験者が5段階の主観評価値を付与
- 実験方法:
 - 個々の被験者の好きな楽曲をランダムに選択 検索クエリ構築
 - 検索結果の上位楽曲に対する評価値を元に検索精度を評価
- 結果:
 - ランダム推薦の約2倍の割合で、各ユーザが好む(=評価値4以上)楽曲が検索できた
 - 適合フィードバックにより、一定の割合での検索精度向上も確認

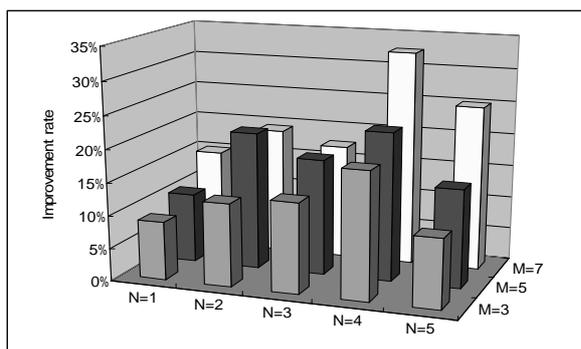


Fig. 適合フィードバックによる検索精度改善率
 (ACMMM'03発表資料より抜粋)

• 携帯での音楽再生アプリの課題

- 大容量化にともなう楽曲の埋没
 - 携帯に格納する楽曲数が多くなるにつれ、聴きたい曲を探すことが困難に限られた楽曲のみを頻繁に聴く
 - 携帯(iPod)が楽曲の「墓場」になりつつある
- インタフェースの制限
 - 携帯のテンキー・十字キーのみでは楽曲リストの閲覧がわずらわしい(iPodのスクロールホイールですら面倒)

➡ 携帯への「にたうた検索」実装により、解決(改善)が可能!

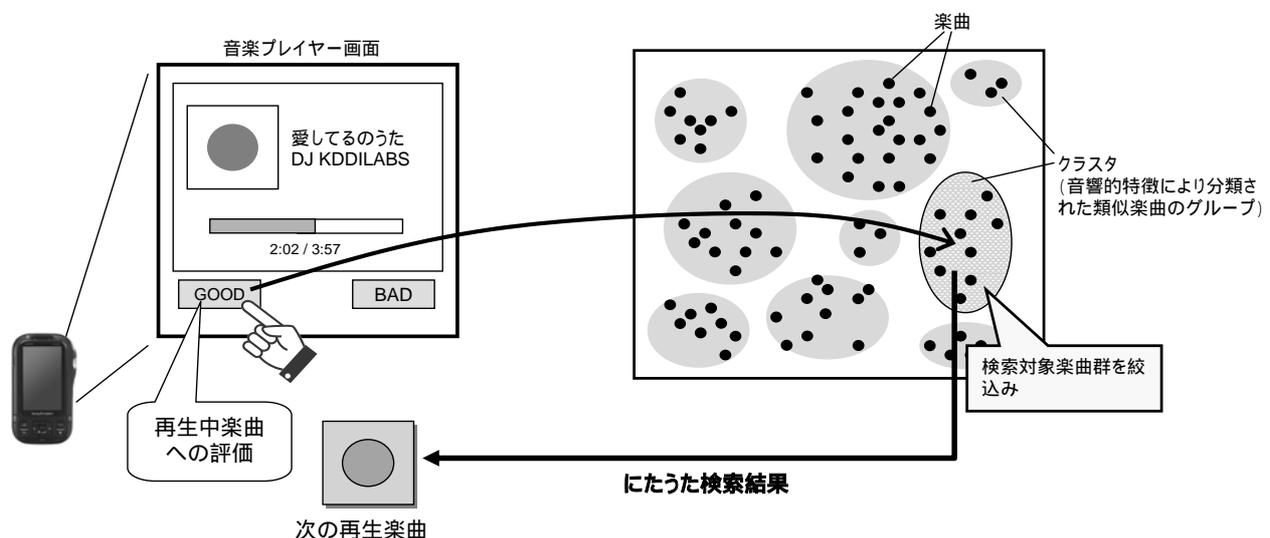
• 携帯への「にたうた検索」実装における課題

- 検索にかかる処理負荷
 - 検索対象楽曲数の増加 計算量の増加
 - 携帯自体の処理能力の限界

➡ 大幅な検索処理の効率化が不可欠

• 「にたうた検索」技術の大幅な効率化を実現

- ユーザの音楽嗜好(ユーザプロファイル)との類似性が高い楽曲に検索対象範囲を絞り込む「階層型」検索アルゴリズムを開発



- 携帯版「にたうた検索」技術をベースにした音楽検索・再生用BREWアプリケーション

- 処理フロー
 1. 再生中の楽曲に対し、「GOOD」「BAD」の評価を入力
 2. 入力評価情報(過去の履歴を含む)からユーザプロフィールを生成し、にたうた検索
 3. スコア最上位の楽曲を再生
 4. (1.に戻る)
 - メリット
 - 簡易な評価入力により、携帯の中の楽曲から聞きたい曲を効率的に検索(発見)することができる
- より使い勝手の良い携帯音楽再生アプリの実現!



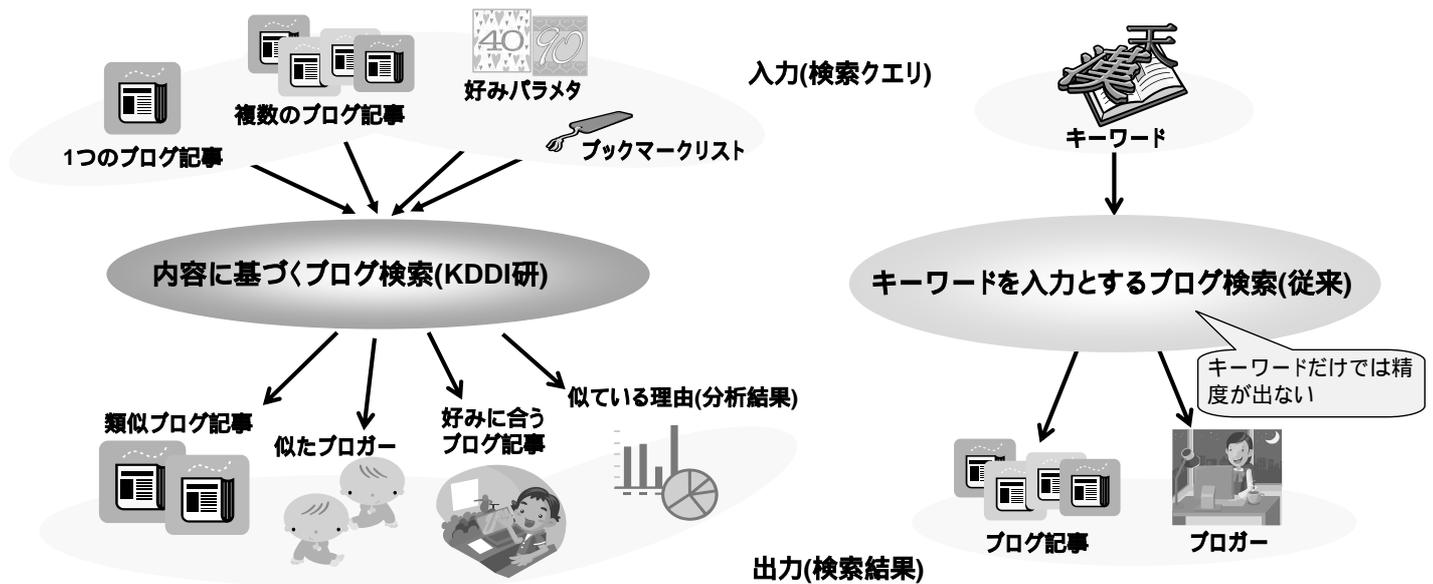
Outline

1. イントロ
 - 情報系・工学系の視点から
2. 事例1:コンテンツベースの事例(音楽)
3. 事例2:コンテンツベースの事例(ブログ)
4. 事例3:履歴ベースの事例
 - クロスメディアレコメンデーション
5. 事例4:属性ベースの事例
 - シーン別映画推薦
6. まとめ

3. コンテンツベースの事例(ブログ)

背景

Challenge for Innovation



技術課題: 単語が少ないブログ記事でも検索できること
大量のブログを実時間内で検索できること

25

3. コンテンツベースの事例(ブログ)

検索機能概要

Challenge for Innovation

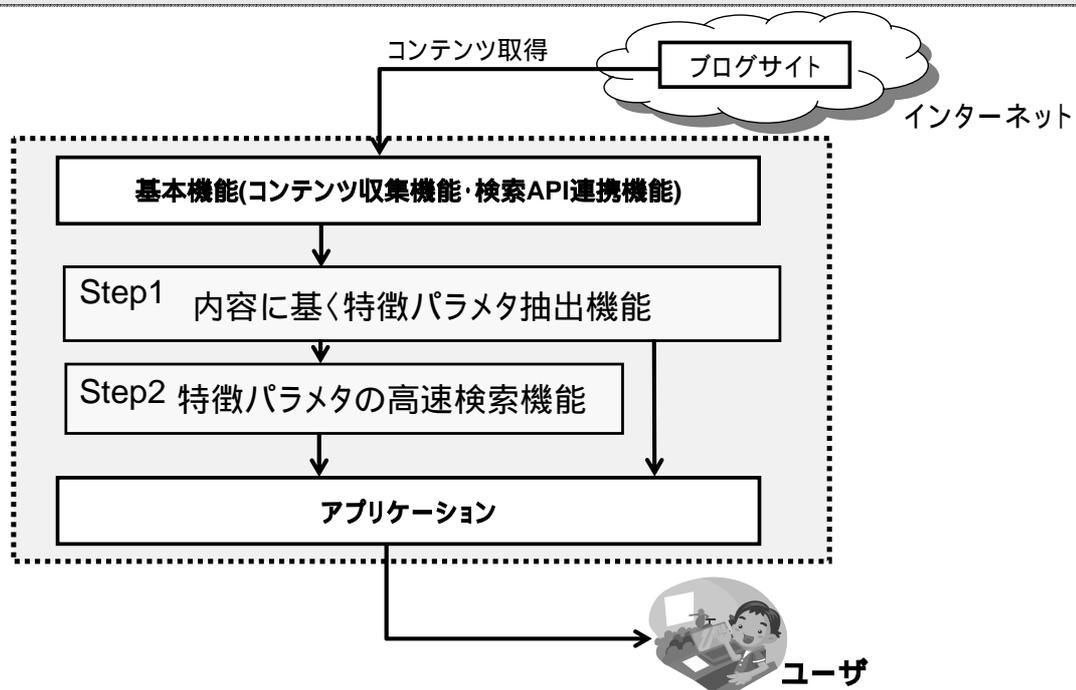


Step1: 内容に基づく特徴パラメタ抽出機能

ブログ内に出現する単語の数と重要度に基づく特徴ベクトル生成し、あらゆるブログ記事を数値化

Step2: 特徴パラメタの高速検索機能

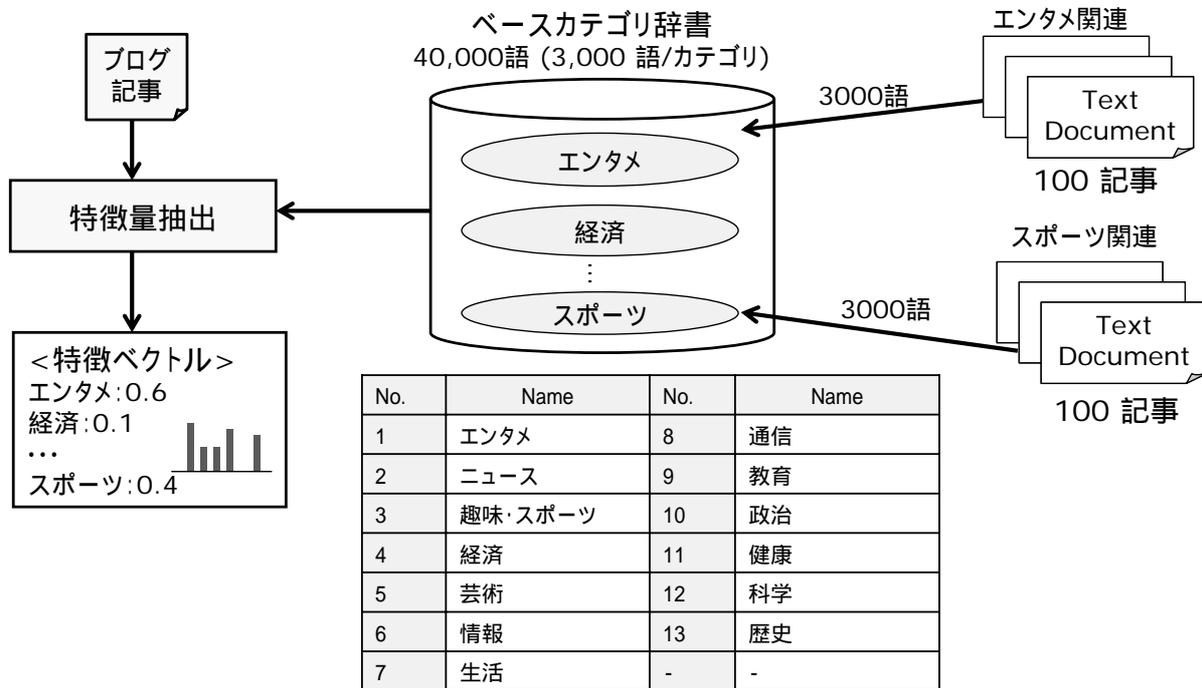
2段階処理で高速検索を実現



26

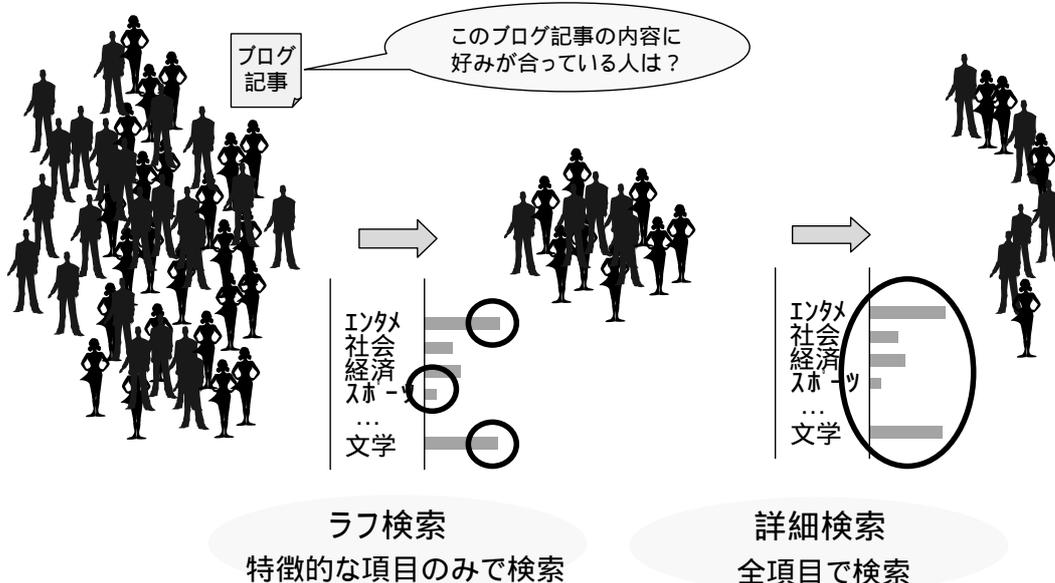
3. コンテンツベースの事例(ブログ) Step1.内容に基づく特徴パラメタ抽出機能

- 文書分析用の分類辞書を生成
 - ブログ記事が各分類辞書にどの程度近いかを算出し、特徴量として数値化する
 - あらゆるテキスト文書を数値化できる
 - 各言語対応の辞書さえ作れば、日本語と英語のブログの相互検索も可能

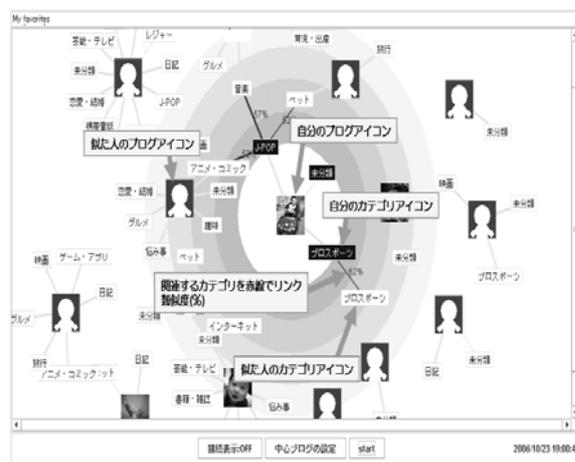
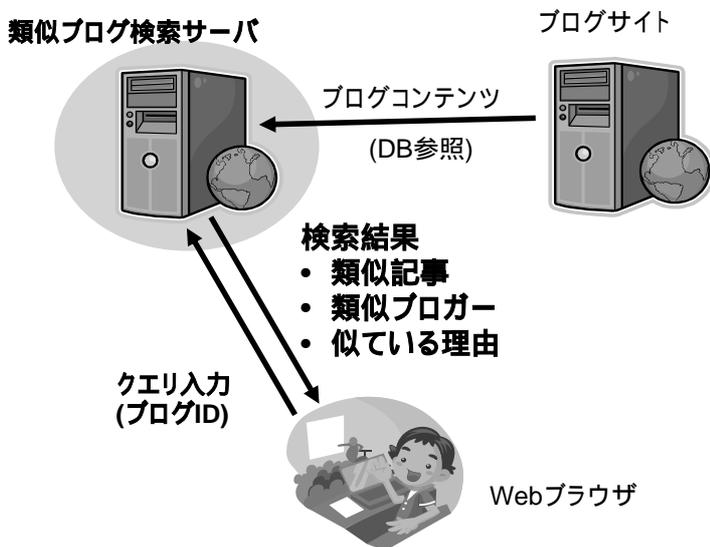


3. コンテンツベースの事例(ブログ) Step2.特徴パラメタの高速検索機能

- 2段階の絞り込みにより高速検索を実現
 - ラフ検索: 記事の特徴づける項目のみで1次検索を行い粗く絞り込む
 - 詳細検索: 1次検索の結果に対し、全13項目を対象とした2次検索を行う
- 10000人 20人の絞り込みで約0.15秒
 - 単純な検索方法では約6秒

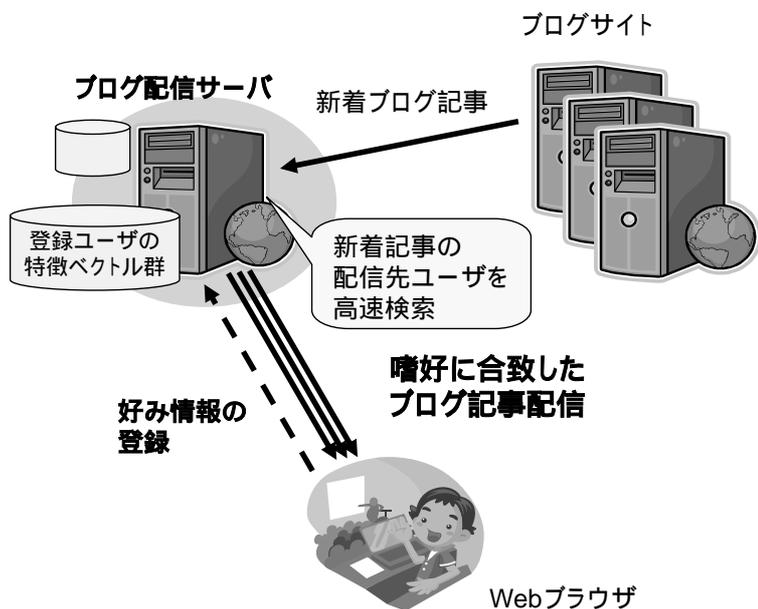


指定したブログに類似する記事、ブロガーを類似理由とともに検索・提示



検索結果表示例
 (ブログコミュニティ検索@auoneラボ)

新着ブログ記事を、ユーザが登録した好みに応じて自動配信



配信されたブログの表示例(STB)



配信されたブログの表示例(携帯電話)

1. イントロ
 - 情報系・工学系の視点から
2. 事例1:コンテンツベースの事例(音楽)
3. 事例2:コンテンツベースの事例(ブログ)
4. 事例3:履歴ベースの事例
 - クロスメディアレコメンデーション
5. 事例4:属性ベースの事例
 - シーン別映画推薦
6. まとめ

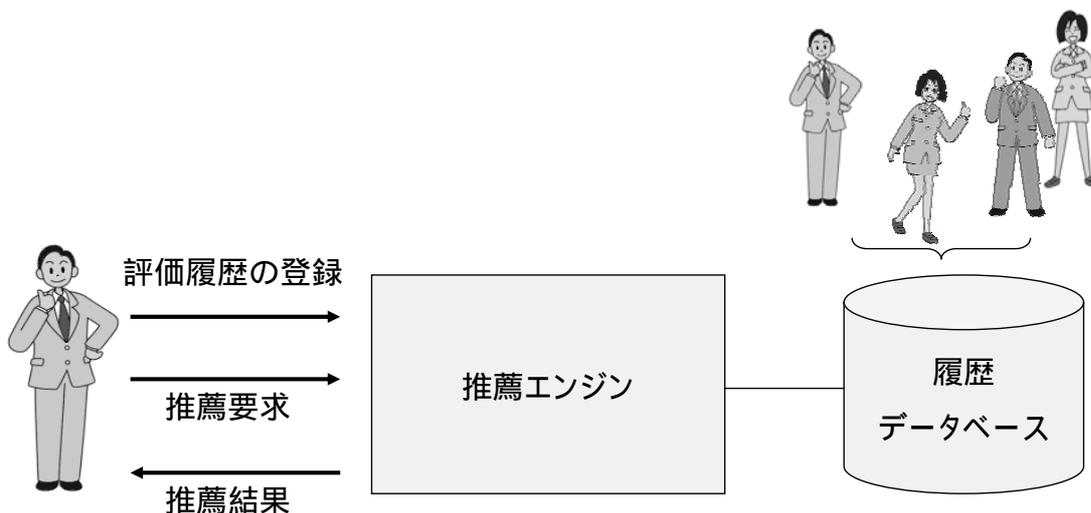
4. 履歴ベースの事例

一般的な履歴ベース方式(協調フィルタリング)の概要(1)

「行動(=評価履歴)が類似したユーザ同士は嗜好も類似している」

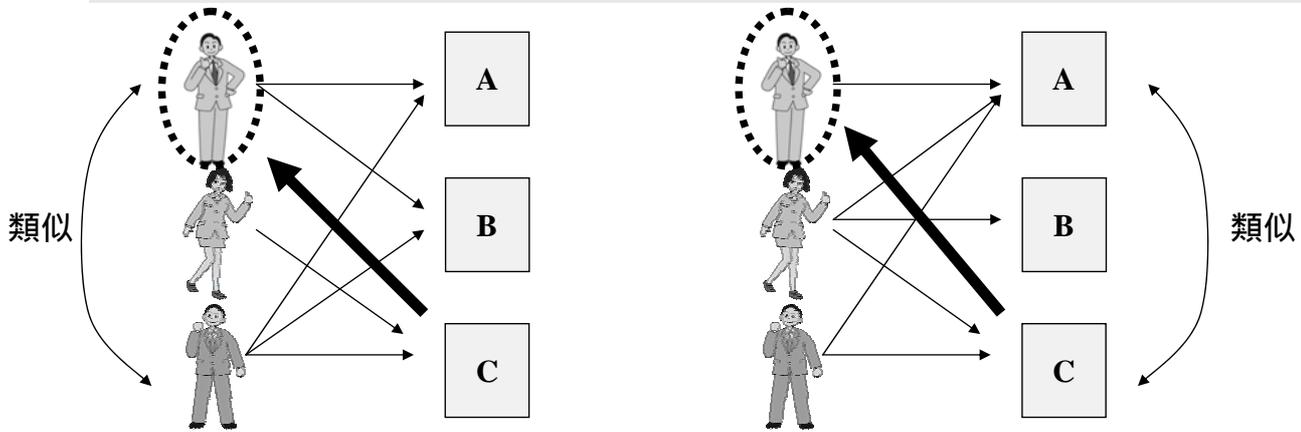


対象ユーザと似たような行動をとっているユーザの評価履歴から対象ユーザの嗜好を推測



類似度の見方

1. User-Based: ユーザ間の類似度を基準に予測評価値の算出
2. Item-Based: コンテンツ間の類似度を基準に予測評価値の算出



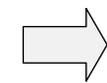
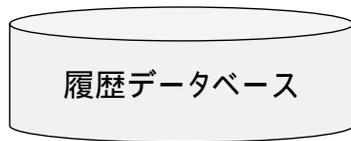
評価値の形式

1. Multi Value(多値): 5(とても好き) - 4 - 3 - 2 - 1(とても嫌い)
2. Binary(2値): 1(好き) - 0(嫌い)
3. Unary(単値): 1(買った、見た) - Null

注: 空白は“嫌い”を意味しているわけではない

処理ステップ (User-based, Multi Valueの場合)

1. 類似ユーザの探索と類似度の計算
2. 対象ユーザの予測評価値の計算
3. 推薦コンテンツの選定



対象ユーザ

ユーザ

		コンテンツ						
		C ₁	C ₂	C ₃	C ₄	C ₅	C ₆	C ₇
ユーザ	U ₁	5		5	4		2	
	U ₂		4	4	4	5		
	U ₃			1	2		4	4
	U ₄		5	4	4	4	1	5
	U ₅						5	4
	U _i		5	4	4		No1	

• 協調フィルタリングの課題

- 履歴情報がないユーザには Recommend できない
 - 履歴情報がないアイテムを Recommend できない
- 流行っていないサイト・立ち上げ直後のサイトでは推薦できないケースが増大



Book



CD

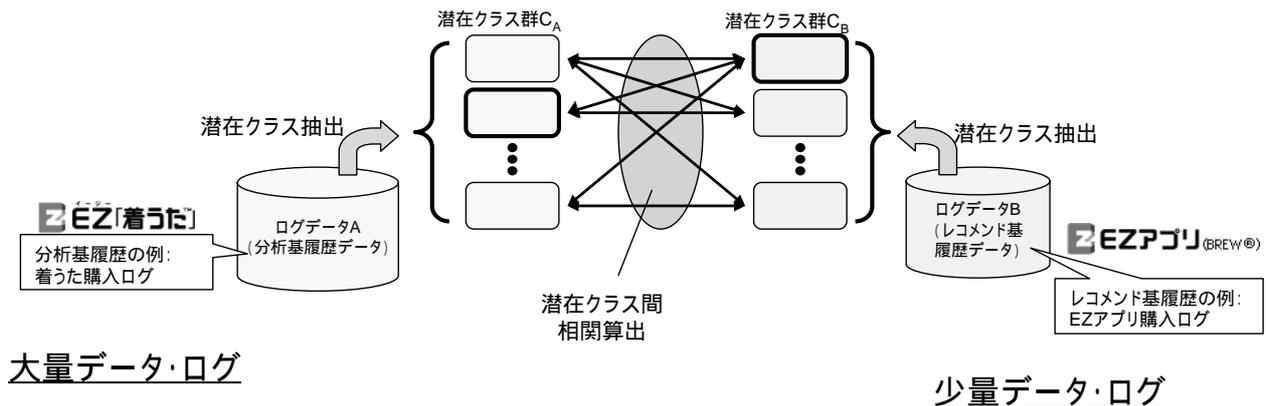
	item_f	item_g	item_h	item_i	item_j
User_A	0	5	4	3	4
User_B	4	2	4	4	3
User_C	0	3	1	3	4
User_F	3	3	4	0	2
User_G	3	0	3	4	0

	item_a	item_b	item_c	item_d	item_e
User_A	0	0	0	0	0
User_B	4	0	0	4	0
User_C	0	3	0	3	4
User_D	0	0	0	0	0
User_E	0	0	0	0	0

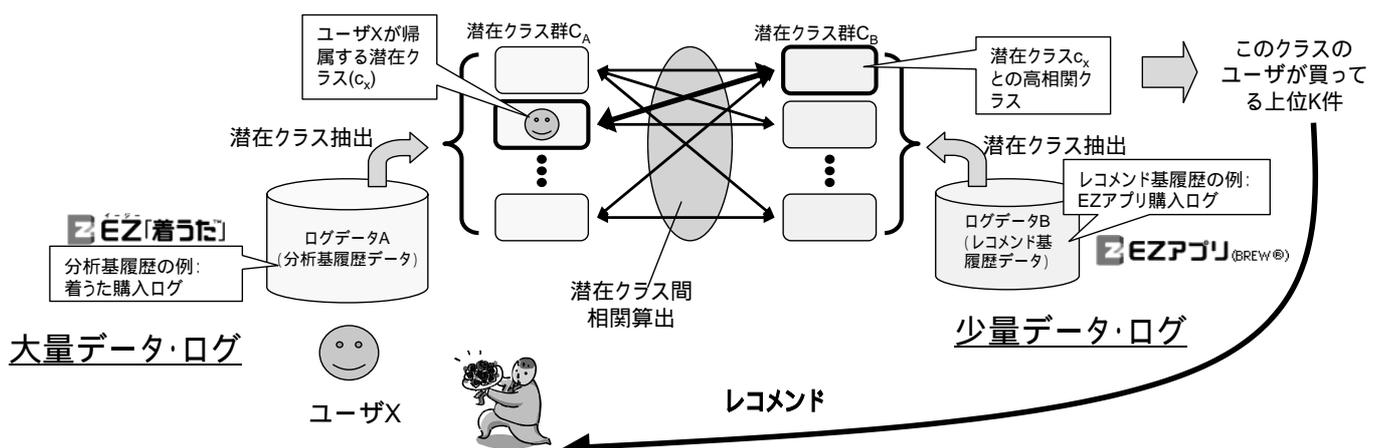
共通ユーザが存在するサイトがあれば、そのサイトの履歴と連結して推薦？

安易な連結は危険。本の趣味が似ているからといって、CDの趣味は異なる可能性あり

- 方針:
 - ログデータが少ないサイトの商品を、ログデータが多いサイト履歴との関連性を利用して推薦
- クロスメディア分類システムの構築
 - ステップ1: 2種類の履歴データ(分析基・レコメンド基)からそれぞれ「潜在クラス」を抽出
 - 分析基: 多量のユーザ・履歴情報を含むログデータ
 - レコメンド基: レコメンドしたいアイテムが含まれるログデータ(小規模)
 - ステップ2: 潜在クラス間の類似度を算出

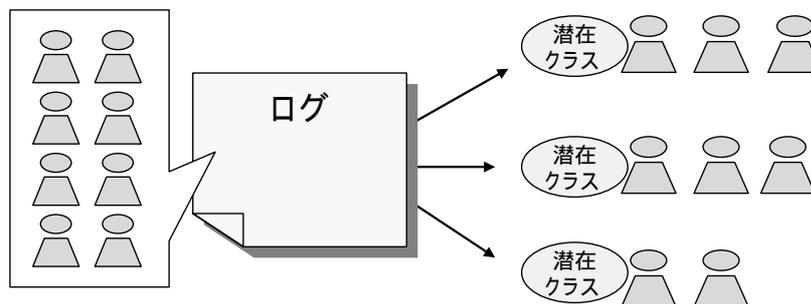


- 分類システムを利用した推薦手順
 - ユーザXがレコメンド基のいずれかの潜在クラスに帰属すれば同クラスの人気アイテムを推薦
 - そうでなければ、
 - ユーザXが帰属する分析基の潜在クラスを特定
 - 同潜在クラスと最も類似度の高いレコメンド基の潜在クラスを特定
 - レコメンド基上の同潜在クラスの人気アイテムを推薦



• 潜在クラスの抽出

- 潜在クラスモデル[4]を適用し、分析基、レコメンド基それぞれから抽出



分析基データの潜在クラス L_a と推奨基データの潜在クラス L_b は次のように表現

$$L_a = \{t_{a_1}, t_{a_2}, \dots, t_{a_c}\} \dots (1)$$

$$L_b = \{t_{b_1}, t_{b_2}, \dots, t_{b_d}\} \dots (2)$$

ユーザ U_i が潜在クラス t に所属する確率を $q_t^{U_i}$ とした場合,

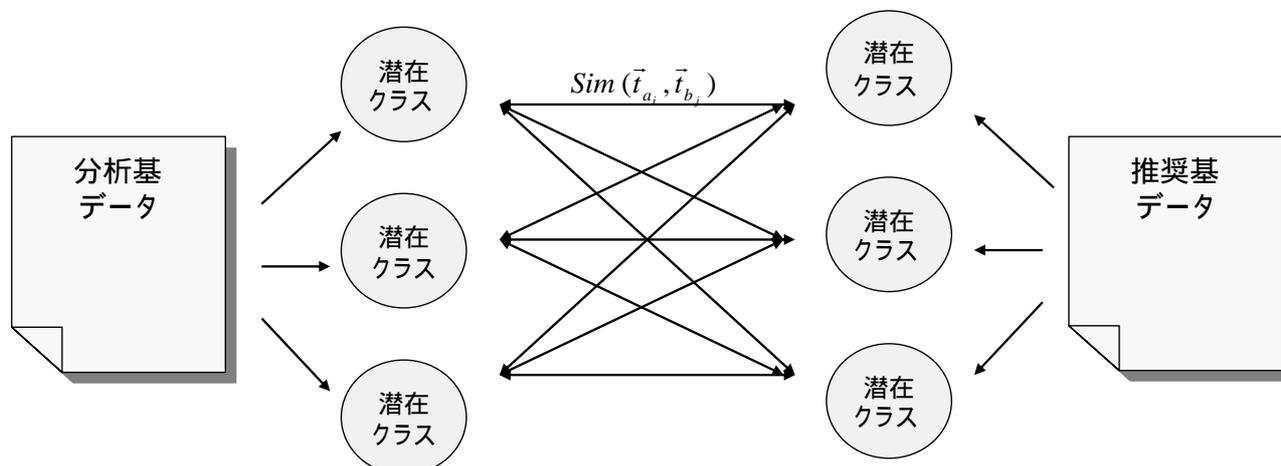
潜在クラス t は次のように n 次元(共通ユーザ数分)のベクトルとして表現.

$$\vec{t} = (q_t^{U_1}, q_t^{U_2}, \dots, q_t^{U_n}) \dots (3)$$

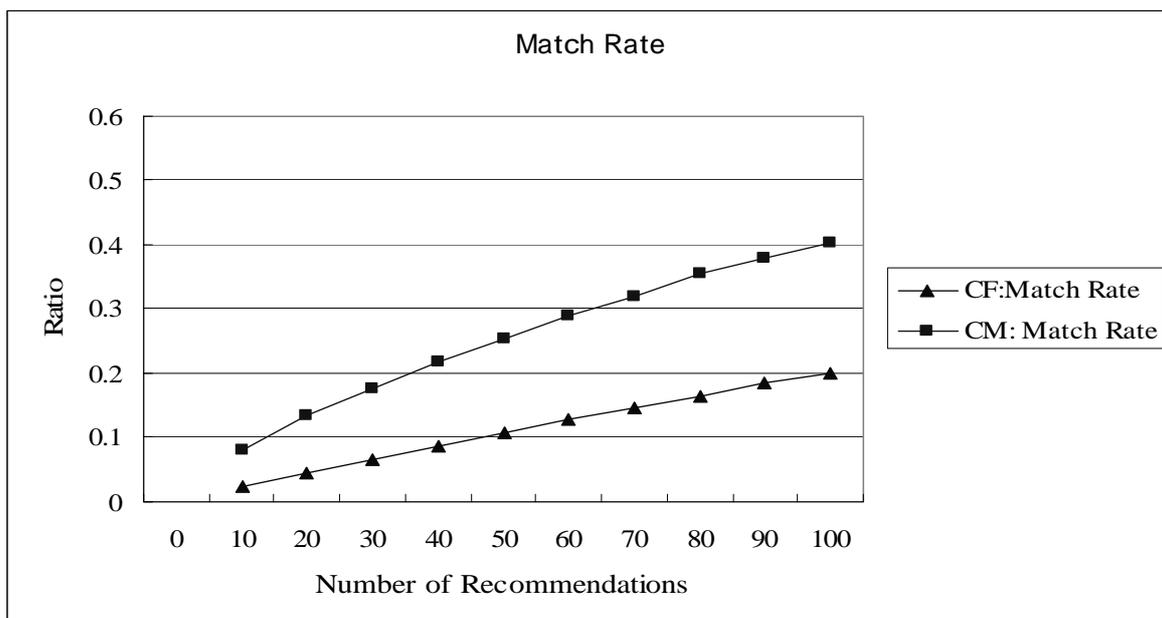
L_a = 分析基データ内の潜在クラス
 L_b = 推奨基データ内の潜在クラス
 c = 分析基データ内の潜在クラス数
 d = 推奨基データ内の潜在クラス数
 n = 共通ユーザ数

• 潜在クラス間類似度の算出

- 共通ユーザを用いてベクトル表現した各潜在クラス間の類似度 (sim)をピアソン相関係数やコサイン類似度で算出



- 目的
 - 履歴情報がないユーザに対するレコメンデーション結果の精度を計測する.
- 比較対象
 - “Item-based Top-N Recommendation Algorithm”[3]を実装したレコメンデーションエンジンを用いた.
- 評価用データ
 - GroupLens[2]プロジェクトにて公開されている映画を鑑賞したユーザの10万件分の履歴情報を用いた.
 - dramaに属する映画の履歴情報から、drama以外の映画を推奨する.
- パラメータ
 - 潜在クラス数をそれぞれ20個、相関係数はコサイン類似度を使用
- 評価指標: Match Rate:
 - 過去の履歴情報から、その後実際評価したアイテムを予測した場合の再現性)
 - 推奨基データに含まれているユーザから評価用ユーザをサンプリングし、そのユーザの推奨基データの履歴情報を削除する.
 - レコメンド結果を計算した後、削除される前の履歴情報と比較し一致した比率



クロスメディアレコメンデーション方式が協調フィルタリングに比べ、高い再現性(Match Rate)を達成

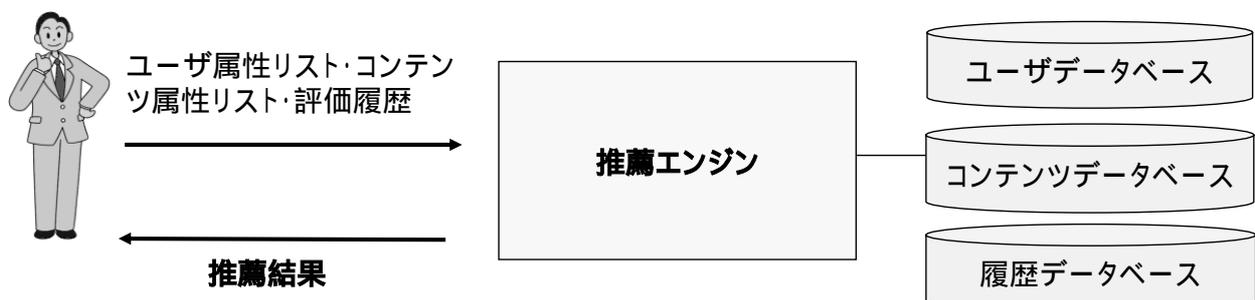
1. イントロ
 - 情報系・工学系の視点から
2. 事例1:コンテンツベースの事例(音楽)
3. 事例2:コンテンツベースの事例(ブログ)
4. 事例3:履歴ベースの事例
 - クロスメディアレコメンデーション
5. 事例4:属性ベースの事例
 - シーン別映画推薦
6. まとめ

5. 属性ベースの事例
一般的な属性ベース方式の概要

「特徴(属性)の類似しているユーザ同士やコンテンツ同士は好み方・好まれ方が類似しているだろう」

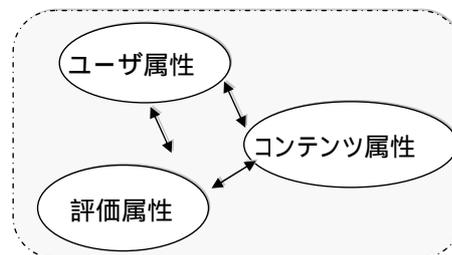


対象ユーザと類似した属性を持つユーザが高く評価(購買)したコンテンツや
対象ユーザが高く評価(購買)したコンテンツに類似した属性を持つコンテンツを推薦



1. ユーザ属性とコンテンツ属性が付いた評価履歴から、ユーザ属性～コンテンツ属性～評価属性間の関係性を抽出
2. 抽出した関係性(モデル)を利用し、対象となるユーザの評価が高いと予測されるコンテンツ(属性)を出力
3. 推薦コンテンツを選定

関係性を表現



ユーザの嗜好は個人差(Personalized)と状況差(Context-aware)

嗜好に影響を与える状況の例

- 時間帯(朝～昼～夜)、曜日、ビジネス・レジャーなどの“時”関連
- よく行く場所、知らない場所などの“場所”関連
- 誰と: 一人で、友人と、家族と、恋人と
- 気分(笑いたい、泣きたい)、忙しい、暇など
- 体調(疲労度、空腹度)
- 気候(天気、気温)

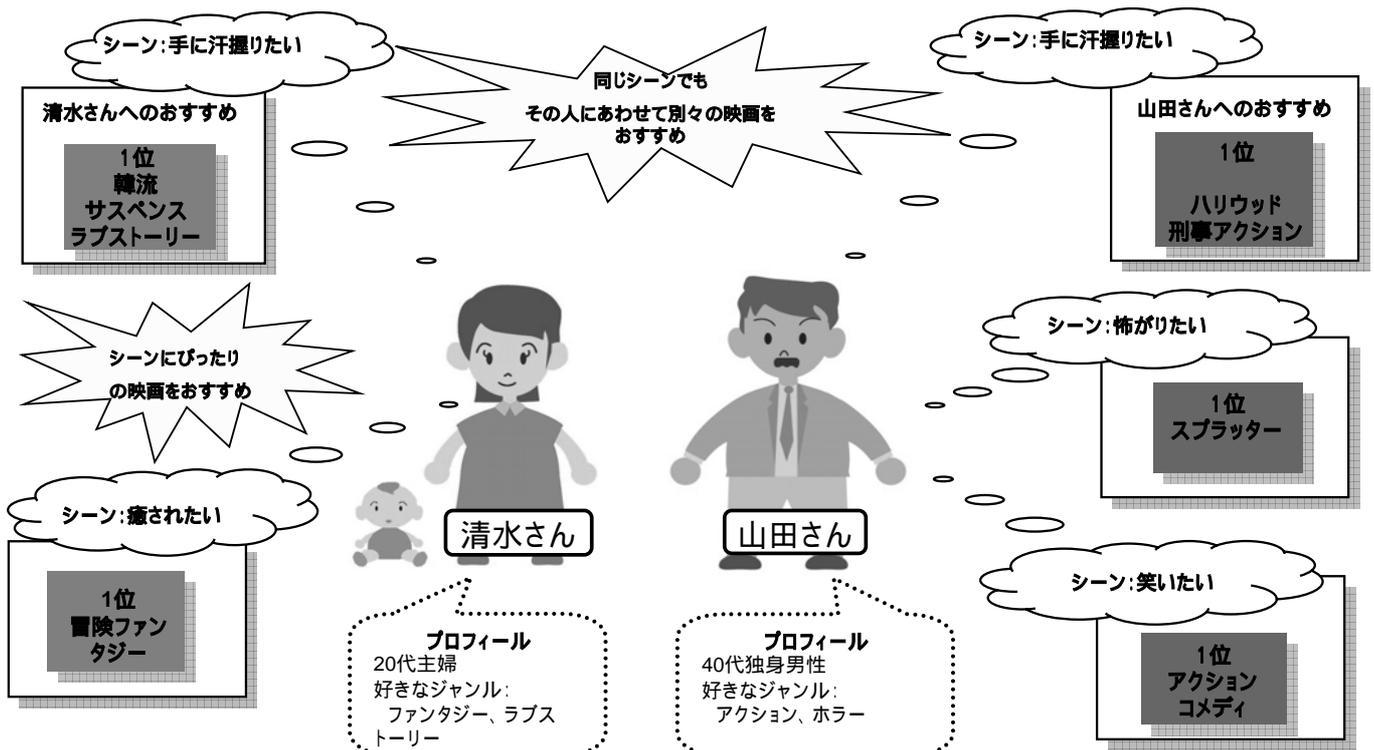
など

**Context, situation, occasionなど・・・
統一的な定義は確立されていない**

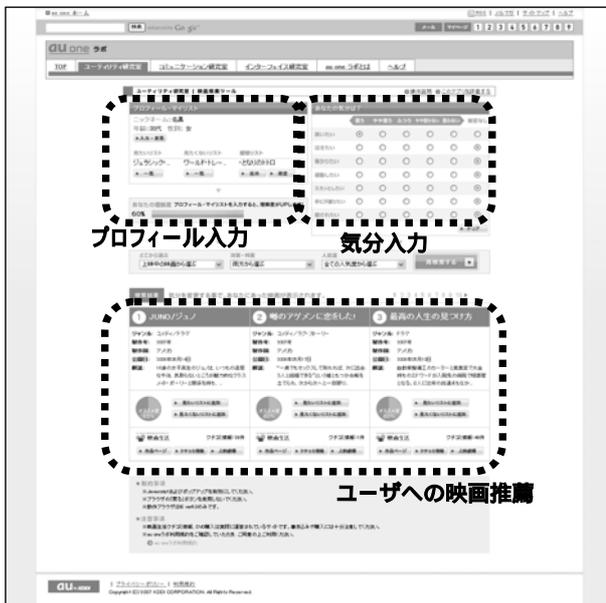
ユーザの気分や状況を考慮した映画推薦システムの2つの目標

- **個人化 (Personalized)**
 - ユーザの嗜好 (個人間の差) を考慮
- **状況依存 (Situation, Context dependent)**
 - ユーザの状況 (個人内の差) を考慮

上記の両方を同時に満たす



ユーザの気分や情報を考慮した「映画推薦ツール」



au oneラボ (https://labs.auone.jp/CGI/top.cgi/) で公開中:

KDDI研究所成果の公開実験サイト
映画推薦ツールを含む6つのアプリを公開

特徴:

- 属性ベース方式のアプローチ
- 「プロフィール」×「気分」で興味を推定
- プロフィール入力が少なくても何らかの推薦
- 入力情報が多くなればより適切な推薦
- 映画内容を記述する独自の属性語の採用により迅速な新作映画への対応を実現
- フィードバックデータにより自動的に精度向上

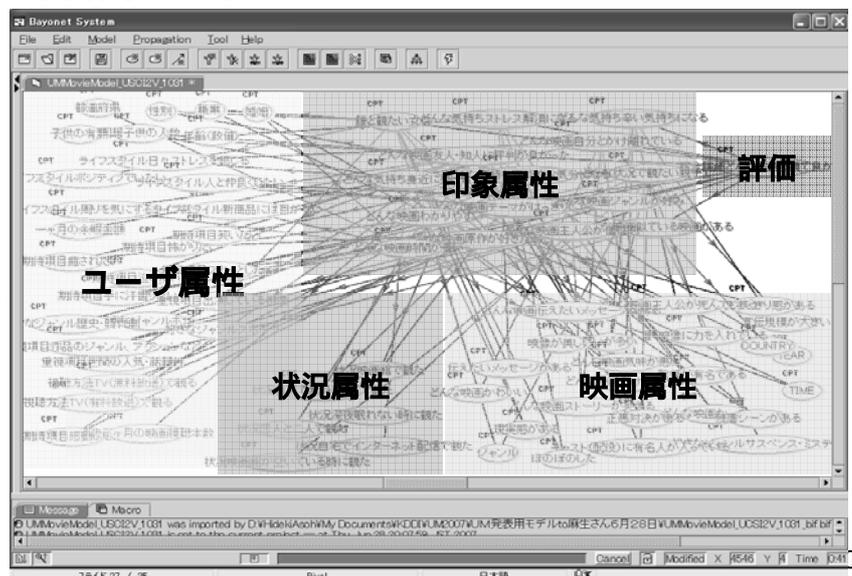
- 「ユーザ属性」「状況属性」「映画属性」「評価属性」間の関係性を表現した映画嗜好モデルを構築して利用
- 映画嗜好のモデル化に、確率的手法(ベイジアンネット)を利用
 - 属性の同時確率分布 $P(\text{ユーザ属性, 状況属性, 映画属性, 評価})$ を表現
 - 75 ノード(属性), 115 リンクで映画嗜好を表現

映画評価データ
(WEB調査により収集)

映画メタデータ

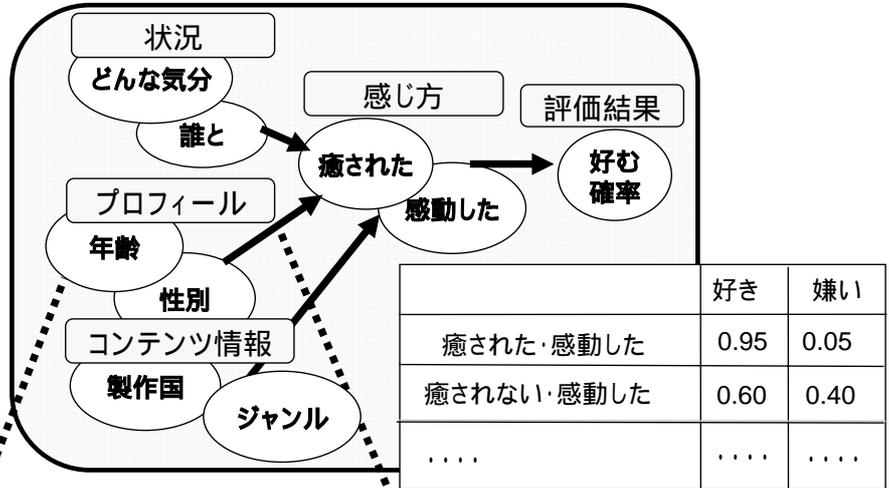
モデル構築

ユーザに関わるデータ



- 1) 確率変数をノード、
- 2) 変数間の定性的依存関係を非循環の有向リンク、
- 3) 変数間の定量的依存関係を条件付確率表で表現した確率モデル

ベイジアンネットで表現した映画に関するユーザ嗜好モデルの例

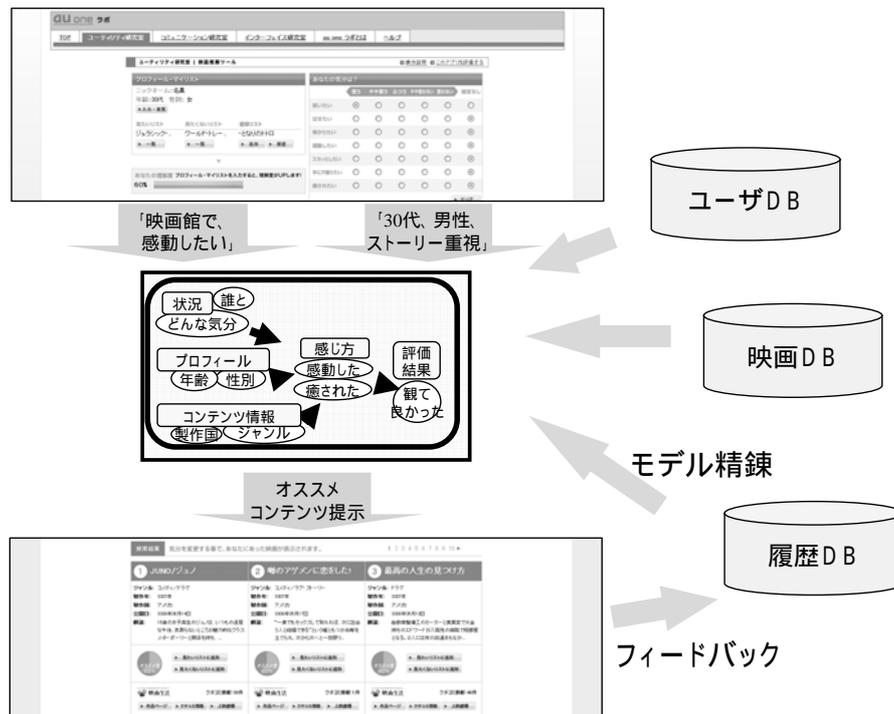


例: ユーザ嗜好に関わる各種データ

例: “女性だと癒されたと感じやすい”

良い評価を受ける確率の高い映画を推薦

= 評価の事後確率 $P(\text{評価}=\text{Positive} | \text{ユーザ属性, 状況属性, 映画属性})$ で比較

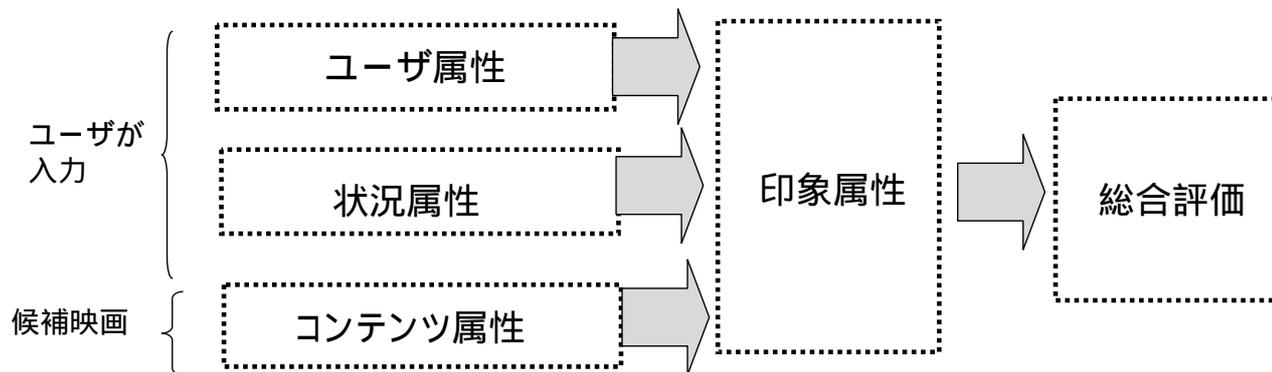


映画嗜好のベイジアンネットモデルの構築

1. ネットワーク構造(ノード間のリンク関係)の決定
2. 条件付確率表の決定

KDDI研究所のアプローチ: ハイブリッドアプローチ

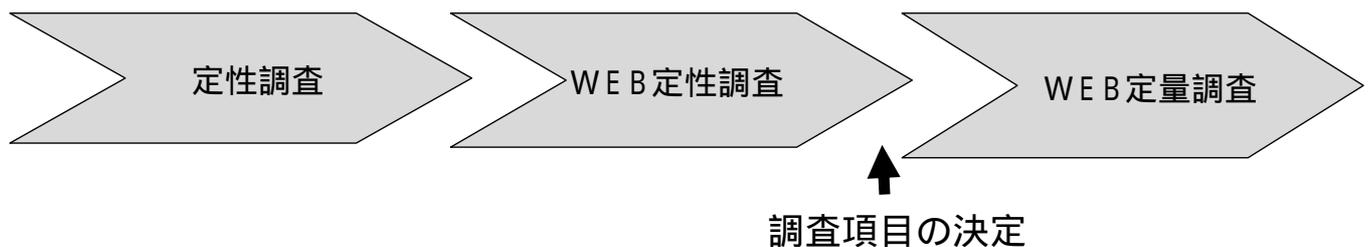
- ネットワーク構造の大枠(変数の親子関係)を領域知識から決定
 - 「商品の評価構造」= 総合評価は印象の影響を受け、印象は他の属性の影響を受ける
- ネットワーク構造の詳細(変数選択、相互依存性評価)と条件付確率表は学習データを使って決定
 - 学習データとしてアンケートデータを収集・利用



53

方針:

- コンテンツ(視聴した映画)に対するユーザの評価、印象、見た状況、およびユーザの属性を詳しく収集



定性調査: 小数の被験者にインタビューを実施。ユーザの視点で評価用語を抽出する

WEB定性調査: 大規模の被験者に自由記述アンケート。評価用語を抽出する。

WEB定量調査: 大規模の被験者に選択式アンケート。学習のための定量データ。

54

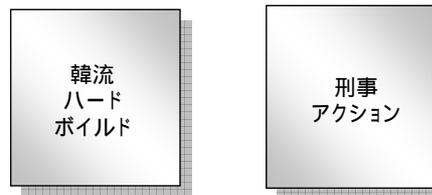
定性調査の実施例

被験者インタビュー: 映画に関する意識調査

目的: ユーザの視点で映画に対する評価用語を洗い出す
 のめりこめる、俳優が好み、映像がきれい……

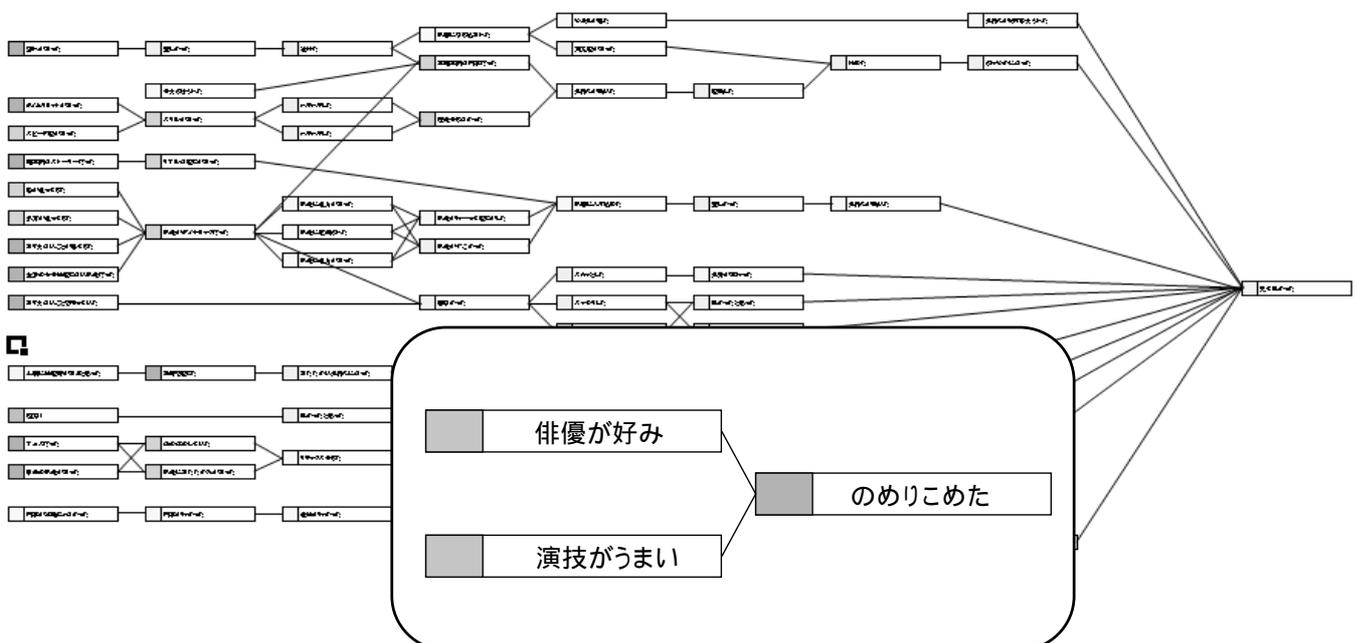
実施例: 半構造化面接法(ラダリング法、評価グリッド法など)

評価対象のペアを提示し、評価の理由を聞き出す



Q: 右と左どちらがいいですか?
 Q: それはなぜですか?
 Q: どうしてそう思うのですか?
 ……

被験者インタビュー結果の例



- 定量調査概要

調査形態 WEBアンケート

調査時期 2006年3月

- 被験者

- 採用条件 映画鑑賞頻度が比較的高い被験者

- 被験者数 2,153人

- 映画作品数 197作品

- 調査方法

- 各被験者の197作品の視聴経験を調査し、視聴経験のある作品からランダムに5~10作品を提示し評価

- 調査項目

- ユーザ属性

- デモグラフィック・ライフスタイル属性(30属性): 年齢、性別、居住地、職業、...

- 映画視聴に関する態度属性(32属性:7段階): 好きなジャンル、映画を選ぶときの重視項目など

- ユーザの映画評価: 1作品あたり以下の質問

- 鑑賞時の状況属性(43属性): 誰と観たかなど

- 鑑賞時の印象属性(358属性、7段階): 感動した、俳優が好み等

- 鑑賞後の総合評価(1属性、7段階) 観て良かった、やや観てよかった、

- 映画属性

- ジャンル、長さ、製作国、俳優、監督、映画レビューから抽出したキーワードなど

- 構造探索

- 各グループの重要属性間の親子関係の探索

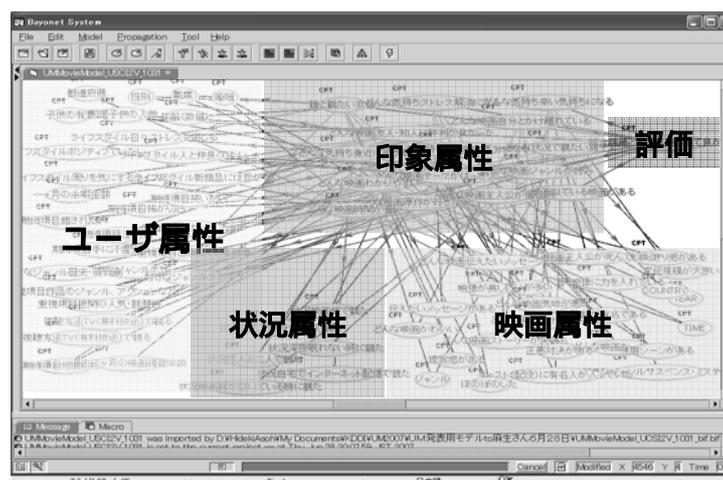
- 候補となる構造からデータに適合する構造を選択

- 適合度: AIC、MDL等で判定

- 条件付確率表の推定

- 決定した構造に基づき学習データを利用して推定

- 出来上がったモデル(75 ノード(属性), 115 リンク)



評価条件

- モデル化に用いたデータ

- ユーザ数: 2,153人
- 映画作品数: 197件
- 評価データ数: 7,019件
- 一人あたりの評価データ数: 平均3.26作品(比較的少ない= 協調フィルタリングには不向き)

- 精度評価に用いたデータ(モデル化に利用していないデータ)

- 1,754人分の評価データ

- 評価指標

- 指標1: 推薦結果が出たユーザ数と割合
- 指標2: 予測評価の予測誤差

評価モデル	評価1: 推薦結果が出た人数(割合)	評価2: 予測評価の予測 誤差(MAE)	
協調フィルタリング方式1(User-Based)	72件(4.1%)	0.975	(悪い)
協調フィルタリング方式2(Item-Based)	221件(12.6%)	0.930	
ベイジアン方式1:UC-V	1754件(100%)	0.887	
ベイジアン方式2:UCS-V	1754件(100%)	0.869	
ベイジアン方式3:UC-I-V	1754件(100%)	0.862	
ベイジアン方式4:UCS-I-V	1754件(100%)	0.854	(良い)

(注: 5段階評価を予測したので、5を1と予測すると、誤差は4となる)

評価1

× 協調フィルタリング方式は推薦結果を出せないユーザが多い
(履歴のない(少ない)ユーザ、コンテンツは扱えない)
ベイジアンネットモデルに基づく方式は全ユーザに対して推薦結果を出せた

評価2

状況(S)を考慮したベイジアンネットモデルによる
方式が最も良い性能を示した

まとめ

本日の講演内容:

- イントロ
 - 情報系・工学系の視点から
- 事例1:コンテンツベースの事例(音楽)
- 事例2:コンテンツベースの事例(ブログ)
- 事例3:履歴ベースの事例
 - クロスメディアレコメンデーション
- 事例4:属性ベースの事例
 - シーン別映画推薦

ご質問は KDDI研究所の小野までお気軽に ono@kddilabs.jp

関連文献

イントロ

[1] ACM International Conference on Recommender Systems: <http://recsys.acm.org/index.html>

[2] Recommenders06: <http://recommenders06.com/>

コンテンツベース方式(音楽)

[3] K. Hoashi, K. Matsumoto, N. Inoue: "Personalization of user profiles for content-based music retrieval based on user preferences," Proc. ACM Multimedia '03, pp. 110-119, 2003.

[4] K. Hoashi, K. Matsumoto, F. Sugaya, H. Ishizaki, J. Katto: "Feature space modification of content-based music retrieval based on user preferences," Proc. ICASSP '06, Vol. V, pp. 517-520, 2006. 他

履歴ベース方式

[5]柳原 正, 帆足 啓一郎, 松本 一則, 菅谷 史昭, "潜在クラスを利用した クロスメディアレコメンデーション方式の提案", 情報処理学会第68回全国大会, 2006, No.3, pp.1-2

属性ベース方式

[6] 小野, 本村, 麻生, "移動端末におけるユーザの状況を考慮した嗜好抽出技術", 情報処理学会誌 2007年9月号 特集 利用者の好みをとらえ活かす ~ 嗜好抽出技術の最前線 ~

[7]小野, 黒川, 本村, 麻生, "ユーザ嗜好の個人差と状況差を考慮した映画推薦システムの実現と評価", 情報処理学会論文誌, Vol.49, No.1, pp.130-140, 2008

[8] C. Ono, M. Kurokawa, Y. Motomura, and H. Asoh, "A. Context-aware movie preference model using a Bayesian network for recommendation and promotion", Proc. User Modeling 2007 11th International Conference, UM 2007, Corfu, Greece, July, 2007, Proceedings, Lecture Notes in Computer Science 4511 pp. 247-257, 2007