順序例からの学習

神嶌敏弘　赤穂昭太郎 (産業技術総合研究所)

「順序例からの学習」の提案

Learning from Order Examples; LOE)

順序を対象とした学習問題

・順序：何かの基準の順に整列したもの

例：a, b, c の順に好きである a ≻ b ≻ c

bよりaが好きだが，どれくらい好きかは分からない

・順序付けされたアイテム集合が訓練事例

未整列アイテム集合の順序を推定する規則の学習

・この学習タスクの解法を幾つか示し，人工データへの適用実験により，各解法の特徴を解析

順序を用いることの利点

例：ものの嗜好に関する調査

SD(Semantic Differential)法：般的な方法

被験者に「好き」から「嫌い」までを，例えば5段階で評価してもらう方法

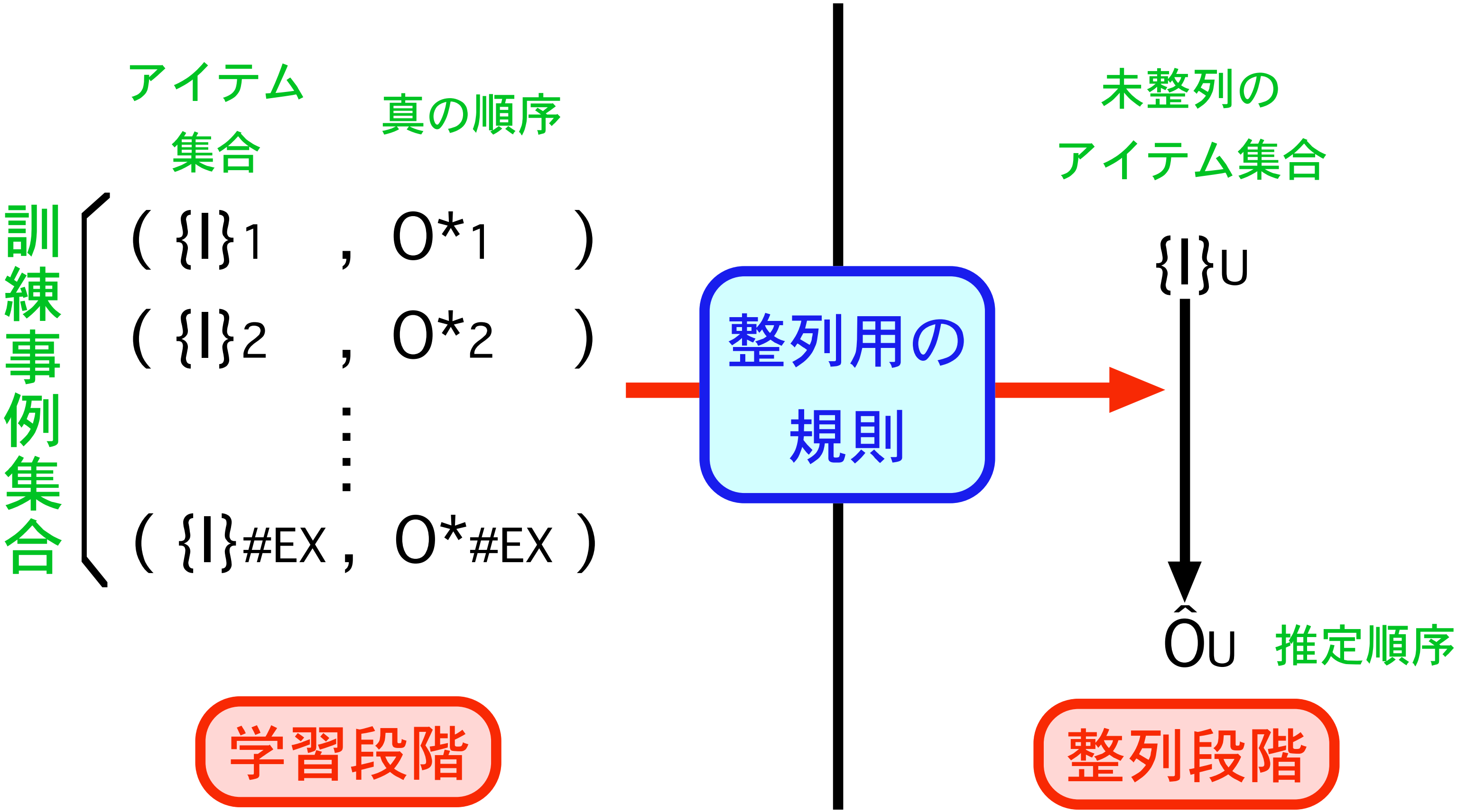
・3と4のどちらが適切か，被験者が迷う

順位法

被験者に「好き」な方から順にアイテムを並べてもらう方法

・絶対的な評価が困難な場合でも，相対的評価なら可能なことも

順序例からの学習



各アイテムは属性ベクトルで記述

➡ 訓練事例にないアイテムも整列可能

順序例からの学習の仮定と順位相関係数

仮定：真の順序(学習事例中の順序) ＝
　　絶対順序(一般的な部分) + ノイズ(個性の部分)

絶対順序：アイテムの全集合にある全順序

※ アイテムの全集合は学習事例中に現れないアイテムも含む

絶対順序に対するエラーが小さくなるような推定順序を求める

スピアマンの順位相関係数(RCI)：順序エラー尺度

二つの順序の間の，全体的な類似性を示す

0なら無関係，1なら一致，－1なら逆順

関連研究

◇[流郷01] [佐藤 95] 比較やマッチングに順位相関係数を利用．順序の予測はしない

◇[Cohen 99]

・前提：アイテムの集合が与えられる

・教師信号：二つのアイテムのどちらが前にあるかの関係を示した情報

・出力：教師信号で示された，二つのアイテム間の関係情報を，できるだけ保存した順序

[Cohen 99]の研究との相違点

◇エラーの評価

関数$\mathrm{PREF}(I^x, I^y)$：2アイテム間の順序関係の確信度

Cohenら は，関数 PREF のエラーについては議論しているが，最終的な順序のエラーについては議論していない

・関数 PREFのエラーをいくら小さくしても，順序が正しくなるかは保証されない

・教師信号が2個のアイテムの関係だけなので，順序全体の評価は本質的に不可能

手法SC: クラス分類手法を用いた解法1

SumClass(SC)法：Cohenの問題に変換して解く

- 学習段階：学習事例中の順序 O_i を，アイテム対の順序関係に分解
- 分解した順序関係から関数 PREF を推定
- 整列段階：次式を最大にする順序の獲得が目標

$$\sum_{x,y:I^x \succ I^y} \mathrm{PREF}(I^x, I^y)$$

- PREF の和に基づき，一番前にあるアイテムを推定し，それを一つずつ取り出して順序に追加

手法PC: クラス分類手法を用いた解法2

ProdutClass(PC)法：SC法の派生手法

- 関数 PREF の総和ではなく，次式を最大にする順序の獲得が目標

$$\prod_{x,y:I^x \succ I^y} \mathrm{PREF}(I^x, I^y)$$

- Cohenが用いた総和は完全なヒューリスティックス
- この関数 PREF の積は，二つのアイテムの順序関係が独立とした場合の尤度

手法R: 回帰分析手法を用いた解法

Regression(R)法：回帰分析を用いた手法

- 学習段階：学習事例中のアイテムを，真の順序となるべく矛盾しないような全順序に整列
- アイテムの属性から，この全順序中の順位を推定する関数 $f(A(I^x))$ を回帰分析で求める

- 整列段階：関数 $f(A(I^x))$ の値に従って，アイテムを整列

実験手法

・アイテムの全集合＝9種

属性値数＝3，5，7　　　属性数=3，4，5

・絶対順序と全集合の対＝90種

各アイテム全集合に10種類の絶対順序を，ランダムに重みを決めた線形関数に従い決定

・事例集合 810種

アイテム集合の大きさ＝3，5，10

事例数=10，30，50

※予稿では属性値数=7の場合を含まないので，以後の実験の数値は予稿と異なる

ノイズのない場合の実験結果

ノイズのない場合：絶対順序と真の順序は無矛盾

・絶対順序と推定順序の間のRCIの平均
・アイテム集合の大きさごとにまとめた結果と全体の結果

	全体	3	5	10
SC法	0.808	0.667	0.825	0.932
PC法	0.808	0.667	0.825	0.932
R法	0.802	0.617	0.837	0.950

・アイテム集合の大きさが大きいほど結果は精度が向上
・大きさ10では，危険率1%でも有意な相関が平均的に観測される

三種類の手法の比較

3手法の比較：RCIの差を対応のある*t*-検定で調査

	全体	3	5	10
SC－PC	- 0.1472	- 0.2709	+0.3806	- 1.5912
SC－R	+1.4430 > t0.90	+4.4143 > t0.99	- 2.2272 < t0.03	- 8.5784 < t0.01
PC－R	+1.4626 > t0.90	+4.4254 > t0.99	- 2.3547 < t0.01	- 8.5023 < t0.01

黒字=差なし 青字=前者優位 赤字=後者優位

>t0.99は99-パーセンタイル以上，<t0.01は1-パーセンタイル以下

・手法SCとPCには差がない

・手法Rは，アイテム集合の大きさが小さいときは他の手法に劣るが，大きくなると逆に優位になる

ノイズがある場合

隣接したアイテムが入れ替わるノイズ

- ３手法とも，10%の対が入れ替わった場合，RCIの低下は3%程度
- ノイズの影響は，どの手法も差はない

属性値が変化するノイズ

- ３手法とも，属性の値が１０％で変化した場合，RCIの低下は4～5%程度
- ノイズのレベルが低い場合に，他の手法と比べて，R法は性能の低下が若干小さい

SC法とPC法での最適解との比較

評価関数を最大にする順序を欲張り法で探索

➡ 最適解に対する性能低下の度合いは？

〈最適解のRCI〉－〈欲張り法のRCI〉 のt-値

SC法: − 2.7915　　PC法: −2.9306

順序のRCIは，評価関数を最適にする手法の方が，危険率1%でも有意に低い，すなわち，精度が劣る

違う！ { アイテム対の順序関係を保存する順序
全体として，よく整列された順序

➡ Cohenの研究とLOEの差が明確に

計算量に関する考察

学習段階

・SC法とPC法：$O(\sum_{i}^{\#EX}(\# I_i)^2)$

・R法：$O((\# I_C)^3)$

SC法やPC法の方が高速

整列段階

・SC法とPC法：$O((\# I_U)^3)$

・R法：$O(\# I_U \log(\# I_U))$

R法の方が高速

まとめと今後の予定

まとめ

・新しい学習問題「順序例からの学習」を提案

・3種類の解法を提案し，人工データに適用

・3手法とも適切な順序を推定することが可能

・アイテム数が少ない場合はSC・PC法が有利，多い場合はR法が有利

・ノイズの影響は，手法間であまり差がなかった

今後の予定

・間接的ではなく，RCIを直接的に最小化する順序を推定する規則の学習方法の開発