F-039

数値を含むデータからの効率的な識別パターン発見に向けて

Towards Efficient Discriminative Pattern Mining in Hybrid Domains

亀谷 由隆 [1]

Yoshitaka Kameya

1 はじめに

識別パターン発見 (discriminative pattern mining) は興味の対象のクラスとその他の違いを特徴づけるパターンを見つけ出すタスクであり，顕在パターン発見 (emerging pattern mining) [32] やサブグループ発見 (subgroup discovery) [33] とも呼ばれる [5, 19]．識別パターン発見によって我々はデータに関する知見を得ることができるが，それだけでなく，得られた識別パターンを利用することで連関分類器 (associative classifier) [24] と総称される高精度の予測モデルを構築することもできる．

識別パターン発見における一つの大きな問題は数値データの扱いである．識別パターン発見ではクラスラベルが与えられているため，Fayyad と Irani の方法 [6] に代表される教師付き離散化が適用可能であるが，離散化手法は単属性に基づくものが多く，適切な離散化を事前に行うのは多くの場合困難である．また，「$u \leq A < v$」のような区間をアイテム $\langle A, [u, v)\rangle$ を見なした場合（A は数値属性，u と v はそれぞれ区間の左端と右端），これらのアイテムは概念束を成す．図 1 に概念束の例を示す．$\bot$ を除いた最下層の区間 $[v_i, v_{i+1})$ が基本区間と呼ばれ，データ中の生の数値はこの基本区間のどれかに属する．一方，その基本区間を包摂 (subsume) する上位区間が概念束の上方に設けられる．データ中の数値一つに対してもそれを満たす上位区間は数多く考えられ，その組み合わせとなるパターンの探索空間は非常に大きくなる．従ってアルゴリズム上の工夫がなければ実時間内で全パターンを列挙するのは困難である．いくつかの既存手法 [3, 8] では連続属性の数に上限を設け，その状況固有の性質を利用して探索を大幅に高速化している．

本研究では，数値データと記号データが混在したトランザクションデータに対して FP-growth [12] に基づく識別パターン発見手法を提案する．元々FP-growth は頻出パターン発見手法として知られているが，我々は分枝限定法を導入して FP-growth を識別パターン発見用に拡張する．FP-growth は FP-tree と呼ばれるデータ構造にトランザクションデータを格納し，FP-tree の構築を繰り返しながら深さ優先探索を行っており，探索木の深い地点に進むにつれてこの FP-tree が縮小されるため，効率良く探索できる．この仕組みは再帰的データベース縮約 [31] の一種と見なせる．また，FP-tree は水平配置 (horizontal layout) と垂直配置 (vertical layout) [31] を兼ね備えたデータ構造であり，FP-tree 内のデータに対して無駄のないアクセスが実現されている．提案手法では，数値データを取り扱う際にこの FP-growth の特長を生かすような工夫がなされている．

Iris データセット [2] の各クラス c に対して提案手法が出力する識別パターン $\boldsymbol{x}$ を表 1 に示す．読みやすくするため，出力結果においては区間アイテムは不等式の形に変換している．各パターン $\boldsymbol{x}$ の良さは F 値 $\mathrm{F}_c(\boldsymbol{x})$ で測られる．この結果を見ると比較的長いパターンが出力されている．最小限の一般化しか行わない飽和制約（後述）が効いていると思われる．また，クラス versicolor と virginica に対する出力結果は一通りではなく，確信度（精度）$p(c \mid \boldsymbol{x})$ と正サポート（再現率）$p(\boldsymbol{x} \mid c)$ のバランスが異なるパターンを何通りか出力している．なお，最良カバー制約（後述）により，正事例が全て表 1 のパターンでカバーされることは保証されている．

本論文は以下の構成をとる．まず 2 節で用語・記法の導入と背景の説明を行う．3 節では提案手法の詳細を記述する．4 節ではプロトタイプ実装を動作させた実験の結果を示す．5 節で本論文のまとめを行い，今後の課題を述べる．

2 準備

2.1 アイテム

はじめにいくつかの用語や記法を導入する．まず，入力データセットは表形式であるとする．すなわち，各事例は属性 A とその値 v のペア $\langle A, v\rangle$ の集合によって記述される．属性は記号属性と数値属性の 2 種類あり，属性 A が記号属性の場合，v は A が固有にもつ記号値の有限集合から一つ選ばれる．属性 A が数値属性の場合，v は何らかの実数値である．

提案手法ではこの表形式の入力をトランザクション形式に変換したものからパターン発見を行う．まず，アイテムは記号アイテムと区間アイテムの 2 種類を用意する．記号アイテムは表形式データ中の記号属性 A とその値 v のペア $\langle A, v\rangle$ をそのままアイテムと見なしたものである．従来のアイテム i は任意の定数 $\bullet$ を導入し，$\langle i, \bullet\rangle$ と置き直して考えればよい．一方，区間アイテムは数値属性 A と区間 $[u, v]$ のペア $\langle A, [u, v)\rangle$ であると考える．区間アイテムは基本区間アイテムと上位区間アイテムの 2 種類から成る．基本区間アイテムは相異なる実数軸上の点 $v_1, v_2, \ldots, v_{n-1}$ で分割した n 区間のそれぞれに対応した $\langle A, [v_i, v_{i+1})\rangle$ である．ここで $i = 0, 1, \ldots, n$ であり，$v_0 = -\infty$, $v_n = \infty$ である．数値属性 A の分割点集合 $\{v_1, v_2, \ldots, v_{n-1}\}$ を Δ_A とおく．上位区間アイテムは $0 \leq i \leq n$, $0 \leq j \leq n$, $i < j$ を満たす $\langle A, [v_i, v_j)\rangle$ である [3]．ただし，$\langle A, [-\infty, \infty)\rangle$ はパターンの中に含めないよう考える．$v_i \leq v_{\mathrm{raw}} < v_{i+1}$ を満たすとき，表形式データ中の数値属性 A とその値 v_{raw} は基本区間アイテム $\langle A, [v_i, v_{i+1})\rangle$ に置き換えられる．各属性 A の分割点集合 Δ_A は入力データの内容に基づき定めることにする（3.1 節）．

2.2 トランザクション

変換後のトランザクション形式のデータセット $\mathcal{D}$ の内容は $\{t_1, t_2, \ldots, t_N\}$ と表記される．ここで t_i $(1 \leq i \leq N)$ は i 番

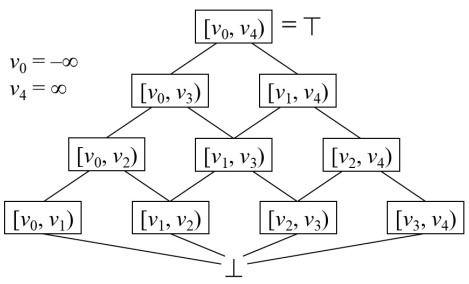

図 1: 区間に関する概念束（基本区間数 $n = 4$）．

[1] 名城大学理工学部情報工学科

[2] http://archive.ics.uci.edu/ml/datasets/Iris

[3] 区間アイテムを実装する際には添え字 i, j を使った $\langle A, i, j\rangle$ のような表現を使った方が扱いやすい．

表 1: Iris データセットから提案手法によって得られる識別パターン．

クラス c	$p(c \mid \boldsymbol{x})$	$p(\boldsymbol{x} \mid c)$	$\mathrm{F}_c(\boldsymbol{x})$	パターン $\boldsymbol{x}$
setosa	1.000	1.000	1.000	{petal_len $<$ 2.45, petal_wid $<$ 0.8, sepal_len $<$ 5.85, 2.25 $\leq$ sepal_wid}
versicolor	1.000	0.940	0.969	{2.45 $\leq$ petal_len $<$ 4.95, 0.8 $\leq$ petal_wid $<$ 1.65, 4.85 $\leq$ sepal_len $<$ 7.05, sepal_wid $<$ 3.45}
	0.942	0.980	0.961	{2.45 $\leq$ petal_len $<$ 5.15, 0.8 $\leq$ petal_wid $<$ 1.75, 4.85 $\leq$ sepal_len $<$ 7.05, sepal_wid $<$ 3.45}
	0.891	0.980	0.933	{2.45 $\leq$ petal_len $<$ 5.05, 0.8 $\leq$ petal_wid $<$ 1.85, 4.85 $\leq$ sepal_len $<$ 7.05, sepal_wid $<$ 3.45}
virginica	0.958	0.920	0.939	{4.45 $\leq$ petal_len, 1.65 $\leq$ petal_wid, 4.85 $\leq$ sepal_len, 2.45 $\leq$ sepal_wid $<$ 3.85}
	0.891	0.980	0.933	{4.75 $\leq$ petal_len, 1.35 $\leq$ petal_wid, 5.55 $\leq$ sepal_len, 2.10 $\leq$ sepal_wid $<$ 3.85}

目の事例の属性-値ペア集合を変換したアイテム集合であり，トランザクションと呼ばれる．また，元の入力データセットにおいて i 番目の事例が属するクラスを c_i とおく．表形式からトランザクション形式への変換方法から，t_i の中に同じ属性に関するアイテムが複数存在しないのは明らかである．

2.3 パターン

パターン $\boldsymbol{x}$ もまたアイテムの集合であり，同じ属性に関するアイテムは複数存在しない．一方，トランザクションには記号アイテムもしくは基本区間アイテムしか含まれないが，パターンには記号アイテム，基本区間アイテム，上位区間アイテムが含まれ得る．アイテムを一般的に参照する場合は $x, y, z, \ldots$ といった変数を使う．表記を簡単にするため，パターン $\boldsymbol{x}$ は文脈に応じてベクトル $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$, 集合 $\boldsymbol{x} = \{x_1, x_2, \ldots, x_n\}$, 連言 $\boldsymbol{x} = (x_1 \wedge x_2 \wedge \ldots \wedge x_n)$ のいずれかに読み替えらえる．また，アイテム x はそれ自身のみから成るパターン $\{x\}$ に読み替えられることがある．

2.4 包摂関係

アイテムの間には一般・特殊の関係が成り立つ．これを本論文では包摂 (subsumption) 関係と呼ぶ．具体的には，2 つの区間アイテム $x = \langle A, [u, v)\rangle$ と $y = \langle A', [u', v')\rangle$ について $A = A'$ かつ $u \leq u'$ かつ $v' \leq v$ が成り立つとき，x の方が y より一般的であるという意味で「x は y を包摂する」と言い，$x \succeq y$ と表記する．また簡単のため，記号アイテムについては，互いに等しいかどうかのみを考え，記号アイテム x が自身を包摂する（$x \succeq x$ が成り立つ）以外の包摂関係は考えない．更にパターンの間に包摂関係を以下のように導入する．2 つのパターン $\boldsymbol{x}, \boldsymbol{y}$ に対して，全ての $x \in \boldsymbol{x}$ について $x \succeq y$ なる $y \in \boldsymbol{y}$ が存在するとき，「$\boldsymbol{x}$ は $\boldsymbol{y}$ を包摂する」と言い，$\boldsymbol{x} \succeq \boldsymbol{y}$ と表記する．ここで定めたアイテム間，パターン間いずれの包摂関係も半順序関係である．

2.5 統計量

興味のあるクラス c に対して，$\mathcal{D}_c(\boldsymbol{x}) = \{t_i \mid c_i = c, \boldsymbol{x} \succeq t_i, 1 \leq i \leq N\}$ と定義する．クラス c に属するトランザクションの集合 $\mathcal{D}_c$ は $\mathcal{D}_c(\emptyset)$ と考えればよい．また，c 以外のクラスをまとめて一つのクラスとして扱い，$\neg c$ で参照する．例えば $\mathcal{D}_{\neg c} = \mathcal{D} \setminus \mathcal{D}_c$, $\mathcal{D}_{\neg c}(\boldsymbol{x}) = \mathcal{D}(\boldsymbol{x}) \setminus \mathcal{D}_c(\boldsymbol{x})$ 等が成り立つ．$\mathcal{D}_c$ 中の（$\mathcal{D}_{\neg c}$ 中の）トランザクションを正（負）トランザクションと呼ぶ．

提案手法では $\mathcal{D}$ から得られる経験確率を基本的な統計量として考える．例えば，クラス c の出現確率 $p(c)$ は $|\mathcal{D}_c|/N$, クラス c とパターン $\boldsymbol{x}$ の同時確率 $p(c, \boldsymbol{x})$ は $|\mathcal{D}_c(\boldsymbol{x})|/N$ で求められ，これらの確率から周辺確率（例えば $p(\boldsymbol{x})$）や条件確率（例えば $p(\boldsymbol{x} \mid c)$）を計算する．本論文では $p(\boldsymbol{x} \mid c)$ を正サポート，$p(\boldsymbol{x} \mid \neg c)$ を負サポートと呼ぶ．データセット $\mathcal{D}$ がクラス情報付きで与えられていれば，クラス c およびそれ以外のクラスの出現確率 $p(c)$, $p(\neg c)$ は定数と見なせる．

2.6 関連度

興味あるクラス c に対するパターン $\boldsymbol{x}$ の良さを測るスコアを関連度 (relevance) と呼び，$R_c(\boldsymbol{x})$ で表す．よく使われる関連度の多くは正サポート $p(\boldsymbol{x} \mid c)$ と負サポート $p(\boldsymbol{x} \mid \neg c)$ の関数となっている．例えば，F 値 $\mathrm{F}_c(\boldsymbol{x}) = 2p(c \mid \boldsymbol{x})p(\boldsymbol{x} \mid c)/(p(c \mid \boldsymbol{x}) + p(\boldsymbol{x} \mid c))$ は Dice 係数 $2p(\boldsymbol{x}, c)/(p(c) + p(\boldsymbol{x}))$ と等しく，これは更に $2p(c)p(\boldsymbol{x} \mid c)/(p(c) + p(c)p(\boldsymbol{x} \mid c) + p(\neg c)p(\boldsymbol{x} \mid \neg c))$ と変形される．規則 $\boldsymbol{x} \Rightarrow c$ のみから成る分類器を考えたとき，ROC 分析では正サポートは真陽性率 (true positive rate, TPR)，負サポートは偽陽性率 (false positive rate, FPR) と呼ばれる．また，我々はクラス c の特徴づけを考えるため，$p(\boldsymbol{x} \mid c) \geq p(\boldsymbol{x} \mid \neg c)$（これは $p(c \mid \boldsymbol{x}) \geq p(c)$ と同値）が成り立つパターン $\boldsymbol{x}$ のみに注目する．

ここで，$p(\boldsymbol{x} \mid c) \geq p(\boldsymbol{x} \mid \neg c)$ を満たす任意のパターン x について，$R_c(\boldsymbol{x})$ が $p(\boldsymbol{x} \mid c)$ に関して単調増加し，$p(\boldsymbol{x} \mid \neg c)$ に関して単調減少するとき，関連度 R_c は双単調 (dual-monotonic) であるという [15]．以降では，関連度は双単調性を満たすものとし，双単調性に基づいて枝刈りや冗長性排除の条件を導出する．F 値，χ^2 値，情報利得，support difference 等，識別パターン発見で用いられる多くの関連度が双単調性を満たす．従来は関連度には凸性 [20, 30] が仮定されていたが，双単調性はより緩い条件となっている．例えば F 値は凸性を満たさない．また，双単調性は Piatetsky-Shapiro が唱える関連度が満たすべき 3 条件 [9, 22] のうち 2 つ分に相当する [4]．

2.7 分枝限定法に基づく上位 k パターン発見

関連度上位の k パターンのみ出力しようと考える．これを上位 k パターン発見と呼ぶ．パターン $\boldsymbol{x}$ を訪問している時点の k 番目のパターンを $\boldsymbol{z}$ とおく．パターンを拡大しながら探索する場合，分枝限定法 [17, 20, 30] では，上界（$\boldsymbol{x}$ を拡大して得られる関連度の上限）$\overline{R}_c(\boldsymbol{x})$ を求め，$\overline{R}_c(\boldsymbol{x}) < R_c(\boldsymbol{z})$ であれば，$\boldsymbol{x}$ 以下を枝刈りする．$\overline{R}_c(\boldsymbol{x})$ を得るには $p(\boldsymbol{x} \mid \neg c) := 0$ を代入する [5]．すると，関連度 R_c の双単調性から $\overline{R}_c(\boldsymbol{x})$ が $p(\boldsymbol{x} \mid c)$ の単調増加関数であるため，$\overline{R}_c(\boldsymbol{x})$ がパターン $\boldsymbol{x}$ の拡大に対し減少する（逆単調になる）と分かる．このとき，$\boldsymbol{x} \subset \boldsymbol{x}'$ なる $\boldsymbol{x}'$ に対して $R_c(\boldsymbol{x}') \leq \overline{R}_c(\boldsymbol{x}') \leq \overline{R}_c(\boldsymbol{x}) < R_c(\boldsymbol{z})$ が成り立つ．従って上述の枝刈りは安全である．

更に，$\overline{R}_c(\boldsymbol{x})$ が $p(\boldsymbol{x} \mid c)$ の単調増加関数であることを利用して，枝刈り条件 $\overline{R}_c(\boldsymbol{x}) < R_c(\boldsymbol{z})$ を $p(\boldsymbol{x} \mid c)$ について解き [6]，$p(\boldsymbol{x} \mid c) < U_c(\boldsymbol{z})$ の形の不等式を得ることができる．例えば，$\overline{\mathrm{F}}_c(\boldsymbol{x}) = 2p(\boldsymbol{x} \mid c)/(1 + p(\boldsymbol{x} \mid c))$ であり，$\overline{\mathrm{F}}_c(\boldsymbol{x}) < \mathrm{F}_c(\boldsymbol{z})$ を解いて枝刈り条件 $p(\boldsymbol{x} \mid c) < \mathrm{F}_c(\boldsymbol{z})/(2 - \mathrm{F}_c(\boldsymbol{z}))$ が得られる．そして，その後の探索で新たな k 番目のパターン $\boldsymbol{z}'$ ($R_c(\boldsymbol{z}') > R_c(\boldsymbol{z})$) を見つけたら $\sigma_{\min} := U_c(\boldsymbol{z}')$ として，以降訪問するパターン $\boldsymbol{x}$ について $p(\boldsymbol{x} \mid c) < \sigma_{\min}$ が成り立て

[4] 残りの一つは $p(\boldsymbol{x} \mid c) = p(\boldsymbol{x} \mid \neg c)$ のとき $R_c(\boldsymbol{x}) = 0$.

[5] TPR が変化せずに FPR が 0 になるという最も楽観的な状況を考えることに相当する．

[6] 多くの関連度では解析的に解けるが，情報利得等，数値的に解かざるを得ないと思われる関連度も存在する．

ば $\boldsymbol{x}$ 以下を枝刈りする（FP-growth では FP-tree を縮小する）．$U_c(\boldsymbol{z})$ は $R_c(\boldsymbol{z})$ に関して単調増加するため，探索が進むにつれて $\sigma_{\min}$ が上昇していく．

最小サポート上昇 (minimum support raising) [13, 31] は元々頻出パターン発見で導入された技法だが，このように双単調性を満たす関連度に基づく識別パターン発見でも利用できる．上界 $\overline{R}_c$ そのものを計算・維持する必要がないため，FP-growth をはじめとする頻出パターン発見手法をベースにできるなど，アルゴリズムの設計・実装を簡素化できる．

2.8 パターン間の制約

出力されるパターン間の冗長性も識別パターン発見における問題の一つである．例えばアイテム x が興味あるクラス c と強く関連する場合，x を含むパターン $\{x, y\}$, $\{x, z\}$, $\{x, y, z\}$ 等のパターンもまた c と関連しやすく，関連度の上位が x を含む似たパターンで占められる場合がある．それを防ぐためにパターン間に制約を設け，その制約に違反するパターンを冗長であるとして削除することが行われる．

頻出パターン発見におけるパターン間の制約の代表例として飽和 (closedness) 制約 [21] が知られる．一方，識別パターン発見では正トランザクション上の飽和制約（正の飽和制約）[10] を考えることが多い．すなわち，カバーする正トランザクションの集合が同一であるパターンの集合（同値類）の中で関係 $\succeq$ に関して最も特殊なもののみを残す．

最良カバー (best-covering) 制約 [14] は「出力パターンはそれがカバーする正トランザクション t のいずれかにおいて，t をカバーするパターンの中で最大の関連度を持たなければいけない」という制約である．この制約は連関分類器の一つとして知られる HARMONY [26] で使われている highest confidence covering という制約が対象とする関連度（確信度 $p(c \mid \boldsymbol{x})$）を双単調性を満たす関連度に一般化したものである．最良カバー制約は識別パターン発見でよく使われる生産性 (productivity) 制約 [2, 17, 28] と比べて同等かそれ以上に強い制約であることが分かっている [14].

双単調性を満たす関連度の下では，正の飽和制約を満たすパターンが同値類の中で最大の関連度を持つため [10, 23]，正の飽和制約と最良カバー制約は整合することが多い．ただし，ある正トランザクションで同じ最大関連度を持つパターン $\boldsymbol{x}$ と $\boldsymbol{x}'$ が存在し，$\boldsymbol{x}$ の方がより特殊であるとき，正の飽和制約は $\boldsymbol{x}$ を好み，最良カバー制約は $\boldsymbol{x}'$ を好む．この場合は正の飽和制約を優先することを考える [14].

2.9 全被覆法

規則学習の標準的手法として逐次被覆法（sequential covering, あるいは単に被覆法） [7, 9, 29] が知られる．この方法では，正クラスを後件部に持つ規則を抽出した後にその規則にカバーされる正事例を全て削除してから別の規則を探索しに行く [7]．しかし，抽出された規則にカバーされる正事例を削除するという作業は手続き的であり，最終的に得られた規則の集まりを宣言的に解釈できなくなる．また，事例の削除は統計的な信頼性を損ねるという問題もある．そのため，Domingos らは conquering without separating という戦略で各規則を全事例から学習する方法を提案している [4].

全被覆法 (exhaustive covering) [14] は conquering without separating と同様の意図で提案されている．全被覆法では最良カバー制約の利用を前提とし，正トランザクションの各々に対して同時並行で上位 1 パターン発見を行う．具体的には，まず正トランザクション t の各々に候補リスト $L[t]$ を用意する．そして，現在訪問中のパターン $\boldsymbol{x}$ に対して，$\boldsymbol{x}$ がカバーする正トランザクション t の各々について候補リスト $L[t]$ を参照し，$L[t]$ が空なら $L[t]$ に $\boldsymbol{x}$ を登録する．$L[t]$ が空ではないなら，$L[t]$ 内の暫定の上位 1 パターン $\boldsymbol{z}$ と関連度を比較する．$\boldsymbol{x}$ の関連度が高ければ $L[t]$ 内のパターンをすべて削除した後に $\boldsymbol{x}$ のみを登録し，等しければ $L[t]$ に $\boldsymbol{x}$ を追加する．$\boldsymbol{x}$ の方が関連度が低ければ $L[t]$ に対して何も行わない．パターンに訪問する度に以上を繰り返す．その後，探索が終了したら $L[t]$ の集合和をとり，出力パターンの集合とする．また，訪問中のパターン $\boldsymbol{x}$ について，$\boldsymbol{x}$ がカバーする正トランザクション t の候補リスト $L[t]$ の上位 1 パターン $\boldsymbol{z}$ の関連度の中で最小のものから計算される $\sigma_{\min}$（2.7 節）を使い，分枝限定法の枝刈りが行われる．

枝刈りの結果，全ての正トランザクションについて上位 1 パターンが確定した時点で探索が終了する．上位 k パターン発見では，出力してほしいパターン数 k をユーザが指定すれば敏感な閾値である $\sigma_{\min}$ を指定しなくてよいという利点があったが，更に全被覆法では k も指定せずに済んでいる．

2.10 動的優先付け

2.7 節の記述から分かるように，上位 k パターン発見においては関連度の高いパターンが早く見つかるほど効果的な枝刈りが可能になる．従って，動的優先付け (dynamic reordering) [1, 16, 27] を行って探索効率の向上を図ることは合理的である．また，探索空間が膨大なときには探索を有限時間で打ち切らなければならない．その際にも動的優先付けを行えば早期に関連度の高いパターンが得られる可能性が高いため，anytime アルゴリズムとして利用することができる．

3 提案手法

本論文では，FP-growth をベースとする全被覆法に基づき，記号データと数値データが混在するデータセットから識別パターンを発見する方法を提案する．その際に，最良カバー制約と飽和制約を利用することでパターン間の冗長性を抑える．基本的な探索手順は 2.7 節から 2.9 節にかけて説明した通りであるので，以下では数値データに対して特別処理している箇所を順に説明する．

3.1 区間アイテムの生成

2.1 節で述べたように，入力データセットをトランザクション形式に変換する際に数値属性 A の値を基本区間アイテムに変換するための分割点集合 Δ_A を定める必要がある．

まず，各数値属性 A について入力データセットに出現する値を昇順に並べ，隣り合う値の中点を初期の分割点とする．その際，事前に十分小さい刻み幅 ε を用意しておき，$[m\varepsilon, (m+1)\varepsilon)$ の範囲に入る値は全て同一の値と見なす．すると，分割点間の領域には 1 つ以上の値が存在することになる．そして，各々の値 v について，入力データセット中の i 番目の事例において属性 A が値 v をとるとき，値 v をクラス c_i と紐づけできる．このとき，分割点間の領域には (i) 正クラスの値のみが含まれる場合，(ii) 負クラスの値のみが含まれる場合，(iii) 正負両クラスの値が混合する場合の 3 通りになる．提案手法では (i) の領域が 2 つ以上連続する場合，それらの領域を可能な限り大きく併合する．(ii) の領域が連続する場合も同様に可能な限り大きく併合する．この併合が全て終わった後の分割点を Δ_A とする．

この併合操作は Brin らによって最初に導入され [3]，後に Grosskreutz らも使っている [11]．一方，提案手法ではこの併合操作を関連度の双単調性と最良カバー制約によって一般的に正当化できる．まず (i) の併合について，正クラスの値のみを含む併合済みの連続領域を $[u, v)$ とおく．そして $v_l \leq u < t < v \leq v_r$ なる他の分割点 v_l, t, v_r を考える．t は連続領域 $[u, v)$ の間に存在する分割点である．このとき，

$$\begin{aligned} p(\langle A, [v_l, t)\rangle \cup \boldsymbol{x} \mid c) &< p(\langle A, [v_l, v)\rangle \cup \boldsymbol{x} \mid c) \\ p(\langle A, [v_l, t)\rangle \cup \boldsymbol{x} \mid \neg c) &= p(\langle A, [v_l, v)\rangle \cup \boldsymbol{x} \mid \neg c) \end{aligned}$$

[7] 連関分類器の構築時に規則削減の目的で使われるデータベース被覆 (database coverage) と呼ばれる方法も類似した処理を行う [24].

$$p(\langle A,[t,v_r)\rangle \cup \boldsymbol{x} \mid c) \quad < \quad p(\langle A,[u,v_r)\rangle \cup \boldsymbol{x} \mid c)$$
$$p(\langle A,[t,v_r)\rangle \cup \boldsymbol{x} \mid \neg c) \quad = \quad p(\langle A,[u,v_r)\rangle \cup \boldsymbol{x} \mid \neg c)$$

が成り立つ（$\boldsymbol{x}$ は任意のパターン）．関連度の双単調性と最初の 2 式から $R_c(\langle A,[v_l,t)\rangle \cup \boldsymbol{x}) < R_c(\langle A,[v_l,v)\rangle \cup \boldsymbol{x})$ が言えるが，最良カバー制約の下では特殊なパターンの関連度が一般的なパターンの関連度を下回る場合，特殊なパターン $\langle A,[v_l,t)\rangle \cup \boldsymbol{x}$ の方が制約違反となる [14]．同様に最後の 2 式から $\langle A,[t,v_r)\rangle \cup \boldsymbol{x}$ も最良カバー制約に違反する．従って，t を分割点として残す意味がないことが分かる．(ii) の併合の場合も同様の議論で正当化される．

3.2 区間アイテムの関連度の計算

識別パターン発見用の FP-growth [17] では訪問中のパターン $\boldsymbol{x}$ に対する条件付き FP-tree を構築する際に，条件付きトランザクション集合に出現するアイテム x の正負の出現回数をカウントし，各々 $N_c(x \cup \boldsymbol{x})$, $N_{\neg c}(x \cup \boldsymbol{x})$ と解釈する．そしてこの出現回数から正サポート $p(x \cup \boldsymbol{x} \mid c)$, 負サポート $p(x \cup \boldsymbol{x} \mid \neg c)$[8]，更には関連度 $R_c(\boldsymbol{x})$ が計算される．

条件付きトランザクション集合に出現するのは記号アイテムと基本区間アイテムのみであるため，上位区間アイテムについては図 1 のような概念束上の動的計画法に基づき [9]，正負のサポートを計算する．これを具体的に述べると，$\Delta_A = \{v_1, v_2, \ldots, v_{n-1}\}$, $v_0 = -\infty$, $v_n = \infty$ とおいたとき，$d = 2, 3, \ldots, n-1$ の各々について順に正サポートを

$$p(\langle A,[v_i,v_{i+d})\rangle \cup \boldsymbol{x} \mid c) = p(\langle A,[v_i,v_{i+1})\rangle \cup \boldsymbol{x} \mid c) + p(\langle A,[v_{i+1},v_{i+d})\rangle \cup \boldsymbol{x} \mid c)$$

によって計算する $(0 \leq i \leq n-d)$．負サポートも同様である．

また，条件付きトランザクション集合に出現するアイテム x について，$\sigma_{\min}$ に基づき，正サポートが小さいものは条件付きトランザクションから削除し，FP-tree のヘッダ表にも含めない．こうすると以降の探索で $x \cup \boldsymbol{x}$ を訪問することはなくなる．x が記号アイテムなら単純に $p(x \cup \boldsymbol{x} \mid c) < \sigma_{\min}$ のときに削除すればよいが，x が数値属性 A の区間アイテムのとき，x を削除できるのは（x を含めた）条件付きトランザクション集合に出現する A の区間アイテムの正サポートの総和が $\sigma_{\min}$ 未満になったときのみである．

この時点で飽和制約を利用した枝刈りが可能である．すなわち，$p(\langle A,[v_i,v_j)\rangle \cup \boldsymbol{x} \mid c) = p(\langle A,[v_i,v_{j-1})\rangle \cup \boldsymbol{x} \mid c)$ もしくは $p(\langle A,[v_i,v_j)\rangle \cup \boldsymbol{x} \mid c) = p(\langle A,[v_{i+1},v_j)\rangle \cup \boldsymbol{x} \mid c)$ であるとき，上位区間アイテム $\langle A,[v_i,v_j)\rangle \cup \boldsymbol{x}$ は自身が包摂するパターンと等しい正サポートを持っていることになる．これは飽和制約への違反であるため，$\langle A,[v_i,v_j)\rangle$ は FP-tree のヘッダ表に含めないことにする．

3.3 区間アイテムを考慮した FP-tree の拡張

元々の FP-growth では条件付きトランザクションから条件付き FP-tree を構築する際に，条件付きトランザクションに出現する各アイテム x をヘッダ表 H に登録する．また，ヘッダ表 H には FP-tree 中で x の節点を繋ぐリストが同時に作成・登録される．このリストを $H[x]$ と表記する．

一方，提案手法では条件付きトランザクションには記号アイテムと基本区間アイテムしか含まれないため，ヘッダ表に上位区間アイテムを登録し，対応するリストを別途構築する必要がある．具体的には，$\Delta_A = \{v_1, v_2, \ldots, v_{n-1}\}$, $v_0 = -\infty$, $v_n = \infty$ とおいたとき，条件付きトランザクション中の基本区間アイテム $x = \langle A,[v_i,v_{i+1})\rangle$ に対応する節点 w が FP-tree に挿入される際に，x を包摂する上位区間アイテム $y = \langle A,[v_j,v_{j'})\rangle$ $(0 \leq j \leq i$ および $i+1 \leq j' \leq n$, ただし自身と $\langle A,[-\infty,\infty)\rangle$ は除く）を考え，y がヘッダ表に未登録であり，かつ $p(y \cup \boldsymbol{x} \mid c) \geq \sigma_{\min}$ と飽和制約を満たすなら，y を登録し，w を $H[y]$ の最初の要素にする．既に y がヘッダ表に登録されているなら $H[y]$ の末尾に w を繋ぐ．

このときに，基本区間アイテム x ごとに可能な上位区間アイテム $y = \langle A,[v_j,v_{j'})\rangle$ を全て検査するのは無駄であるため，正サポートに基づく枝刈りを行う．すなわち，j と j' の走査範囲を工夫し，より一般的な上位区間アイテム y から検査を始め，$p(y \cup \boldsymbol{x} \mid c) < \sigma_{\min}$ だったときに，y に包摂される x の他の上位区間アイテムは検査しないようにする．

3.4 基本区間の動的な併合に基づく枝刈り

3.1 節にて，初期トランザクション集合において正（負）のみのトランザクションしか存在しない分割点間の領域を併合して基本区間アイテムを作ると述べた．これを更に進め，探索を行っている途中の条件付きトランザクションを観察してみると，正（負）のみの出現となる基本区間アイテムが出現することが分かる．これは，条件付きトランザクション集合の初期トランザクション集合に比べてサイズが小さくなり，正負の偏りが大きくなりやすいためである．

そこで，正（負）のみの出現基本区間アイテムが 2 つ以上隣接するとき，それらの基本区間アイテムを一つに併合し，対応する上位区間アイテムで置き換える．それに伴い概念束も一部変更する．この作業には一定のコストがかかるが，概念束上の関連度計算の手間は小さくなり，探索における分岐数も減るため，結局は探索効率が向上する．

3.5 飽和制約の利用

最良カバー制約を利用する既存手法 [14] では頻出飽和パターン発見で著名な LCM [25] と同様の閉包 (closure) 操作を行うことで予め正の飽和制約を満たすパターンの中から最良カバー制約を満たすものを抽出する．一方，本論文では各正トランザクションで最大関連度を持つパターンが複数存在する場合，探索中はそれらを貯めておき，探索が終わってから後処理として正の飽和制約，最良カバー制約の順で検査を行い，制約違反のパターンを削除する．閉包操作を導入しなかったのは最良カバー制約の枝刈りも強力であることと，実装の容易さを考慮したことによる．現実のデータセットに対して閉包操作の計算コストがどの程度になるか，探索空間がどの程度抑えられるか，等について今後検討が必要である．

3.6 動的優先付けにおける留意点

動的優先付け（2.10 節）を行う場合，分岐の順序は新たに追加するアイテムの種類（記号アイテム，基本区間アイテム，上位区間アイテム）によらず，単に関連度の降順としてよい [10]．一方，条件付きトランザクション内でアイテムを並べ直す順序（分岐の順序と一致しなくてもよい）には注意が必要である．すなわち，区間アイテムは異なる数値属性に属するアイテムが混ざりあって並ぶことは許されない．

4 実験

本節では提案手法のプロトタイプ実装の実行結果を示す．実装言語は Java である．入力としたのは German Credit データセット [11] である．このデータセットは信用貸付に関するもので，サイズは事例数 1000, 属性数 20（うち数値属

[8] 実際の実装では正負の出現回数のまま扱うが，ここでは論文中の他の記述に合わせている．

[9] この概念束は格子状をしているので，2 次元配列を用いた三角行列で簡潔に実装できる．

[10] Grosskreutz らの方法 [11] らの方法では記号アイテムを先にパターンに追加していかなければいけないが，提案手法ではそのような制約はない．

[11] https://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data)

性 7, 記号属性 13）であり，一部の記号属性は数値情報を予め離散化した形になっている．

このデータセットの優良顧客 (good) クラス，問題顧客 (bad) クラスの各々に対して提案手法を適用し，識別パターンを得た．使用したプロセッサは Core i7 3820 (3.6GHz) である．優良顧客 (good) クラスに対しては訪問パターン数 28,491,232 で，実行時間は 686 秒（12 分弱）であった．問題顧客 (bad) クラスに対しては訪問パターン数 2,100,728,918 で，実行時間は 24,707 秒（7 時間弱）であった．

また，両クラスに対して得られた識別パターンを表 2 に示す．ここから，借入額 (credit_amount) と借入期間 (duration) が小さければ優良顧客と見なせることが多いが，問題顧客の判断は難しく，借入額と借入期間以外にも年齢 (age)，預金状況 (savings_status)，外国人労働者かどうか (foreign_worker) 等を考慮する必要があることが分かる．

5 おわりに

本論文では，FP-growth をベースとする全被覆法に基づき，記号データと数値データが混在するデータセットから識別パターンを発見する方法を提案した．また，提案手法のプロトタイプ実装を 2 つのベンチマークデータセット (Iris, German Credit) に適用した結果を示した．今後の課題としては，他のデータセットでも試すなどして提案手法の振る舞いをより詳しく調査することがまず挙げられる．また，入力データセットから自動決定される基本区間の数が大きくなる場合に備えて，基本区間を適切な尺度に基づき併合する方法を導入することを考えている．例えば，ヒストグラム密度推定に基づく離散化手法 [18]（の教師付き版）の利用が考えられる．更に，提案手法で得られた識別パターンを用いる連関分類器の開発を行いたい．

参考文献

[1] Atzmueller, M., Lemmerich, F.: Fast subgroup discovery for continuous target concepts. In: Proc. of ISMIS-09. pp. 35–44 (2009)

[2] Bayardo, R., Agrawal, R., Gunopulos, D.: Constraint-based rule mining in large, dense databases. Data Mining and Knowledge Discovery 4, 217–240 (2000)

[3] Brin, S., Rastogi, R., Shim, K.: Mining optimized gain rules for numeric attributes. IEEE Trans. on Knowledge and Data Engineering 15(2), 324–338 (2003)

[4] Domingos, P.: The RISE system: conquering without separating. In: Proc. of ICTAI-94. pp. 704–707 (1994)

[5] Dong, G., Bailey, J. (eds.): Contrast Data Mining: Concepts, Algorithms, and Applications. CRC Press (2012)

[6] Fayyad, U.M., Irani, K.B.: Multi-interval discretization of continuous-valued attributes for classification learning. In: Proc. of IJCAI-93. pp. 1022–1029 (1993)

[7] Friedman, J.H., Fisher, N.I.: Bump hunting in high-dimensional data. Statistics and Computing 9, 123–143 (1999)

[8] Fukuda, T., Morimoto, Y., Morishita, S., Tokuyama, T.: Mining optimized association rules for numeric attributes. In: Proc. of PODS-96. pp. 182–191 (1996)

[9] Fürnkranz, J., Gamberger, D., Lavrač, N.: Foundations of Rule Learning. Springer (2012)

[10] Garriga, G.C., Kralj, P., Lavrač, N.: Closed sets for labeled data. J. of Machine Learning Research 9, 559–580 (2008)

[11] Grosskreutz, H., Rüping, S.: On subgroup discovery in numerical domains. Data Mining and Knowledge Discovery 19(2), 210–226 (2009)

[12] Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: Proc. of SIGMOD-00. pp. 1–12 (2000)

[13] Han, J., Wang, J., Lu, Y., Tzvetkov, P.: Mining top-k frequent closed patterns without minimum support. In: Proc. of ICDM-02. pp. 211–218 (2002)

[14] Kameya, Y.: An exhaustive covering approach to parameter-free mining of non-redundant discriminative itemsets. In: Proc. of DaWaK-16. pp. 143–159 (2016)

[15] Kameya, Y., Asaoka, H.: Depth-first traversal over a mirrored space for non-redundant discriminative itemsets. In: Proc. of DaWaK-13. pp. 196–208 (2013)

[16] Kameya, Y., Ito, K.: Dynamic re-ordering in mining top-k productive discriminative patterns. In: Proc. of TAAI-17. pp. 172–177 (2017)

[17] Kameya, Y., Sato, T.: RP-growth: top-k mining of relevant patterns with minimum support raising. In: Proc. of SDM-12. pp. 816–827 (2012)

[18] Kontkanen, P., Myllymäki, P.: MDL histogram density estimation. In: Proc. of AISTATS-07. pp. 219–226 (2007)

[19] Kralj Novak, P., Lavrač, N., Webb, G.I.: Supervised descriptive rule discovery: a unifying survey of contrast set, emerging pattern and subgroup mining. J. of Machine Learning Research 10, 377–403 (2009)

[20] Morishita, S., Sese, J.: Traversing itemset lattices with statistical metric pruning. In: Proc. of PODS-00. pp. 226–236 (2000)

[21] Pasquier, N., Bastide, Y., Taouli, R., Lakhal, L.: Discovering frequent closed itemsets for association rules. In: Proc. of ICDT-99. pp. 398–416 (1999)

[22] Piatetsky-Shapiro, G.: Discovery, analysis, and presentation of strong rules. In: Knowledge Discovery in Databases, pp. 229–248. AAAI Press (1991)

[23] Soulet, A., Crémilleux, B., Rioult, F.: Condensed representation of emerging patterns. In: Proc. of PAKDD-04. pp. 127–132 (2004)

[24] Thabtah, F.: A review of associative classification mining. Knowledge Engineering Review 22(1), 37–65 (2007)

[25] Uno, T., Asai, T., Uchida, Y., Arimura, H.: An efficient algorithm for enumerating closed patterns in transaction databases. In: Proc. of DS-04. pp. 16–31 (2004)

[26] Wang, J., Karypis, G.: HARMONY: efficiently mining the best rules for classification. In: Proc. of SDM-05. pp. 205–216 (2005)

[27] Webb, G.I.: OPUS: an effficient admissible algorithm for unordered search. J. of Artificial Intelligence Research 3, 431–465 (1995)

[28] Webb, G.I.: Discovering significant patterns. Machine Learning 68, 1–33 (2007)

[29] Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, 2nd edn. (2005)

[30] Zimmermann, A., De Raedt, L.: Cluster grouping: from subgroup discovery to clustering. Machine Learning 77, 125–159 (2009)

[31] 宇野毅明, 有村博紀: データインテンシブコンピューティング — その 2 頻出アイテム集合発見アルゴリズム —. 人工知能学会誌 22(3), 425–436 (2007)

[32] 加藤直樹, 羽室行信, 矢田勝俊: データマイニングとその応用. 朝倉書店 (2008)

[33] 竹村彰通（監訳）: 機械学習 : データを読み解くアルゴリズムの技法. 朝倉書店 (2017), (P. Flach: Machine Learning: The Art and Science of Algorithms that Make Sense of Data, Cambridge University Press, 2012)

表 2: German Credit データセットから提案手法によって得られる識別パターン．

クラス c	$p(c \mid \boldsymbol{x})$	$p(\boldsymbol{x} \mid c)$	$\mathrm{F}_c(\boldsymbol{x})$	パターン $\boldsymbol{x}$
good	0.714	0.987	0.829	{credit_amount $<$ 10920, duration $<$ 66}
	0.713	0.989	0.828	{credit_amount $<$ 11190, duration $<$ 66}
	0.711	0.990	0.827	{credit_amount $<$ 11790, duration $<$ 66}
	0.709	0.993	0.827	{credit_amount $<$ 12300, duration $<$ 66}
	0.706	0.997	0.827	{credit_amount $<$ 14250, duration $<$ 66}
	0.702	1.000	0.825	{credit_amount $<$ 15900, duration $<$ 66}
bad	0.411	0.647	0.503	{19.5 $\leq$ age $<$ 61.5, 430.5 $\leq$ credit_amount, 11.5 $\leq$ duration $<$ 66, exist_cred $<$ 3.5, savings_status = less_than_100}
	0.407	0.657	0.503	{19.5 $\leq$ age $<$ 66.5, 430.5 $\leq$ credit_amount, 11.5 $\leq$ duration $<$ 66, exist_cred $<$ 3.5, savings_status = less_than_100}
	0.401	0.673	0.502	{age $<$ 61.5, 430.5 $\leq$ credit_amount, 8.5 $\leq$ duration $<$ 66, exist_cred $<$ 2.5, savings_status = less_than_100}
	0.403	0.663	0.501	{age $<$ 61.5, 430.5 $\leq$ credit_amount, 7.5 $\leq$ duration $<$ 66, exist_cred $<$ 2.5, foreign_worker = yes, savings_status = less_than_100}
	0.395	0.683	0.501	{age $<$ 74.5, 605 $\leq$ credit_amount, 8.5 $\leq$ duration $<$ 66, exist_cred $<$ 2.5, savings_status = less_than_100}
	0.406	0.630	0.494	{age $<$ 61.5, 430.5 $\leq$ credit_amount, 5.5 $\leq$ duration $<$ 66, exist_cred $<$ 2.5, foreign_worker = yes, 1.5 $\leq$ install_commit, savings_status = less_than_100}
	0.388	0.680	0.494	{age $<$ 61.5, 430.5 $\leq$ credit_amount, 5.5 $\leq$ duration $<$ 66, exist_cred $<$ 2.5, foreign_worker = yes, savings_status = less_than_100}
	0.382	0.697	0.494	{age $<$ 74.5, 430.5 $\leq$ credit_amount, 5.5 $\leq$ duration $<$ 66, exist_cred $<$ 2.5, foreign_worker = yes, savings_status = less_than_100}
	0.341	0.883	0.493	{19.5 $\leq$ age $<$ 61.5, 430.5 $\leq$ credit_amount, 11.5 $\leq$ duration, foreign_worker = yes}
	0.338	0.900	0.492	{19.5 $\leq$ age $<$ 69, 430.5 $\leq$ credit_amount, 11.5 $\leq$ duration, foreign_worker = yes}
	0.332	0.933	0.490	{age $<$ 61.5, 430.5 $\leq$ credit_amount, 8.5 $\leq$ duration, foreign_worker = yes}
	0.318	0.963	0.478	{age $<$ 61.5, 430.5 $\leq$ credit_amount, 5.5 $\leq$ duration, foreign_worker = yes}