最小サポート上昇法に基づく上位 k 関連パターン発見

Mining Top-k Relevant Patterns using Minimum Support Raising

亀谷由隆 [1*]　佐藤泰介 [1]
Yoshitaka KAMEYA[1] and Taisuke SATO[1]

[1] 東京工業大学 大学院情報理工学研究科
[1] Graduate School of Information Science and Engineering, Tokyo Institute of Technology

Abstract: One practical inconvenience in frequent pattern mining is that it often yields a flood of common or uninformative patterns, and thus we should carefully adjust the minimum support. To alleviate this inconvenience, based on FP-growth, this paper proposes RP-growth, an efficient algorithm for top-k mining of discriminative patterns which are highly relevant to the class of interest. RP-growth conducts a branch-and-bound search using anti-monotonic upper bounds of the relevance scores such as F-score and χ^2, and the pruning in branch-and-bound search is successfully translated to minimum support raising, a standard, easy-to-implement pruning strategy for top-k mining. Furthermore, by introducing the notion of weakness and an additional, aggressive pruning strategy based on weakness, RP-growth efficiently find k patterns of wide variety and high relevance to the class of interest.

1　はじめに

頻出パターン発見手法は知識発見・データマイニングにおける代表的な手法として知られているが，その一方で，最小サポート (minimum support) と呼ばれる閾値を注意深く設定しないと有益でないパターンが大量に出力されるという問題が経験的に知られている．この問題への一つの対処法として上位 k (top-k) パターン発見手法が提案されている [9, 19]．上位 k パターン発見手法では，ユーザは手に入れたいパターンの数 k を指定すればよく，最小サポートはデータベースに依存して自動調整されるという利点がある．また，データベース中の各トランザクションにクラスラベルが与えられている場合に，識別 (discriminative) パターンと呼ばれる，より有益なパターンを得る試みも数多く為されており，それらの試みは subgroup discovery [20], 顕在 (emerging) パターン発見 [4, 5, 17], contrast set mining [1], supervised descriptive rule discovery [12], cluster grouping [22] 等の名で知られている．

本論文では，頻出アイテム集合発見アルゴリズム FP-growth [8] に基づき，関連 (relevant) パターンと呼ばれる識別パターン上位 k 個の発見を行うアルゴリズム RP-growth を提案する．この RP-growth では，興味あるクラス c に対し，関連度 (relevance score) $R_c(\boldsymbol{x})$ の高いパターン（アイテム集合）$\boldsymbol{x}$ 上位 k 個を発見する．ただし $R_c(\boldsymbol{x})$ がパターンの包含関係に関する反単調性を一般には満たさないという問題があるため，既存研究でも行われているように [15, 16, 19, 20, 22], RP-growth では分岐限定 (branch-and-bound) 法を利用する．すなわち，探索木で訪問中の各パターン $\boldsymbol{x}$ について，反単調性を満たすような $R_c(\boldsymbol{x})$ の上界 $\overline{R}_c(\boldsymbol{x})$ が計算され，必要とされる閾値をこの上界が下回ったら $\boldsymbol{x}$ 以下の部分木は枝刈りされる．提案手法の特徴の一つとして，分岐限定法の枝刈り操作が最小サポート上昇法 (minimum support raising) [9] に翻訳されることが挙げられる．最小サポート上昇法は上位 k 頻出パターン発見で標準的に用いられており，実装が容易である [18]．加えて，データベース縮約 [18] など，最小サポートの制約を利用する既存のパターン発見の高速化技術をそのまま利用できるという利点がある．また本論文では，凹性 (convexity) を満たさないようないくつかの関連度（F 値等）に対しても最小サポート上昇法が適用できることを示す．そして RP-growth では，似た関連パターンが多量に出力されるのを防ぐために二つのパターン間で定義される「より弱い (weaker)」という関係を導入し，更にこの関係に基づく効果的な枝刈りを行うことでより効率的な探索を実現する．

本論文では次のような構成をとる．まず 2 節で本研究の背景を説明する．3 節で RP-growth の詳細な説明を行い，4 節で小規模データでの実験結果を示す．そして 5 節で本論文をまとめる．

*連絡先：東京工業大学 大学院情報理工学研究科
〒 152-8552 東京都目黒区大岡山 2-12-1
E-mail: kameya@mi.cs.tiech.ac.jp

2 背景

2.1 準備

はじめに幾つかの記法を導入する．我々の手元にはサイズ N のデータベース $\mathcal{D} = \{t_1, t_2, \ldots, t_N\}$ があるものとする．ここで t_i はトランザクションと呼ばれるアイテムの集合である．$\mathcal{D}$ に出現する全てのアイテムの集合を $\mathcal{X}$ と書く．そして各トランザクション t_i はクラス集合 $\mathcal{C}$ の中の一つ c_i に属すると考える $(1 \leq i \leq N)$．$\mathcal{X}$ の部分集合をパターンと呼び，可能なパターンの集合を $\mathcal{P}$ と書く．本論文ではアイテム間に全順序 $\prec$（具体的には後述）を導入し，トランザクション／パターン中のアイテムは常に $\prec$ に従った順序で並べられるとする．また，トランザクション／パターン間にも $\prec$ に基づく辞書順序を考える．パターン $\boldsymbol{x}$ は文脈に応じてベクトル $(x_1, x_2, \ldots, x_n)$，集合 $\{x_1, x_2, \ldots, x_n\}$，連言 $(x_1 \wedge x_2 \wedge \ldots \wedge x_n)$ のいずれかに読み替えられる（各 x_j はアイテム，$x_1 \prec x_2 \prec \cdots \prec x_n$）．また，アイテムは自身を要素とするサイズ 1 のパターンと見なされる．

更に，本論文で扱う確率は全てデータベース N における出現回数から計算される．例えば，パターン $\boldsymbol{x}$ を満たすクラス c のトランザクションの数を $N(c, \boldsymbol{x}) = \#\{i \mid c_i = c, \boldsymbol{x} \subseteq t_i, 1 \leq i \leq N\}$（$\#S$ は集合 S の要素数）とおいたとき，同時確率 $p(c, \boldsymbol{x})$ は $N(c, \boldsymbol{x})/N$ から計算される．また，簡単のため否定記号 $\neg$ を使い，$p(c, \neg\boldsymbol{x}) = N(c, \neg\boldsymbol{x})/N = \#\{i \mid c_i = c, \boldsymbol{x} \not\subseteq t_i, 1 \leq i \leq N\}/N$, $p(\neg c, \boldsymbol{x}) = N(\neg c, \boldsymbol{x})/N = \#\{i \mid c_i \neq c, \boldsymbol{x} \subseteq t_i, 1 \leq i \leq N\}/N$ 等と表す．これらの同時確率を使って周辺確率や条件付き確率が $p(\boldsymbol{x}) = p(c, \boldsymbol{x}) + p(\neg c, \boldsymbol{x})$, $p(c) = p(c, \boldsymbol{x}) + p(c, \neg\boldsymbol{x})$, $p(c \mid \boldsymbol{x}) = p(c, \boldsymbol{x})/p(\boldsymbol{x})$ 等と計算される．本論文では $0 < p(c) < 1$ および $0 < p(\boldsymbol{x}) < 1$ が成り立つ状況のみを考える．

2.2 関連度

前述したように，我々は興味あるクラス c と関連のある上位 k 個のパターンを発見することを目的とする．そのために，関連度 (relevance score) と呼ばれる $\mathcal{P}$ から $\mathbb{R}$ への関数 R_c を最初に定める．R_c で測られる関連度はクラス相関規則 (class association rule) $\boldsymbol{x} \Rightarrow c$ における興味深さの指標と見なせる（$\boldsymbol{x}$ はパターン）．過去の研究で多くの関連度が提案されており [7, 12]，幾つかのグループに分けられる：

- 確信度 (confidence) $p(c \mid \boldsymbol{x})$ はクラス相関規則発見で用いられる [13]．一方 $p(\boldsymbol{x} \mid c)$ をクラス c に対する正のサポートと呼び，$p(\boldsymbol{x} \mid \neg c)$ を負のサポートと呼ぶ．単に「サポート」と書いた場合は正のサポートを指す．Growth rate $\mathrm{GR}_c(\boldsymbol{x}) = p(\boldsymbol{x} \mid c)/p(\boldsymbol{x} \mid \neg c)$ は顕在パターン発見 [4] で用いられ，$\boldsymbol{x}$ が c によく現れ，c 以外のクラスにはあまり現れないことを数値的に表現する．点相互情報量 (pointwise mutual information) $\mathrm{PMI}_c(\boldsymbol{x}) = \log p(c, \boldsymbol{x}) - \log\{p(c)p(\boldsymbol{x})\}$ はテキスト分析で採用される指標であり [3]，対数を取らない場合 $p(c, \boldsymbol{x})/(p(c)p(\boldsymbol{x}))$ を lift と呼ぶ [7]．Kralj Novak らの結果を用いると，これらの関連度はいずれも $\mathcal{P}$ 中のパターンに対して同じ順序付けを与える [12]．
- 精度 (precision) と再現率 (recall) は情報検索 [14] や分類器の評価 [10] で用いられる指標であり，我々の文脈では，$p(c \mid \boldsymbol{x})$ と $p(\boldsymbol{x} \mid c)$ をクラス c に対する $\boldsymbol{x}$ の精度と再現率と見なす [7]．また，この 2 つの調和平均 $\mathrm{F}_c(\boldsymbol{x}) = 2p(c \mid \boldsymbol{x})p(\boldsymbol{x} \mid c)/(p(c \mid \boldsymbol{x}) + p(\boldsymbol{x} \mid c))$ を F 値と呼ぶ．更に，クラス c とパターン $\boldsymbol{x}$ の重なりを表現する同様の指標としては Jaccard 係数 $\mathrm{J}_c(\boldsymbol{x}) = p(c, \boldsymbol{x})/(p(c) + p(\boldsymbol{x}) - p(c, \boldsymbol{x}))$ も考えられる．
- Leverage $\mathrm{L}_c(\boldsymbol{x}) = p(c, \boldsymbol{x}) - p(c)p(\boldsymbol{x})$ は相関規則の興味深さの指標として用いられ [19]，subgroup discovery [20] というタスクで用いる weighted relative accuracy と等価である．また，指標 $\mathrm{SD}_c(\boldsymbol{x}) = p(\boldsymbol{x} \mid c) - p(\boldsymbol{x} \mid \neg c)$ は support difference と呼ばれることが多く，contrast set mining [1] で用いられる[1]．上と同様にこれらの関連度は $\mathcal{P}$ 中のパターンに対して同じ順序付けを与える [12]．
- χ^2 値は 2 確率変数間の関連の強さを表す指標である．$\{c, \neg c\}$ と $\{\boldsymbol{x}, \neg\boldsymbol{x}\}$ を各々確率変数 C と X の実現値としたとき，C と X の間の χ^2 値を関連度 $\chi^2_c(\boldsymbol{x}) = \sum_{c' \in \{c, \neg c\}, \boldsymbol{x}' \in \{\boldsymbol{x}, \neg\boldsymbol{x}\}} \tau(c', \boldsymbol{x}')$ と見なす．ただし，

$$\tau(c', \boldsymbol{x}') = \frac{(p(c', \boldsymbol{x}')N - p(c')p(\boldsymbol{x}')N)^2}{p(c')p(\boldsymbol{x}')N}$$

である．$\chi^2_c(\boldsymbol{x})$ は $p(c, \boldsymbol{x})$ および $p(\boldsymbol{x})$ の組に対して凹 (convex) であることが示されている [15]．

これらの関連度が高いパターンを本論文では**関連パターン** (relevant pattern) と呼ぶ．

2.3 最小サポート上昇法

本節では，上位 k パターン発見における標準的かつ実装容易な手法として知られる最小サポート上昇法 [9] について簡単に説明する．まず図 1 のようなパターン探索木を考える．この探索木では各ノード $\boldsymbol{x}$ の親がそのノードより一つだけ短い接頭辞 $\boldsymbol{y}$ となっており，その差分のアイテムは順序 $\prec$ において $\boldsymbol{y}$ 中のアイテムより後ろにある．そして兄弟は $\prec$ に基づく辞書順で左から右へ並ぶ．このような探索木を本論文では接頭探索木と

[1] Support difference の元定義は $|p(\boldsymbol{x} \mid c) - p(\boldsymbol{x} \mid \neg c)|$ [1] だが，クラス c に対する上位 k パターン発見では $p(\boldsymbol{x} \mid c) > p(\boldsymbol{x} \mid \neg c)$ なるパターン $\boldsymbol{x}$ に注目するため，定義を若干修正したものを用いる．

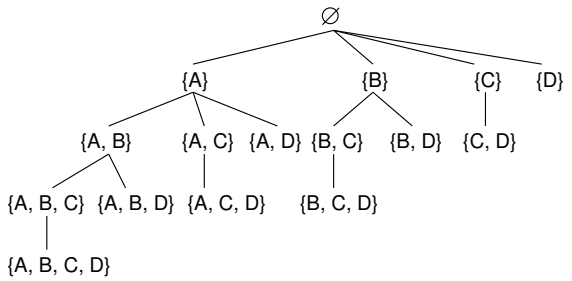

図 1: 接頭探索木の例.

呼ぶ．また，探索戦略は様々考えられるが，本論文では深さ優先探索（兄弟間は左優先）を用いることにする．ここで，興味あるクラス c に対して，$p(\boldsymbol{x} \mid c) \geq \sigma_{\min}$ なるパターン $\boldsymbol{x}$ を全て見つけることを考える（$\sigma_{\min}$ は最小サポート）．このとき，もし $p(\boldsymbol{x} \mid c) < \sigma_{\min}$ ならば，パターンの包含関係に関するサポートの反単調性 (anti-monotonicity) により $\boldsymbol{x}$ 以下の部分木を枝刈りできる．更に，上位 k 頻出パターン発見でもこの性質を利用する．すなわち最初に**候補リスト** (candidate list) を用意し，見つかったパターンをサポートの大きい順に格納する．ここで，探索の途中で候補リストのサイズが k 以上になったときを考え，この時点で k 番目に大きいサポートをもつパターンと $\boldsymbol{z}$ とおく．すると，その後に探索されるパターン $\boldsymbol{x}$ は $p(\boldsymbol{x} \mid c) \geq p(\boldsymbol{z} \mid c)$ を満たす必要があり，$p(\boldsymbol{x} \mid c) < p(\boldsymbol{z} \mid c)$ ならば $\boldsymbol{x}$ 以下の部分木を枝刈りできる．この操作は $\boldsymbol{z}$ を見つけた時点で最小サポートを $\sigma_{\min} := \max\{p(\boldsymbol{z} \mid c), \sigma_{\min}\}$ と上昇させてから最小サポートに従い枝刈りする操作と等価である．また，$(k+1)$ 番目以降のサポートをもつパターンは通常破棄される．最小サポート上昇法では，最小サポートを小さな値（典型的には $1/N$; N はトランザクション数）から始めて探索途中で随時上昇させることを繰り返す．上位 k 関連パターン発見における最小サポート上昇法の実現方法を次節で示す．

3 提案手法

クラスが付与されたトランザクションの集合 $\mathcal{D}$ を考える．このとき，我々の考える上位 k 関連パターン発見とは，興味あるクラス c に対して以下を満たすパターン集合 $\mathcal{P}^*$ を見つけることである．

1. $\forall \boldsymbol{x}' \in \mathcal{P} \setminus \mathcal{P}^*$ に対し，$R_c(\boldsymbol{z}) > R_c(\boldsymbol{x}')$ が成り立つ．ただし $\boldsymbol{z}$ は $\mathcal{P}^*$ において k 番目に大きい関連度をもつパターンである．
2. $\forall \boldsymbol{x} \in \mathcal{P}^*$ に対し $p(\boldsymbol{x} \mid c) \geq \sigma_{\min}$ が成り立つ．
3. $\forall \boldsymbol{x} \in \mathcal{P}^*$ に対し $p(c \mid \boldsymbol{x}) \geq \beta_{\min}$ が成り立つ．
4. $\mathcal{P}^*$ に属する任意の 2 つのパターン $\boldsymbol{x}$ と $\boldsymbol{x}'$ は互いに弱くない．すなわち $\boldsymbol{x} \subset \boldsymbol{x}'$ かつ $R_c(\boldsymbol{x}) \geq R_c(\boldsymbol{x}')$ は成り立たない．

ただし，k (≥ 1), $\sigma_{\min}$ (> 0), $\beta_{\min}$ (> 0) はユーザが指定するパラメータである．典型的には条件 2 において $\sigma_{\min} = 1/|\mathcal{D}|$ を指定し，条件 3 において $\beta_{\min} = p(c)$ もしくは 0.5 を指定する．$\beta_{\min} = 0.5$ を指定する場合は弱分類器となるパターンの発見を目指すことを意味し，識別能力の高いパターンを得る場合はより大きな $\beta_{\min}$ を指定する．また，条件 4 については 3.2 節で詳しく述べる．本節の以降では，上位 k 関連パターン発見アルゴリズムである RP-growth の背景となる概念を順に述べ，3.3 節において RP-growth を記述する．

3.1 分岐限定法から最小サポート上昇へ

先に述べたように，関連度が反単調性を満たさない場合，分岐限定法を導入することが多い [15, 16, 20, 22]．興味あるクラスを c，現在考慮しているパターンを $\boldsymbol{x}$ としたとき，上位 k パターン発見における分岐限定法では反単調性を満たすような $R_c(\boldsymbol{x})$ の上界 $\overline{R}_c(\boldsymbol{x})$ を導入し，$\overline{R}_c(\boldsymbol{x}) < R_c(\boldsymbol{z})$ であれば $\boldsymbol{x}$ 以下の部分木を枝刈りする．ここで $\boldsymbol{z}$ は候補リストにおいて k 番目に大きい関連度をもつパターンである．$\boldsymbol{x}$ を含む任意のパターン $\boldsymbol{x}'$ に対し，$R_c(\boldsymbol{x}') \leq \overline{R}_c(\boldsymbol{x}') \leq \overline{R}_c(\boldsymbol{x}) < R_c(\boldsymbol{z})$ が成り立つため，この枝刈りは安全である．

また既存研究においては，共通の楽観的状況を考えることで関連度の上界を得ている．例えば AprioriSMP [15] においては，$p(\boldsymbol{x}) = p(c, \boldsymbol{x})$ （$p(\neg c, \boldsymbol{x}) = 0$ とも言い換えられる）が成り立つ場合を考えて $\chi^2_c(\boldsymbol{x})$ の上界を得ている．そこで，この条件 $p(\boldsymbol{x}) = p(c, \boldsymbol{x})$ が $p(c \mid \boldsymbol{x}) = 1$, $p(\boldsymbol{x} \mid \neg c) = 0$, $p(\neg \boldsymbol{x} \mid \neg c) = 1$, $p(\boldsymbol{x}) = p(c)p(\boldsymbol{x} \mid c)$, $p(\neg c, \neg \boldsymbol{x}) = p(\neg c)$ 等の条件と同値であるという性質を利用して，まず関連度の定義式 $R_c(\boldsymbol{x})$ に対して同時に $p(c \mid \boldsymbol{x}) := 1$, $p(\boldsymbol{x} \mid \neg c) := 0$, $p(\neg \boldsymbol{x} \mid \neg c) := 1$, $p(\boldsymbol{x}) := p(c)p(\boldsymbol{x} \mid c)$ 等を代入して $\overline{R}_c(\boldsymbol{x})$ を得る．そして，(i) $\overline{R}_c(\boldsymbol{x})$ が $R_c(\boldsymbol{x})$ の上界であり $(\overline{R}_c(\boldsymbol{x}) \geq R_c(\boldsymbol{x}))$, かつ (ii) $\overline{R}_c(\boldsymbol{x})$ が反単調性を満たすと確認できればよい．この手続きは F 値のように凹性を満たさない関連度に対しても適用可能である．ただし growth rate $\mathrm{GR}_c(\boldsymbol{x})$ や確信度 $p(c \mid \boldsymbol{x})$ のように上界が常に大きな値となる場合（例えば $p(\boldsymbol{x} \mid \neg c) = 0$ のとき $\mathrm{GR}_c(\boldsymbol{x}) = +\infty$）には枝刈りの機会が得られない．

例えば $\mathrm{F}_c(\boldsymbol{x}) = 2p(c \mid \boldsymbol{x})p(\boldsymbol{x} \mid c)/(p(c \mid \boldsymbol{x}) + p(\boldsymbol{x} \mid c))$ と定義される F 値に対し，$p(c \mid \boldsymbol{x}) := 1$ を代入して上界

$$\overline{\mathrm{F}}_c(\boldsymbol{x}) \quad = \quad \frac{2p(\boldsymbol{x} \mid c)}{1 + p(\boldsymbol{x} \mid c)}.$$

を得る．同様に χ^2, support difference の上界も各々次のように得られる（導出過程は省略）:

$$\begin{aligned}\overline{\chi^2_c}(\boldsymbol{x}) &= N\frac{p(\boldsymbol{x}\mid c)p(\neg c)}{1-p(c)p(\boldsymbol{x}\mid c)}\\ \overline{\mathrm{SD}}_c(\boldsymbol{x}) &= p(\boldsymbol{x}\mid c).\end{aligned}$$

上が確かに条件 (i) を満たすことは容易に確認できる（χ^2 もその凹性から条件 (i) を満たす）．また，いずれも反単調性をもつ $p(\boldsymbol{x}\mid c)$ の単調関数であるため $(0\le p(\boldsymbol{x}\mid c)\le 1)$，条件 (ii) の反単調性を満たす．

更に，これまで述べた分岐限定法に基づく枝刈りは多くの場合最小サポート上昇法に翻訳可能である．例えば候補リストで k 番目に大きい関連度をもつパターンを $\boldsymbol{z}$ としたとき，$\overline{\mathrm{F}}_c(\boldsymbol{x}) < \mathrm{F}_c(\boldsymbol{z})$ ならば $\boldsymbol{x}$ が最終的に上位 k パターンの中に残ることはない．このことは上の $\overline{\mathrm{F}}_c(\boldsymbol{x}) = 2p(\boldsymbol{x}\mid c)/(1+p(\boldsymbol{x}\mid c))$ を使うと，

$$p(\boldsymbol{x}\mid c) < \frac{\mathrm{F}_c(\boldsymbol{z})}{2-\mathrm{F}_c(\boldsymbol{z})} \tag{1}$$

と書き直すことができる．これは最小サポートを

$$\sigma_{\min} := \max\left\{\frac{\mathrm{F}_c(\boldsymbol{z})}{2-\mathrm{F}_c(\boldsymbol{z})}, \sigma_{\min}\right\}.$$

と上昇させて，その最小サポートに基づき枝刈りを行う操作と等価である．他の関連度についても

$$\begin{aligned}\sigma_{\min} &:= \max\left\{\frac{\chi^2_c(\boldsymbol{z})}{p(c)\chi^2_c(\boldsymbol{z})+p(\neg c)N}, \sigma_{\min}\right\}\\ \sigma_{\min} &:= \max\left\{\mathrm{SD}_c(\boldsymbol{z}), \sigma_{\min}\right\}\end{aligned}$$

と更新式が得られる．これらの式の一般形として以降では $\sigma_{\min} := \max\{U_c(\boldsymbol{z}), \sigma_{\min}\}$ を用いる．最小サポート上昇法を用いることで，データベース縮約 [8, 18] のように最小サポートを利用した頻出パターン発見手法の高速化技法をそのまま利用できる．また，最小サポート上昇法を実装した頻出パターン発見プログラムに対する修正も僅かで済む．ただし，関連度としてよく用いられる情報利得 (information gain) [15, 16, 22] に対しては式 1 のように明示的な条件式を得るのが難しい．

3.2 弱い関連パターン

上位 k パターン発見においても冗長なパターンが出力される場合が多く見られる．例えばパターン $\{\mathsf{A}\}$ がクラス c と非常に関連が強い場合，$\{\mathsf{A},\mathsf{B}\}, \{\mathsf{A},\mathsf{C}\}, \{\mathsf{A},\mathsf{B},\mathsf{C}\}$ のように A を含むパターンもまた関連が強いと判断され，結果としてこれらのパターンが最終的に得られる上位 k パターンを独占する．このように冗長なパターンを排除するためにパターンの包含関係に関する**極小性** (minimality) の制約を導入することが多い [5, 6, 17]．すなわち，必要な制約を満たした二つの候補 $\boldsymbol{x}$, $\boldsymbol{x}'$ に対し，$\boldsymbol{x}\subset\boldsymbol{x}'$ であれば $\boldsymbol{x}'$ を冗長として取り除く．しかし，深さ優先探索に基づく上位 k パターン発見では極小性の取扱いに関して二つの問題がある．一つ目の問題は候補リストの動的な振る舞いである．例えば図 1 中の二つのパターン $\boldsymbol{x}=\{\mathsf{A},\mathsf{C}\}, \boldsymbol{x}'=\{\mathsf{A},\mathsf{C},\mathsf{D}\}$ を考える．そして $\boldsymbol{x}$ に既に訪問済みで候補リストに入っているとする．このとき，$R_c(\boldsymbol{x})\ge R_c(\boldsymbol{x}')$ であれば $\boldsymbol{x}'$ は直ちに削除してよいが[2]，そうでない場合，$\boldsymbol{x}'$ が削除可能かどうかは $\boldsymbol{x}$ が最終の上位 k パターンに残るかどうかに依存するため，直ちに $\boldsymbol{x}'$ を削除できない．また，二つ目の問題は深さ優先探索における走査順序である．例えば図 1 において，後に訪問する $\boldsymbol{x}''=\{\mathsf{C}\}$ の関連度が $\boldsymbol{x}=\{\mathsf{A},\mathsf{C}\}$ より高ければ，$\boldsymbol{x}$ もまた削除の対象となる．二つの問題いずれにおいても，削除可能性が未定のパターンを暫定的に候補リストに残してしまうと候補リストの維持作業自体が複雑になり，更に最小サポート上昇法も効果的に働かなくなる．

まず一つ目の問題に対しては，極小性の代わりに二つのパターン間で相対的に定義される「より弱い (weaker)」という関係を導入する．この概念はベイジアンネットの説明的分析を行う most relevant explanation [21] という枠組みの中で提案され，以下のように定義される[3]．

定義 1 興味あるクラス c, $\mathcal{P}$ 中のパターン $\boldsymbol{x}$, $\boldsymbol{x}'$ に対し，$\boldsymbol{x}\subset\boldsymbol{x}'$ かつ $R_c(\boldsymbol{x})\ge R_c(\boldsymbol{x}')$ が成り立つ時，またその時に限り「$\boldsymbol{x}'$ は $\boldsymbol{x}$ より弱い」と言う．

例えば $R_c(\{\mathsf{A},\mathsf{C},\mathsf{D}\}) > R_c(\{\mathsf{A},\mathsf{C}\})$ が成り立つ時，$\{\mathsf{A},\mathsf{C}\}$ に D を付け加えるのはクラス c の特徴を捉えるのに有効であると言える．一方，$R_c(\{\mathsf{A},\mathsf{C},\mathsf{D}\}) \le R_c(\{\mathsf{A},\mathsf{C}\})$ であるとき，$\{\mathsf{A},\mathsf{C}\}$ に D を付け加えるのは冗長（あるいは有害）と考えられる．更に先の例において，候補リストに入っているパターン $\boldsymbol{x}=\{\mathsf{A},\mathsf{C}\}$ と現在訪問中のパターン $\boldsymbol{x}'=\{\mathsf{A},\mathsf{C},\mathsf{D}\}$ を考える．$R_c(\boldsymbol{x})\ge R_c(\boldsymbol{x}')$ の場合は上と同様 $\boldsymbol{x}'$ を直ちに削除できる．一方 $R_c(\boldsymbol{x}) < R_c(\boldsymbol{x}')$ の場合，$\boldsymbol{x}'$ は $\boldsymbol{x}$ より弱くないため，$\boldsymbol{x}$ が最終的に上位 k パターンに残るか否かに関わらず，$\boldsymbol{x}'$ をそのまま候補リストに追加してよい．

また，二つ目の問題に対しては図 2 のようなパターン探索木を考えればよい．この探索木では各ノード $\boldsymbol{x}$ の親がそのノードより一つだけ短い接尾辞 $\boldsymbol{y}$ であり，その差分のアイテムは順序 $\prec$ において $\boldsymbol{y}$ 中のアイテムより前にある．そして兄弟は順序 $\prec$ に基づく辞書順で左から右へ並ぶ．このような探索木を接尾探索木と呼ぶ．図 2 より，接尾探索木では「$\boldsymbol{x}$ を訪問する時点で

[2] $\boldsymbol{x}'$ が最終の上位 k パターンに残るときには $\boldsymbol{x}$ も同時に残るため，$\boldsymbol{x}'$ は極小性に違反する．

[3] 関係「より弱い」は元々[21] で strong dominance と呼ばれる関係を逆にしたものである．ただし，most relevant explanation では 3.1 節で述べた分岐限定法に基づく探索や，後述する「より甚弱である」という関係に基づく枝刈りの仕組みは提供していない．

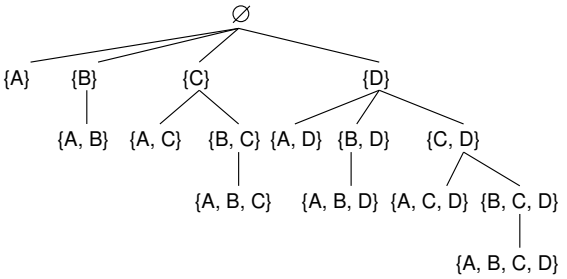

図 2: 接尾探索木の例.

表 1: トランザクションの例（左）とその F 値（右）.

Class c	Transaction
+	{A, B, C, D}
+	{A, B, D, F}
+	{B, C, E}
−	{A, C}
−	{B, C, D}
−	{A, C, E, F}

Item x	$N(+,x)$	$N(-,x)$	$F_+(x)$
B	3	1	0.857
D	2	1	0.667
A	2	2	0.571
C	2	3	0.500
E	1	1	0.400
F	1	1	0.400

$\boldsymbol{x}$ の真の部分パターンは全て訪問済みである」ことが分かる．更に，各トランザクション中のアイテムが順序 $\prec$ に従って並ぶ時，FP-growth の探索は（暗黙のうちに）接尾探索木上で行われている（詳細は後述）．以上の議論から,「より弱い」という相対的関係と接尾探索木の導入により，我々は新たに得られたパターン $\boldsymbol{x}$ に対して，もし $\boldsymbol{x}$ がその時点の候補リスト中のパターンどれに対しても弱くなければ $\boldsymbol{x}$ をそのまま候補リストに追加するだけでよい．

更に,「より弱い」の特殊な関係である「より甚弱である (prunably weaker)」を利用した枝刈りを導入する．

定義 2 興味あるクラス c, $\mathcal{P}$ 中のパターン $\boldsymbol{x}$, $\boldsymbol{x}'$ に対し，$\boldsymbol{x} \subset \boldsymbol{x}'$ かつ $R_c(\boldsymbol{x}) \geq \overline{R}_c(\boldsymbol{x}')$ が成り立つ時，またその時に限り「$\boldsymbol{x}'$ は $\boldsymbol{x}$ より甚弱である」と言う．

候補リスト内のあるパターン $\boldsymbol{x}$ と訪問中の $\boldsymbol{x}'$ を考える．$\boldsymbol{x}'$ が $\boldsymbol{x}$ より甚弱であるとき，$\boldsymbol{x}'$ を含むパターンは全て $\boldsymbol{x}$ より弱いと保証される．従って $\boldsymbol{x}'$ 以下の部分木は枝刈り可能である．言い換えると，パターン $\boldsymbol{y}$ を候補リストに追加する際に，$\boldsymbol{y}$ が候補リスト内の全パターンに比べ甚弱でない限りは，$\boldsymbol{y}$ 以下の部分木が枝刈りされてしまうため，この枝刈りは効果的である．

そして接尾探索木の根からパターン $\boldsymbol{x}$ に至るパス上は全て $\boldsymbol{x}$ の部分パターンであるため，甚弱性の検査におけるパターン間の包含検査を一部省略できる．具体的には，接尾探索木における根から葉への各パス π において，**局所最小サポート** $\sigma^{\pi}_{\min}$ を導入し，パターン $\boldsymbol{x}$ を訪問した直後に，3.1 節で述べたように $\sigma^{\pi}_{\min} := \max\{U_c(\boldsymbol{x}), \sigma^{\pi}_{\min}\}$ と更新する．この局所最小サポートによる枝刈りは探索の初期段階において特に効果的で

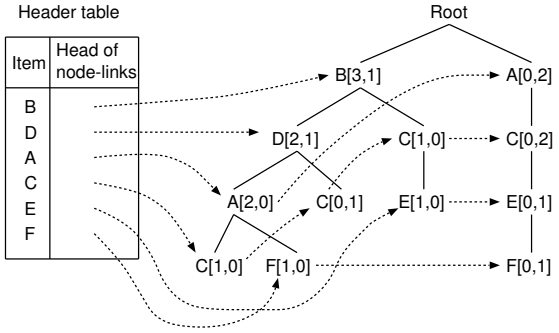

図 3: 初期の RP-tree とヘッダ表.

あると考えられる．また，$\sigma^{\pi}_{\min}$ は深さ優先探索を実現する再帰ルーチンの引数として容易に実装できる．

3.3 RP-growth アルゴリズム

本節ではこれまで述べてきた概念に基づき RP-growth を記述する．まず，RP-growth が用いるデータ構造である *RP-tree* を説明する．まず，興味あるクラスをラベル + で参照し，残りのクラスに属するトランザクションにはラベル − を与える（2 クラス化する）．RP-tree は FP-growth [8] で用いる FP-tree の簡単な拡張であり，+/− ラベルが付与されたトランザクション集合をコンパクトに格納するデータ構造である．そのようなトランザクション集合の例（[17] による）を表 1 左に示す．+ クラス，− クラスに属するトランザクションの数を各々 N^+, N^- とおく（表 1 左では $N^+ = N^- = 3$）．

我々はまずトランザクション集合をスキャンし，各アイテム x についてクラス +, − における出現度数 $N(+,x)$, $N(-,x)$ と関連度 $R_c(\{x\})$ を計算する．表 1 左の場合における各アイテムの出現度数と F 値を表 1 右に示す．例えばアイテム B の F 値 $F_+(\mathsf{B})$ は $p(+ \mid \mathsf{B}) = N(+,\mathsf{B})/(N(+,\mathsf{B}) + N(-,\mathsf{B})) = 3/4$ と $p(\mathsf{B} \mid +) = N(+,\mathsf{B})/N^+ = 1$ の調和平均として得られる．その後，アイテム間の順序 $\prec$ を関連度（F 値）の降順（関連度の値が同じ場合はアルファベット順）と定め，各トランザクション内のアイテムを $\prec$ に従い並べ直す．表 1 の場合は $\mathsf{B} \prec \mathsf{D} \prec \mathsf{A} \prec \mathsf{C} \prec \mathsf{E} \prec \mathsf{F}$ となる．

このような準備の後に初期 RP-tree が構築される．クラス +, − の出現度数を分離すること以外，初期 RP-tree の構築は通常のトライ (trie) 構築の場合と同じである．図 3 は表 1 に対する初期 RP-tree である．この初期 RP-tree において，根からノード $\mathsf{A}[2,0]$ に至るパスはパターン $\{\mathsf{B},\mathsf{D},\mathsf{A}\}$ がクラス + において 2 回，クラス − において 0 回出現することを表す．また初期 RP-tree と同時にヘッダ表 (header table) も構築される．図 3 に示すように，ヘッダ表のエントリ（アイテム）x は $x[n^+, n^-]$ の形のラベルで参照されるノードを

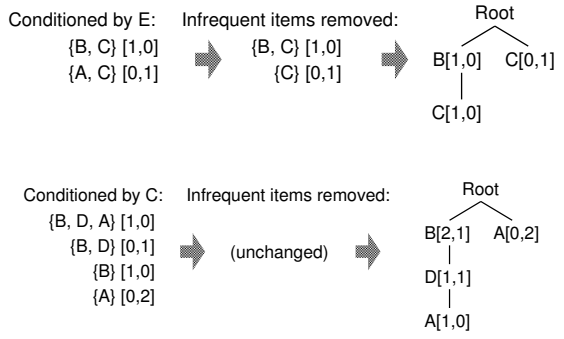

図 4: E（上）と C（下）で条件付けられた RP-tree.

Algorithm 1 RP-Growth-Main(c_0, k_0, $\mathcal{D}$)

Require: c_0: the class of interest, k_0: the number of non-weak relevant patterns, $\mathcal{D}$: the dataset
1: $c := c_0$ and $k := k_0$
2: $T :=$ the initial RP-tree constructed from $\mathcal{D}$
3: $L :=$ an empty candidate list
4: $\boldsymbol{x} :=$ an empty pattern
5: $\sigma_{\min} := 1/|\mathcal{D}|$
6: Call RP-Growth(T, $\boldsymbol{x}$, $1/|\mathcal{D}|$)
7: Output the patterns in L

結ぶリンクのヘッダと対応付けられる．以降 n^+, n^- を各々**正の度数**，**負の度数**と呼ぶ．

また RP-growth では探索の途中で**条件付き** *RP-tree* と呼ばれる RP-tree が構築される．探索木上で初期パターン $\emptyset$ からパターン {E} に訪問する場合，図 3 の初期 RP-tree から E で条件付けられた RP-tree が図 4 上のように構築される．初期の最小サポートは $1/N^+ = 1/3$ としている．その手続きは以下の通りである．まず，初期 RP-tree において E と対応づけられるヘッダから辿れる二つのノード $\mathsf{E}[1,0]$, $\mathsf{E}[0,1]$ を見つける．次に根とノード $\mathsf{E}[1,0]$ の間にあるノード B, C から**度数付きパターン** $\{\mathsf{B},\mathsf{C}\}[1,0]$ を得る．ここで $[1,0]$ はノード $\mathsf{E}[1,0]$ から引き継がれる．同様にノード $\mathsf{E}[0,1]$ から度数付きパターン $\{\mathsf{B},\mathsf{C}\}[0,1]$ が得られる．ここでパターン {E} の正/負の度数 $[N(+,\{\mathsf{E}\}), N(-,\{\mathsf{E}\})]$ は $[1,0]+[0,1]=[1,1]$ と計算され，{E} の F 値は $N(+,\{\mathsf{E}\})/(N(+,\{\mathsf{E}\}) + N(-,\{\mathsf{E}\})) = 1/(1+1) = 1/2$, $N(+,\{\mathsf{E}\})/N^+ = 1/3$ の調和平均として計算される．次に度数付きパターン中のアイテムのうち，頻出でないものは削除する．例えば図 4 上で A, B, C の度数が各々 $[0,1]$, $[1,0]$, $[1,1]$ と得られるが，現在の最小サポート $1/N^+ = 1/3$ に対し A の正の度数が 0 であるため，A は削除される．最後に，再び通常のトライ構築手続に従い，図 4 上の右側の図のように条件付き RP-tree が構築される．図 4 上で A が削除されたように，探索木の深い所に行くにつれ条件付き RP-tree は縮約され，探索の高速化につながる [18]．また図 4 上から分かるように E で条件付けられた RP-tree には順序 $\prec$ において E の前に位置する B, C しか出現しないため，パターン {E} の子ノードとして探索されるのはパターン $\{\mathsf{B},\mathsf{E}\}$, $\{\mathsf{C},\mathsf{E}\}$ である．このように RP-growth (FP-growth) の探索木は接尾探索木となる．一方，C で条件づけられた RP-tree が図 4 下のように得られる．

以上の概念に基づき，上位 k 関連パターン発見を行うアルゴリズム RP-growth を Algorithm 1, 2 のように提案する．RP-Growth-Main は RP-growth の主手続きであり，RP-Growth は RP-Growth-Main から呼び出される再帰手続きである．ここでは興味あるクラス c, 出力するパターン数 k, 最小サポート $\sigma_{\min}$, 最小確信度 $\beta_{\min}$, 候補リスト L を大域変数とする．まず RP-Growth-Main では，各大域変数を初期化し（初期 RP-tree の構築含む），RP-Growth を呼び出す．そして RP-Growth の実行終了後，候補リストに格納される上位 k パターンを出力する．各呼び出し RP-Growth$(T_0, \boldsymbol{x}_0, \sigma_0)$ では RP-tree T_0 に出現する各アイテムに対しループを回す．アイテム x に対するループでは，まず x で条件付けられた条件付き RP-tree を構築する（行 3）．ここには最小サポート（$\sigma_{\min}$ と局所最小サポート σ_0）に基づく枝刈りが含まれる[4]．その後現在のパターン $\boldsymbol{x}_0$ の前に x を足して $\boldsymbol{x}$ とし（行 4），関連度 $R_c(\boldsymbol{x})$ を計算する（行 5）．パターン $\boldsymbol{x}$ が確信度に関する制約を満たしたら（行 6），今計算した $R_c(\boldsymbol{x})$ に基づき，局所最小サポートを上昇させる（行 7）．更に，$\boldsymbol{x}$ が候補リスト L 中のどのパターンに比べても弱くない場合（行 8），$\boldsymbol{x}$ を L に追加する．また L が一杯なら余分なパターンを削除し（行 11），最小サポートを上昇させる（行 12）．$\boldsymbol{x}$ が L 中のパターンより弱いことがあっても RP-Growth への再帰呼び出し（行 17）により，探索木の更に深くまで進む可能性があるが，$\boldsymbol{x}$ が L 中のいずれかのパターンに比べ甚弱である場合はそれ以上の探索は行われない．

4 実験

提案手法の効率性・有用性を示すために，20 newsgroup データを用いた実験を行う．このデータは 20 のニュースグループの約 20,000 記事から成り，各記事が属するニュースグループをクラスと見なす．我々は `http://people.csail.mit.edu/jrennie/20Newsgroups/` で公開されている前処理済みのデータを使い，更に本実験用に前処理を行った[5]．最終的に得られた 20 newgroup データは 5,666 単語から成る 17,930 記事である．各記

[4] 更に効率化を図る場合は，行 5–7 の処理を行 3 の条件付き RP-tree の構築処理に埋め込み，$\sigma_{\min}$ と σ（更新された局所最小サポート）に基づく枝刈りを行う．

[5] 具体的には SMART システムのリストに基づく stop word の削除，Porter の stemming アルゴリズム [14] の適用，頻出しない単語（50 回未満）の削除，短い記事（10 単語未満）の削除を行った．

Algorithm 2 RP-GROWTH(T_0, $\boldsymbol{x}_0$, σ_0)

Require: T_0: the current RP-tree, $\boldsymbol{x}_0$: the current pattern, σ_0: the current local minimum support
1: H_0 := the header table associated with T_0
2: **for all** x in the key items of H_0 **do**
3: Construct an RP-tree T conditioned on x from T_0 (pruning based on $\sigma_{\min}$ and σ_0 is embedded)
4: $\boldsymbol{x} := \{x\} \cup \boldsymbol{x}_0$
5: Compute $R_c(\boldsymbol{x})$ and $p(c \mid \boldsymbol{x})$ from the positive/negative counts stored in T
6: **if** $p(c \mid \boldsymbol{x}) \geq \beta_{\min}$ **then**
7: $\sigma := \max\{U_c(\boldsymbol{x}), \sigma_0\}$
8: **if** $\boldsymbol{x}$ is *not* weaker than any patterns in L **then**
9: Insert $\boldsymbol{x}$ into L following the descending order of R_c
10: **if** $|L| \geq k$ **then**
11: Remove the patterns with the $(k+1)$-th greatest or smaller relevance scores (if any) from L
12: $\sigma_{\min} := \max\{U_c(\boldsymbol{z}), \sigma_{\min}\}$, where $\boldsymbol{z}$ is the k-th pattern in L
13: **end if**
14: **end if**
15: **end if**
16: **if** $\boldsymbol{x}$ is *not* prunably weaker than any patterns in L **then**
17: Call RP-GROWTH(T, $\boldsymbol{x}$, σ)
18: **end if**
19: **end for**

事はそこに出現する単語から成る集合に変換され，一つのトランザクションと見なされる．

初めに RP-growth で得られた関連パターンが直観的であることを確認する．まず，各クラス（ニュースグループ）c に対して，c をクラス $+$，c 以外の 19 クラスを $-$ と置き直す（2 クラス化する）．クラス `comp.graphics`, `rec.sport.hockey`, `talk.politics.guns` における関連パターン上位 25 個を表 2 に示す．関連度は F 値を用い，$p(c \mid \boldsymbol{x})$ に対する閾値 $\beta_{\min}$ は 0.5 に設定した．この表から，第 1 位の関連パターンはニュースグループ名そのものに含まれ（接尾辞は stemming により置換・削除されている），他のパターンも各クラスによく関連している．ただしニュースグループ `comp.graphics` において単語 graphic が単独では関連パターンになれず，他の単語と結びついて上位パターンを構成する．上位 25 個のリストでは識別力（確信度・精度）の高い単語（`comp.graphics` における {polygon} 等）と支持度（再現率）の高い単語（`talk.politics.guns` における {gun} 等）がうまく混合されている．また，試しに「弱い」関連パターンも上位 25 個のリストに含めることを許すと，`talk.politics.guns` では 16 個のパターンに単語 gun が含まれる．多様なパターンを得る上で「弱い」という関係が重要な役割を果たしている．しかし一方で，2 クラス化したときにクラス分布 $\{p(+), p(-)\}$ が偏るため，$\beta_{\min} = 0.5$ という設定では比較的支持度が小さいパターンが得られる．そのため，例えばニュースグループ `alt.atheism` では {keith, caltech} 等，固有名詞を含むパターンが多くなる．この場合，$\beta_{\min}$ を $p(+)$ と 0.5 の間に設定した方がよいと思われる．

また，関連度によって探索時間が大きく異なることが分かった．χ^2 値を関連度としたとき，ニュースグループ `comp.graphics`, `comp.windows.x` において求める識別パターン数 k が大きいとき，2 時間以内に探索が終了しなかった（実装言語: Java, CPU: Core i7 2.66GHz）．それに対し，F 値では `comp.os.ms-windows.misc` を除く全てのニュースグループで 1 分以内で探索が終了した（`comp.os.ms-windows.misc` では $k = 500$ の時に約 17 分）．そして support difference でも同様に `comp.os.ms-windows.misc` を除く全てのニュースグループで 1 分以内で探索が終了した（`comp.os.ms-windows.misc` では $k = 500$ の時に約 5 分）．

更に，関連パターンを利用して識別力の高いテキスト分類器を構築することを考える[6]．まず追加の実験設定について述べる．各トランザクション t は各単語 w を属性に対応づけた 2 値ベクトルに変換される．すなわち $w \in t$ であれば，対応する属性の値を 1, そうでなければ 0 とする．更に RP-growth で得られた関連パターンに基づき**合成属性** (combined feature) を構築する．すなわち，合成属性の各々は関連パターンの一つ $\boldsymbol{x}$ に対応し，$\boldsymbol{x} \subseteq t$ であれば値 1, そうでなければ値 0 をとる．また，サポートベクターマシン (SVM) を共通の分類器とする．分類結果を評価するために，10 分割の層別 (stratified) 交差検定[7]を行う．一回の評価においては 8 分割分のデータを訓練データ，1 分割分をテストデータ，残りを SVM と RP-growth の制御パラメータを調整するための held-out データとする[8]．SVM の実

[6] パターン発見手法で得られたパターンを用いる研究としては，頻出パターンを使用するもの [2, 13], 識別パターンを使用するもの [5, 6, 17] が知られている．

[7] 層別交差検定では各分割におけるクラス分布が可能な限り保存される [10].

[8] SVM の制御パラメータは線形カーネルの場合 C, RBF カーネルの場合 C と γ である．また，RP-growth の制御パラメータは関連パターン数 k と閾値 $\beta_{\min}$ である．これらの制御パラメータは held-out データに対するグリッド探索により調整される．線形カーネルに対する C は $\{2^i \mid i = -7, -6, \ldots, 4, 5\}$ から選ばれ，RBF カーネルに対する C は $\{2^i \mid i = -7, -6, \ldots, 8, 9\}$ から，γ は $\{2^i \mid i = -13, -12, -11, -10, -9, -8, -7, -6, -5, -3,$

表 2: 20 newsgroup データにおける上位 25 の非弱な関連パターン（関連度を F 値，$\beta_{\min} = 0.5$ と設定）.

$c =$ comp.graphics

Pattern $\boldsymbol{x}$	$p(c \mid \boldsymbol{x})$	$p(\boldsymbol{x} \mid c)$	$\mathrm{F}_c(\boldsymbol{x})$
{graphic, program}	0.537	0.136	0.217
{gif}	0.552	0.119	0.196
{graphic, imag}	0.642	0.108	0.185
{imag, program}	0.516	0.110	0.181
{imag, file}	0.531	0.105	0.175
{graphic, find}	0.578	0.087	0.151
{imag, bit}	0.514	0.083	0.144
{graphic, code}	0.613	0.081	0.143
{graphic, bit}	0.545	0.080	0.140
{graphic, packag}	0.591	0.076	0.134
{format, convert}	0.588	0.075	0.132
{graphic, comp}	0.730	0.072	0.132
{imag, format}	0.613	0.072	0.129
{graphic, point}	0.573	0.070	0.125
{graphic, format}	0.670	0.068	0.123
{imag, convert}	0.596	0.066	0.118
{polygon}	0.915	0.060	0.113
{imag, softwar}	0.500	0.062	0.111
{graphic, ftp}	0.500	0.061	0.109
{graphic, algorithm}	0.852	0.058	0.108
{jpeg}	0.825	0.058	0.108
{graphic, group}	0.514	0.060	0.108
{graphic, site}	0.530	0.059	0.106
{graphic, comput, articl}	0.525	0.059	0.106
{code, algorithm}	0.500	0.059	0.105

$c =$ rec.sport.hockey

Pattern $\boldsymbol{x}$	$p(c \mid \boldsymbol{x})$	$p(\boldsymbol{x} \mid c)$	$\mathrm{F}_c(\boldsymbol{x})$
{hockei}	0.943	0.377	0.538
{team}	0.519	0.473	0.495
{playoff}	0.943	0.277	0.428
{game, plai}	0.506	0.273	0.354
{nhl}	0.990	0.206	0.341
{cup}	0.584	0.195	0.292
{player, plai}	0.575	0.190	0.286
{score}	0.510	0.194	0.281
{game, player}	0.561	0.186	0.280
{game, goal}	0.899	0.157	0.267
{game, win}	0.517	0.174	0.260
{game, fan}	0.622	0.164	0.260
{plai, goal}	0.852	0.144	0.246
{wing}	0.515	0.156	0.240
{leaf}	0.894	0.132	0.230
{bruin}	1.000	0.130	0.230
{pittsburgh}	0.567	0.142	0.226
{game, watch}	0.621	0.136	0.224
{detroit}	0.733	0.131	0.222
{penguin}	0.871	0.127	0.222
{game, season}	0.539	0.137	0.219
{game, night}	0.660	0.129	0.216
{ranger}	0.629	0.129	0.214
{plai, win}	0.529	0.134	0.214
{plai, fan}	0.603	0.128	0.211

$c =$ talk.politics.guns

Pattern $\boldsymbol{x}$	$p(c \mid \boldsymbol{x})$	$p(\boldsymbol{x} \mid c)$	$\mathrm{F}_c(\boldsymbol{x})$
{gun}	0.540	0.414	0.469
{weapon}	0.528	0.253	0.342
{fbi}	0.506	0.246	0.331
{firearm}	0.884	0.196	0.321
{batf}	0.662	0.155	0.252
{waco}	0.543	0.154	0.240
{assault}	0.587	0.124	0.205
{cdt, sw}	0.933	0.110	0.196
{cdt, stratu}	0.916	0.110	0.196
{handgun}	0.818	0.111	0.195
{cdt}	0.817	0.110	0.193
{stratu, sw}	0.700	0.110	0.190
{fire, compound}	0.698	0.109	0.188
{stratu}	0.570	0.110	0.184
{bd}	0.530	0.110	0.182
{sw}	0.521	0.110	0.181
{atf}	0.692	0.101	0.176
{arm, law}	0.527	0.086	0.148
{compound, dai}	0.598	0.082	0.144
{nra}	0.696	0.079	0.143
{rocket, special}	0.750	0.077	0.140
{rocket, speak}	0.840	0.076	0.139
{rocket, vo}	0.918	0.075	0.139
{vo, investor}	0.918	0.075	0.139
{vo, speak, todai}	0.918	0.075	0.139

装として LIBSVM[9]を用い，線形カーネルと RBF カーネルの 2 種類を用いる．ここでは次の 3 つの設定における SVM を分類能力を比較する．

1. 単体 (singleton) 属性を用いる線形カーネル
2. 単体属性を用いる RBF カーネル
3. 単体属性と合成属性の両方を用いる線形カーネル

これらの設定を以降では各々L+S (linear + singleton), RBF+S (RBF + singleton), L+SC (linear + singleton/combined) と参照する．L+S はベースラインの指標として用い，良く用いられるカーネルでの性能を示す指標として RBS+S を用いる．これらの指標に比べ，長さ 2 以上の関連パターンを新たな属性として取り込んだ L+SC がどのような性能を示すかに注目する．関連度としては χ^2, F 値, support difference を用いる．

分類結果に対する評価を表 3 に示す[10]．この表において最後以外の各行にはクラス（ニュースグループ）c を正クラスとして 2 クラス分類したときの F 値の平均が格納されている．一方，“*All*” と示された最後の行は全ての合成属性を使い多クラス分類したときの正解率を格納する．各セル中の $\pm$ に続く数字は交差検定における標準誤差である．各行では一番目，二番目に良い成績のものに各々 $**$, $*$ というマークを付けている．

表 3 より，RBF+S と L+SC がベースラインである L+S より良好な分類性能を示していることが分かる．更に RBF+S は 20 のニュースグループのうち半数で最良の成績を収めているが，多クラス分類（最後の行）では L+SC の分類性能は RBF+S に近いものになっている．また，関連パターンに基づく合成属性は理解可能性が高い点も重要である．例えば comp.graphics における分類が難しく，rec.sport.hockey における分類が易しい理由を表 2 中の関連パターン上位 25 個の F 値によって（間接的にではあるが）説明できる．また L+SC では関連度 χ^2, F 値，support difference の間に際立った違いは見られなかった．

5 おわりに

クラスが付与されたトランザクションデータに対して，本論文では効率的な上位 k 関連パターン発見手法である RP-growth を提案した．RP-growth は FP-growth で用いるデータ構造 (FP-tree) を引き継ぎ，反単調性を満たす関連度の上界を利用した分岐限定法に基づく探索を行う．分岐限定法における枝刈りは最小サポート上昇法に翻訳され，データベース縮約等，最小サポートの制約に基づく従来の頻出パターン発見手法における高速化技法をそのまま利用できる．更に関連パターン間で定義される「弱い」という関係を導入することにより，更なる効率化を図っている．また，テキスト分類タスクに関する実験結果によって RP-growth の効率性と有用性を示した．今後の課題としては関連パターンの有用性を更に実証する必要がある．具体的には，分類タスクに加え，RP-growth をベイジアンネットの感度分析 [11] やクラスタリング結果の自動的特徴づけへ

$-1, 1, 3\}$ から選ばれる．また k は $\{50, 100, 200, 300, 400, 500, 600, 700\}$ から，$\beta_{\min}$ は $\{0.2, 0.3, 0.4, 0.5\}$ から選ばれる．

[9] http://www.csie.ntu.edu.tw/~cjlin/libsvm

[10] 関連度 χ^2 に基づく RP-growth が comp.graphics および comp.windows.x において 2 時間以内に終了しなかったため，これらのニュースグループに対して合成属性は考慮していない．

表 3: SVM による分類における F 値と正解率 (accuracy) の平均 (%).

ニュースグループ	単体属性のみ		単体属性と合成属性の混合		
	線形カーネル	RBF カーネル	線形カーネル		
			χ^2	F 値	Support diff.
alt.atheism	**83.97±1.39	*83.78±1.37	83.68±1.14	83.68±1.42	83.76±1.41
comp.graphics	71.18±0.97	**72.96±1.13	N/A	*72.02±0.79	71.98±0.87
comp.os.ms-windows.misc	75.73±1.28	**77.57±1.03	75.84±1.20	76.17±1.31	*76.25±1.28
comp.sys.ibm.pc.hardware	68.75±1.89	*68.93±1.34	68.45±1.32	**69.26±1.60	68.92±1.80
comp.sys.mac.hardware	*79.69±1.23	**80.85±1.29	79.63±1.60	78.72±1.37	78.68±1.43
comp.windows.x	80.65±1.04	*80.68±1.20	N/A	80.63±1.15	**81.06±1.23
misc.forsale	79.59±1.17	**80.08±0.96	N/A	*79.85±1.23	79.83±1.12
rec.autos	81.20±1.16	**82.29±1.38	81.15±1.29	81.24±1.12	*81.47±1.24
rec.motorcycles	91.22±0.53	91.06±0.48	*91.70±0.46	91.67±0.57	**91.70±0.53
rec.sport.baseball	90.14±0.51	90.49±0.54	**90.72±0.52	90.39±0.48	*90.51±0.45
rec.sport.hockey	94.35±0.50	94.44±0.50	94.68±0.47	*94.89±0.53	**94.95±0.54
sci.crypt	91.27±0.60	91.31±0.64	**91.63±0.58	*91.61±0.56	91.39±0.61
sci.electronics	71.27±0.89	**73.65±1.58	71.56±1.00	73.21±0.99	*73.35±0.86
sci.med	85.98±0.78	*86.59±0.70	86.42±0.72	**86.60±0.70	86.42±0.74
sci.space	89.75±0.55	90.22±0.57	**90.85±0.43	*90.54±0.44	90.49±0.47
soc.religion.christian	85.41±0.65	**86.12±0.64	*86.12±0.66	86.03±0.55	85.90±0.58
talk.politics.guns	83.19±0.93	**84.62±1.10	*84.21±1.01	83.89±1.15	83.82±1.17
talk.politics.mideast	*92.65±0.72	**92.70±0.72	92.33±0.65	92.22±0.66	92.10±0.67
talk.politics.misc	76.34±1.23	**77.28±1.13	76.10±1.13	*76.96±1.25	76.62±1.50
talk.religion.misc	62.61±1.44	*62.75±1.50	**63.50±1.84	61.89±2.00	62.52±1.71
All	83.88±0.20	**84.95±0.22	84.48±0.13	84.73±0.22	*84.73±0.23

の適用を考えている.

参考文献

[1] S. D. Bay and M. J. Pazzani. Detecting group differences: mining contrast sets. *Data Mining and Knowledge Discovery*, Vol. 5, pp. 213–246, 2001.

[2] H. Cheng, X. Yan, J. Han, and C.-W. Hsu. Discriminative frequent pattern analysis for effective classification. In *Proc. of Int'l Conf. on Data Engineering (ICDE-07)*, pp. 716–725, 2007.

[3] K. W. Church and P. Hanks. Word association norms, mutual information, and lexicography. In *Proc. of the 27th Annual Meeting on Association for Computational Linguistics (ACL-89)*, pp. 76–83, 1989.

[4] G. Dong and J. Li. Efficient mining of emerging patterns: discovering trends and differences. In *Proc. of the 5th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD-99)*, pp. 43–52, 1999.

[5] H. Fan and K. Ramamohanarao. A Bayesian approach to use emerging patterns for classification. In *Proc. of the 14th Australasian Database Conf. (ADC-03)*, pp. 39–48, 2003.

[6] H. Fan and K. Ramamohanarao. Fast discovery and the generalization of strong jumping emerging patterns for building compact and accurate classifiers. *IEEE Trans. on Knowledge and Data Engineering*, Vol. 18, pp. 721–737, 2006.

[7] L. Geng and H. J. Hamilton. Interestingness measures for data mining: a survey. *ACM Computing Surveys*, Vol. 38, No. 3, pp. 1–32, 2006.

[8] J. Han, J. Pei, and Y. Yin. Mining frequent patterns witout candidate generation. In *Proc. of the 2000 ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD-00)*, pp. 1–12, 2000.

[9] J. Han, J. Wang, Y. Lu, and P. Tzvetkov. Mining top-K frequent closed patterns without minimum support. In *Proc. of the 2002 IEEE Int'l Conf. on Data Mining (ICDM-02)*, pp. 211–218, 2002.

[10] N. Japkowicz and M. Shah. *Evaluating Learning Algorithms: A Classification Prespective*. Cambridge U. Press, 2011.

[11] F. V. Jensen. *An Introduction to Bayesian Networks*. UCL Press, 1996.

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

[13] B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *Proc. of the 4th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD-98)*, pp. 80–86, 1998.

[14] C. Manning, P. Raghavan, and H. Shütze. *Introduction to Information Retrieval*. Cambridge U. Press, 2008.

[15] S. Morishita and J. Sese. Traversing itemset lattices with statistical metric pruning. In *Proc. of the 19th ACM-SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS-00)*, pp. 226–236, 2000.

[16] K. Ohara, M. Hara, K. Takabayashi, H. Motoda, and T. Washio. Pruning strategies based on the upper bound of information gain for discriminative subgraph mining. In *Proc. of the 2008 Pacific Rim Knowledge Acquisition Workshop (PKAW-08)*, pp. 50–60, 2009.

[17] P. Terlecki and K. Walczak. Efficient discovery of top-k minimal jumping emerging patterns. In *Proc. of the 6th Int'l Conf. on Rough Sets and Current Trends in Computing (RSCTC-08)*, pp. 438–447, 2008.

[18] 宇野毅明, 有村博紀. 頻出パターン発見アルゴリズム入門 — アイテム集合からグラフまで. 第 22 回人工知能学会全国大会 AI レクチャー, 2008.

[19] G. I. Webb and S. Zhang. *k*-optimal rule discovery. *Data Mining and Knowledge Discovery*, Vol. 10, pp. 39–79, 2005.

[20] S. Wrobel. An algorithm for multi-relational discovery of subgroups. In *Proc. of the 1st European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD-97)*, pp. 78–87, 1997.

[21] C. Yuan, X. Liu, T.-C. Lu, and H. Lim. Most relevant explanation: properties, algorithms, and evaluations. In *Proc. of the 25th Conf. on Uncertainty in Artificial Intelligence (UAI-09)*, pp. 631–638, 2009.

[22] A. Zimmermann and L. De Raedt. Cluster grouping: from subgroup discovery to clustering. *Machine Learning*, Vol. 77, pp. 125–159, 2009.