# 経路相関ルールマイニング

佐々木勇和†

† 大阪大学大学院情報科学研究科
E-mail: †sasaki@ist.osaka-u.ac.jp

**あらまし** グラフ相関ルールマイニングはグラフデータ内の相関関係を抽出するデータマイニング技術である．本稿では，新たな概念である経路相関ルールマイニングを提案する．経路相関ルールマイニングは大規模グラフ内に頻出する経路パターンの相関関係を発見を目的として，到達可能性経路を適用可能とすることで多様な規則を発見する．経路相関ルールマイニングが NP 困難であることを証明し，非単調性性質を活用したアルゴリズムを開発する．加えて，近似化と並列処理技術を開発し，スケーラブルかつ高速なマイニングを実現とする．大規模な実グラフを用いて評価実験を行い，提案技術の有用性と高速性を示す．

**キーワード** グラフ分析，相関ルールマイニング

## 1 はじめに

相関ルールマイニングは大規模なアイテムセット間の関係性を発見するデータマイニング技術である [1, 2]．相関ルールは $X \Rightarrow Y$ で表現され，$X$ と $Y$ はそれぞれ条件部と結論部と呼ばれ，非重複なセットになる．その有用性から多くの応用例が存在し，例えば，購買分析 [3] や，ウェブマイニング [4]，バイオインフォマティクス [5] に利用されてている．

**グラフ相関ルールマイニング．** グラフ相関ルールマイニングは単一の大規模グラフ内に存在する規則を発見することを目的としている [6–9]．グラフ相関ルールは $G_X \Rightarrow G_Y$ で表現され，$G_X$ と $G_Y$ はグラフパターンを表す．グラフデータが広く活用されているため，グラフ相関ルールマイニングはグラフの分析や知識発見に広く利用されている．グラフ相関ルールマイニングは下記のような応用例がある．

ソーシャル分析: ソーシャル分析は我々の生活を理解し向上させるために不可欠なデータ分析である．ソーシャル関係はグラフによりモデル化されるため，グラフ相関ルールマイニングが人の状況や関係性のパターンの規則性の発見に活用できる．例えば，健康 [10] や幸福 [11] がソーシャル関係により影響されることが知られており，グラフ相関ルールマイニング活用できる．例えば，“もし人が幸せなら，健康であり友人をもつ可能性が高い” といったルールを発見できる．

社会的差別チェック: データ内の社会的差別は公平な機械学習モデルを構築するために解消すべき課題の一つである解決すべき課題である [12]．グラフデータにおける機械学習モデルも差別的なバイアスに影響が及ぼされる [13]．グラフ相関ルールマイニングは差別的なバイアスを含む規則の発見が可能なため，差別的なバイアスを評価および取り除くことが可能である．

知識抽出: 知識ベースは属性付きの節点とラベル付きの枝から構成されるグラフで表現されることが多く，知識グラフ内のパターン抽出により様々な知見を発見できる．例えば，“ある人が特定の職をもつ祖先がいる場合，その人の職は祖先と同じである” というようなパターンにより，社会分析に活用できる．

**動機と研究課題．** 大規模グラフにおけるグラフ相関ルールマイニングはグラフ分析において基礎的な技術であるが，既存技術は上記の応用例に適用できない（詳細は 2 章）．[6–9]．既存技術は 4 つの観点から柔軟性が低い．まず，既存技術は結論部に含まれる節点集合が必ず条件部の節点集合のサブセットになっていることが求められる．つまり，条件部に含まれていない節点を含む結論部があるようなルールは対象外である．次に，条件部と結論部ともに制限があり，枝ラベルと属性を同時に考慮するようなグラフパターンのルールを指定できない．さらに，節点間の接続可能性のようなパターンは捉えることはできない．最後に，それぞれのアルゴリズムはある単一のグラフタイプ（ラベル無し枝など）を想定しており，より一般的なグラフタイプを対象にできない．

そのため，新たなグラフ相関ルールマイニング技術が必要である．三つの研究課題を挙げる．まず，条件部と結合部に制約がないかつ接続性をグラフパターンに含めることができる柔軟な技術が必要である．次に，一般的なプロパティグラフに対して全ての頻出集合を求めることが可能な効率的かつスケーラブルな技術が要求される．最後に，規則は知識抽出や分析のために有用であり，既存のルールの評価尺度である相対支持度やリフトが利用できることが求められる．

**貢献．** 新たなコンセプトである経路相関ルールマイニングを提案し，経路相関ルールを高速に発見する効率的かつスケーラブルなアルゴリズムを開発する．経路相関ルールマイニングは経路パターン間の規則の検出を目的とし，経路パターンは節点の属性と枝ラベルのシーケンスを表す．節点間のラベル制約付き接続性を経路パターンに含むことで接続性を規則性に含むことが可能である．このマイニングでは，経路パターン $p_X$ および $p_Y$ のソースとなる節点集合が支持度の閾値 $\theta$ より大きくなるルール $p_X \Rightarrow p_Y$ を検出する．経路相関ルールは，絶対および相対支持度や，確信度，リフトといった一般的な評価尺度を計算することができる．

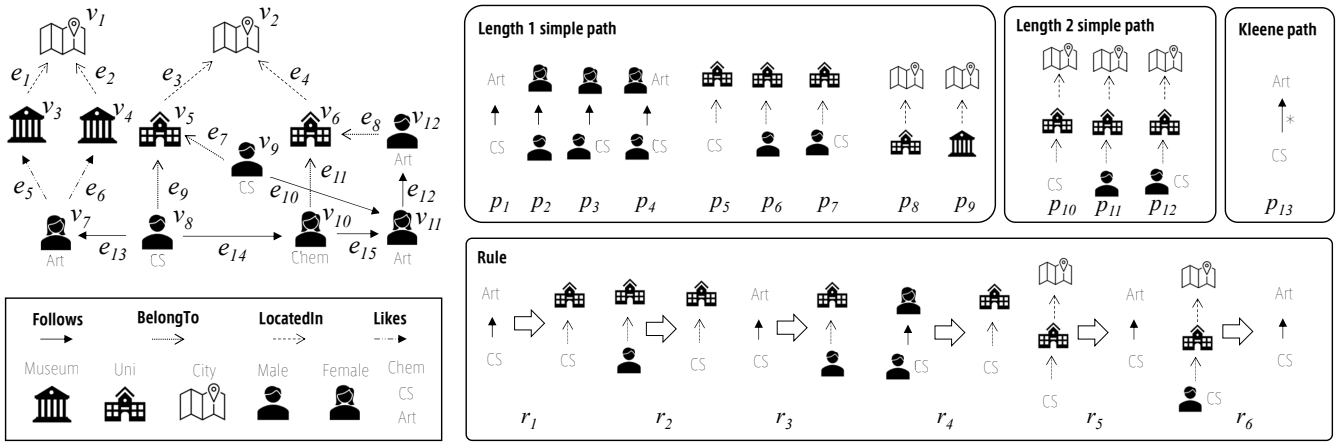



図 1: 経路相関ルールの例

**例 1**　図 *1* に経路相関ルールマイニングの例を示す．与えられたグラフは *12* 個の節点，*15* 個の枝，*4* つのタイプの枝ラベル，*8* つのタイプの節点の属性を含む．このグラフにおいて $\theta$ が *1* とした場合の経路相関ルールの例を示す．経路パターン $\langle\{Male\}, Follows, \{Art\}\rangle$ は $v_8$ と $v_9$ が始点となるため頻出な経路パターンとなる．同様に，$\langle\{CS\}, BelongTo, \{Uni\}\rangle$ も $v_8$ と $v_9$ にマッチする．$\langle\{CS\}, Follows, \{Art\}\rangle \Rightarrow \langle\{Male\}, BelongTo, \{Uni\}\rangle$ は経路相関ルールとして検出される．条件部と結論部に含まれる節点が非重複であり，既存研究では検出できないパターンである．

経路相関ルールマイニングが NP 困難問題であることを証明し，効率的に頻出な経路パターンとルールを発見するアルゴリズムを開発する．提案アルゴリズムは経路パターンにおける非単調性性質を活用し，頻出となりえない経路パターン候補を枝刈りする．経路パターンの枝刈りは，経路パターンを長くするのを止める垂直枝刈りおよび経路パターン内の節点の属性を増やすのを止める水平枝刈りの二つの種類を用いる．提案アルゴリズムは支持度，確信度，およびリフトなどのルールの評価尺度を厳密に計算できる．さらに，近似と並列化技術を開発し，スケーラブルな相関ルールの発見を実現する．近似化では，経路パターンの接尾の削減と節点のサンプリングにより計算コストを削減する．並列化では，経路パターンを計算に必要な節点毎のコストを推定し，コストが均等になるようにスレッドに割り当てることにより，スレッド間の計算コストを均等にする．評価実験では，3 つのグラフにおいて提案アルゴリズムが高速であることを示す．提案技術は厳密解計算において最大で 151 倍高速，近似解計算において微小な精度低下で 485 倍高速可能であることを示す．さらに，経路相関ルールが知識抽出とソーシャル差別確認において有用であることを示す．

**再現性**．提案手法のコードベースを Github にて公開する [1]．我々の知る限り，大規模なグラフに対するグラフ相関ルールマイニングにおける初めての公開コードである．

1：https://github.com/yuya-s/pathassociationrulemining

## 2 関 連 研 究

本章では，経路相関ルールマイニングに関する関連研究として，ひとつの大規模グラフに対する支持度計算とグラフ相関ルールマイニングについて述べる．

**ひとつの大規模グラフに対する支持度計算．**一般的なアイテムセットに対する支持度は一つのグラフに対して非単調性は保証できない．直感的には，経路は節点よりもパターンとして複雑であるが，経路は節点より多いことがありえる．非単調性を保持するために，様々な支持度が提案されている．例えば，maximum independent set based support (MIS) [14] minimum-image-based support (MNI) [15], minimum clique partition (MCP) [16], minimum vertex cover (MVC) [17], および maximum independent edge set support (MIES) [17] があげられる．共通する思想は，グラフパターンに含まれる節点が重複しないよう出現数を数えることである．これらの支持度には 3 つの欠点がある．まず，相対支持度に適応できない点である．これは，グラフ内に出現し得る最大のパターン数を求めるできなことに起因する．次に，時間計算量が大きい点である．例えば， MIS や MNI は NP 困難である．最後に，グラフパターンに含まれる節点と含まれない節点の解釈が困難であり，直感的ではない点である．相対支持度を計算可能な技術が提案されてはいるが，サブグラフマッチングが必要であるため計算量は大きい [6].

**グラフ相関ルールマイニング．**グラフ相関ルールマイニングはグラフデータ集合と一つの大規模グラフに対する技術の二つに大別できる．グラフデータ集合に対するアルゴリズム（例えば，[18–21]）は $\theta$ 個より大きいグラフに含まれるグラフパターンを発見することを目的とするのに対し，一つのグラフデータに対するアルゴリズム [6,7,9,22] は $\theta$ 回より大きい回数が一つのグラフに出現するグラフパターンを発見することを目的とする．非単調性は互いに適用することはできないため，それぞれに使われているアルゴリズムは他方に使用することはできない．

一つの大規模グラフに対するアルゴリズムは少数提案されている [6,7,9,22]．我々の知る限り，経路に着目したものや接続

表 1: 単一大規模グラフに対するグラフ相関ルールマイニングの比較．$V.G_X$ と $V.G_Y$ はそれぞれ $G_X$ と $G_Y$ に含まれる節点集合を表す．

| | グラフタイプ | $G_X$ | $G_Y$ | $G_X$ と $G_Y$ | 出力 |
|---|---|---|---|---|---|
| GPAR [6] | ラベル付き枝<br>ラベル付き節点 | サブグラフ | 単一の枝<br>もしくは空 | $V.G_Y \subseteq V.G_X$ | トップ $k$ 多様パターン |
| Extending GPAR [7] | ラベル無し枝<br>属性付き節点 | サブグラフ<br>(少なくとも一つの枝を含む) | サブグラフ<br>(少なくとも一つの枝を含む) | 接続している<br>かつ枝の重複無し | 頻出パターン |
| GAR [9] | ラベル付き枝<br>属性付き節点 | 属性無しサブグラフ | 単一の枝<br>もしくは単一の属性 | $V.G_X \subseteq V.G_Y$ | 応用を指定した頻出パターン |
| 提案 | ラベル付き枝<br>属性付き節点 | 経路 | 経路 | 始点が共通 | 頻出パターン |

可能性パターンに適応しているものは存在しない．表 1 に各手法の特徴をまとめる．

GPAR は初めて提案された一つの大規模グラフに対するグラフ相関ルールマイニングである [6]．Wang ら [7] は GPAR を拡張し，頻出パターンを取得可能なアルゴリズムを開発した．その際，条件部や結論部，支持度の計算方法を変更し，検出パターン数が少なるようにしている．Fan らは [9] は節点属性に対応可能なように GPAR を拡張している．応用に着目して結論部を事前に指定し，機械学習を用いて不要な節点や枝をグラフから取り除くことで計算量が小さくなる工夫をしている．他にも時系列グラフ [23] や量的サブグラフ [8] に GPAR を拡張した技術も開発されている．

**他のルール検出技術．** グラフからルールを検出技術は多く開発されており，例えばグラフ関数従属性 [24] や Horn ルール [25,26] が挙げられる．Horn ルールは RDF データにおいて，条件部が経路で，結論部が条件部に含まれる節点間の単一の枝となるルールである．これらの技術は一般的なプロパティグラフでは使用できない．

サブシーケンスマイニング [27–31] は，シーケンス内のサブシーケンスパターンの規則を発見する技術である．このマイニングは，グラフが経路グラフの場合におけるグラフ相関ルールであり，特殊例としていうことができるが，複雑なグラフには適用できない．

## 3 経路相関ルールマイニング

本章では，経路相関ルールマイニングを提案する．経路相関ルールマイニングは効果的に節点属性と枝ラベルのシーケンス間の規則性を発見することができる．

### 3.1 表　　記

グラフを $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{L}, \mathcal{A})$ とし，$\mathcal{V}$ は節点集合，$\mathcal{E} \subset \mathcal{V}\times\mathcal{L}\times\mathcal{V}$ は枝ラベル $\ell \in \mathcal{L}$, $\mathcal{L}$ を伴う枝集合，および $\mathcal{A}$ は属性集合を表す．枝 $e \in \mathcal{E}$ は $(v, \ell_e, v')$ と表し，枝ラベル $\ell_e$ を伴う節点 $v$ から節点 $v'$ とする．属性 $a \in \mathcal{A}$ はカテゴリカルな値とし，節点の特徴を表す．$A \subseteq \mathcal{A}$ は属性の集合とし，それぞれの節点 $v \in \mathcal{V}$ は属性集合 $A(v)$ をもつ．

経路は節点と枝のシーケンスとし，s $\langle v_0, e_0, v_1, \ldots, e_{n-1}, v_n\rangle$ と表記する．$n$ は経路長であり，$v_0$ と $v_n$ をそれぞれ始点と終点とする．

### 3.2 経路相関ルール

経路パターンを定義した後に，経路相関ルールマイニングについて説明する．

**経路パターン**．単純経路パターンと到達可能性経路パターンの 2 つを定義する．

- 単純経路パターンは節点属性 $A_i \subseteq \mathcal{A}$ の集合枝ラベル $\ell_i \in \mathcal{L}$ のシーケンス $\langle A_0, \ell_0, A_1, \ldots, \ell_{n-1}, A_n\rangle$ で表現される．節点 $v$ と単純経路パターン $p$ が与えられたとき，もし $v$ が全ての $i$ に対して $A_i{\subseteq}A(v_i)$ および $\ell_i{=}\ell_{e_i}$ を満たす経路の始点であるならば，$v$ は $p$ に適合するとする．
- 接続可能性経路パターンは 2 つの属性集合 $A_0, A_1 \subseteq \mathcal{A}$ および枝ラベル $\ell^* \in \mathcal{L}$ のトリプルである $\langle A_0, \ell^*, A_1\rangle$ で表現される．節点 $v$ と接続可能性経路パターン $p$ が与えられたとき，もし $v$ が $A_0 \subseteq A(v_0)$, $A_1 \subseteq A(v_n)$ および $\ell_i{=}\ell_{e_i}$ を満たす経路の始点であるならば，$v$ は $p$ に適合するとする．

$\mathcal{V}(p)$ および $|\mathcal{V}(p)|$ は経路パターン $p$ に適合する節点集合およびその数をそれぞれ表す．経路パターン内の $A$ は少なくとも一つの要素が入っているものとする．全ての属性集合が一つの要素のみを含む経路パターンを単位経路パターンと呼ぶ．

**定義 1 (支配)**　二つの経路パターン $p = \langle A_0, \ell_0, A_1, \ldots, \ell_{n-1}, A_n\rangle$ および $p' = \langle A'_0, \ell'_0, A'_1, \ldots, \ell'_{m-1}, A'_m\rangle$ が与えられたとき，$m \leqq n$, かつ 0 から $m$ までの全て $i$ において $\ell'_i = \ell_i$ と $A'_i \subseteq A_i$ が成り立つならば，$p$ は $p'$ を支配しているという．$p' \subset p$ は $p'$ が $p$ に支配されていることを表す．

直感的には，支配している経路パターンは支配されている経路パターンより複雑であることを示す．

**経路相関ルール**．経路相関ルールを以下と定義する．

**定義 2**　経路相関ルール $r$ は $p_X \Rightarrow p_Y$ で定義され，$p_X$ と $p_Y$ は経路パターンを表す．$p_X$ と $p_Y$ をそれぞれ条件部と結論部と呼ぶ．節点が $p_X$ と $p_Y$ 両方に適合する場合，$r$ に適合すると呼ぶ．

経路相関ルールは頻出性や条件付確率を評価することができる．ここで，経路相関ルールにおいてはホモモフィズムとするため，一つの経路が $p_X$ と $p_Y$ を共有していてもよい．

### 3.3 相関ルールの指標

経路相関ルールは一般的な相関ルールの指標を計算可能であ

る．支持度，確信度，リフトは既存の定義から自然に計算可能である．

支持度: 支持度はどの程度の節点が経路相関ルールに適合しているかを表す．絶対支持度と相対支持度の両方を下記で示す．ここで，多くのグラフ相関ルールマイニングは相対支持度を計算できない．これは，適合し得る最大のグラフパターン数の計算が困難なためである．

絶対支持度は下記となる．

$$ASupp(p_X \Rightarrow p_Y) = |\mathcal{V}(p_X) \cap \mathcal{V}(p_Y)|.$$

$|\mathcal{V}(p)|$ の最大となりうる値は節点数となるため，相対支持度は以下で計算できる．

$$RSupp(p_X \Rightarrow p_Y) = \frac{|\mathcal{V}(p_X) \cap \mathcal{V}(p_Y)|}{|\mathcal{V}|}.$$

確信度: 確信度は節点が $p_X$ に適合する場合に $p_Y$ が適合する確率を表す．確信度は以下の式で計算できる．

$$Conf(p_X \Rightarrow p_Y) = \frac{|\mathcal{V}(p_X) \cap \mathcal{V}(p_Y)|}{|\mathcal{V}(p_X)|}.$$

リフト: リフトは相対支持度と同様に多くのグラフ不相関ルールマイニングで計算できない指標である．リフトは下記の式で計算できる．

$$Lift(p_X \Rightarrow p_Y) = \frac{|\mathcal{V}(p_X) \cap \mathcal{V}(p_Y)| \cdot |\mathcal{V}|}{|\mathcal{V}(p_X)| \cdot |\mathcal{V}(p_Y)|}.$$

その他: 上記と同様の方法で，他の指標も定義できる．

### 3.4 問題定義

本稿で解く問題を下記で定義する．

**定義 3 (経路相関ルールマイニング問題)** *グラフ $\mathcal{G}$,（絶対もしくは相対）支持度の最小値 $\theta$，最大経路長 $k$ が与えられたとき，経路相関ルールマイニング問題は 3 つの条件 (1) $supp(r)$ が $\theta$ より大きい， (2) 経路長の最大値が $k$, かつ (3) $p_X$ および $p_Y$ はそれぞれ互いに支配されないを満たす全ての $r$ と $r$ の指標を計算する．$|\mathcal{V}(p)| > \theta$ を満たす経路パターンを頻出経路パターンと呼ぶ.*

この問題定義や経路パターンは応用に合わせて柔軟に変更できる．例えば，(3) の支配関係の条件を外すことや，経路パターン内の節点が空集合であることを許可するなどがあげられる．

**定理 1** *経路相関ルールマイニング問題は NP 困難である．*

簡易証明: 経路相関ルールマイニングを NP 困難問題のひとつである頻出サブシーケンス列挙問題 [32] に還元することで証明する．シーケンスおよびサブシーケンスは，それぞれグラフおよび経路パターンのサブクラスである．そのため，経路相関ルールマイニングを解ければ，頻出サブシーケンス列挙問題も解くことができる．したがって，経路相関ルールマイニングは NP 困難である． □

**特徴**. 経路相関ルールはアイテムセットに対する一般的な相関ルールよりも一般化されている．各節点と属性はトランザクションとアイテム集合とそれぞれ見做すことができ，枝が存在しないグラフでは経路相関ルールマイニングはアイテムセットに対する相関ルールと等価である．

## 4 アルゴリズム

本章では，経路相関ルールマイニングのための効率的なアルゴリズムについて述べる．経路相関ルールマイニング問題を解くためには，頻出経路パターンとそれらに適合する節点集合を列挙する必要がある．経路パターンの候補を削減するために，頻出な単位経路パターンをはじめに列挙し，非単調性を用いながら複雑な経路パターンの候補を生成する．

### 4.1 非単調性性質

非単調性性質は経路パターンの候補の削減に効果的に活用できる．経路パターンの非単調性性質は下記となる．

**定理 2** *もし $p' \subset p$ ならば，$\mathcal{V}(p') \supseteq \mathcal{V}(p)$ である.*

上記の定理から下記 2 つの補題を導出できる．

**補題 1** *経路パターンのプリフィックスが頻出ではないなら，経路パターン全体も頻出ではない.*

**補題 2** *もし $p$ が頻出ではない，かつ $p'$ が全ての $i$ に対して $A'_i \supseteq A_i$ と $\ell'_i = \ell_i$ を満たすならば，$p'$ は頻出ではない.*

これらの定理と補題は頻出になりえない経路パターンの枝刈りを可能とする．

### 4.2 ベースラインアルゴリズム

ベースラインアルゴリズムは 4 つの処理手順 (1) 頻出属性集合発見，(2) 頻出単純経路パターン発見，(3) 頻出接続可能性経路パターン発見，および (4) ルール発見で構成される．それぞれの処理手順で補題 1 を活用し，候補の削減を行う．それぞれについて以下で説明する．

（1） **頻出属性集合発見.** 頻出属性集合の集合 $\mathcal{P}_0$（長さ 0 の頻出経路パターン）を求める．この処理は，アプリオリ手法など既存の相関ルールマイニング技術を活用する．

（2） **頻出単純経路パターン発見.** この処理手順では，既に発見した頻出経路パターンを拡張することで頻出単純経路パターンを発見する．長さ $i$ の単純経路パターンの候補を経路パターン $p \in \mathcal{P}_{i-1}$，枝ラベル，および $A \subseteq \mathcal{A}$ を組合わせることで生成する．もし経路パターンが頻出であれば，その経路パターンを $\mathcal{P}_i$ に格納する．経路長が $k$ となるまで繰り返す.

（3） **頻出接続可能性経路パターン発見.** この処理手順では接続可能性経路パターンに適合する節点を探索する．頻出単純経路パターンの発見と異なる点は，各節点から $\ell^*$ を伴う枝のみを通って到達可能な節点集合を求める必要がある点である．

枝ラベルが与えられたとき，各節点から到達可能な節点集合を幅優先探索で列挙する．その後，頻出単純経路パターンの発

見と同様に，経路パターン $p \in \mathcal{P}_0$，枝ラベル，および $A \subseteq \mathcal{A}$ を組合わせることで候補を生成し，そのパターンが頻出であれば $\mathcal{P}^*$ に格納する．

（4） **ルール発見.** ルール発見の処理では，二つの頻出経路パターンに適合する節点を探索する．手順 (2) と (3) で発見した頻出経路パターン $p$ と $p'$ に対して，候補ルール $p \Rightarrow p'$ と $p' \Rightarrow p$ を生成し，両方に適合する節点数を数える．最後に，頻出であるルールの指標を計算する．

ベースラインアルゴリズムは補題 1 を用いて候補数を削減することができる．しかし，経路パターンとルールの候補数は未だ膨大であるため，さらなる候補の削減が必要とされる．

### 4.3 最適化手法

ベースラインアルゴリズムをさらに経路パターンとルールの候補を削減することにより高速化する．サフィックス削減を導入し，さらに，候補生成を最適化する手法を開発する．

**サフィックス削減.** 非単調性性質は頻出にならない経路パターンのプリフィックスを削減できるが，サフィックスを削減することはできない．そこで，サフィックス削減を提案する．

枝ラベル $\ell$ と属性 $a$ が与えられたとき，経路長 $n-1$ の任意の経路パターンに $\ell$ と $A$ が結合した場合の最大適合節点数を計算する．加えて，枝ラベル $\ell$ と属性 $a$ が与えられたとき，任意長の経路パターン $\langle A, \ell, \ldots \rangle$ に適合する最大の節点数を計算する．これらの最大節点数が $\theta$ 以下の場合，サフィックスを削減できる．

二つの値を定義する．一つ目は，節点集合 $A$ をもち枝ラベル $\ell$ の出次枝をもつ節点の数を $|\mathcal{V}(A,\ell)| = |\{v | (v, \ell_e, v') \in \mathcal{E} \wedge A(v) \supseteq A \wedge \ell_e = \ell\}|$ ととする．二つ目は枝ラベル $\ell$ をもち属性集合 $A$ をもつ節点に接続する枝の数を $|\mathcal{E}(A,\ell)| = |\{(v, \ell_e, v') | (v, \ell_e, v') \in \mathcal{E} \wedge A(v') \supseteq A \wedge \ell_e = \ell\}|$ とする．最大の適合節点数は下記の補題となる．

**補題 3** 経路長 $n$, 枝ラベル $\ell$, および属性集合 $A$ が与えられたとき, 長さ $n-1$ の経路パターンに $\ell$ と $A$ を結合した経路パターンに適合する最大の節点数は $|\mathcal{E}(A,\ell)| \cdot (d_m)^{n-1}$ である．$d_m$ は全ての節点における最大入次数を表す．

**補題 4** 枝ラベル $\ell$ および属性集合 $A$ が与えられたとき，経路パターン $p = \langle A, \ell, \ldots \rangle$ *is* $|\mathcal{V}(A,\ell)|$ に適合する最大節点数は $|\mathcal{V}(A,\ell)|$ である．

上記の補題より，経路パターンの候補生成時の枝ラベルと節点集合を削除できる．初めに，$|\mathcal{E}(A,\ell)| \cdot {d_m}^{n-1} > \theta$ を満たす場合，$(A,\ell)$ のペアは経路長 $n-1$ のサフィックス $\langle n, A_n \rangle$ となりうる．次に，$\ell$ が全ての $A$ に対して $\min(|\mathcal{V}(A,\ell)|, |\mathcal{E}(A,\ell)| \cdot {d_m}^{k-1}) \leqq \theta$ を満たさない場合，$\ell$ は経路パターンにおける $\ell_0$ および $\ell^*$ になりうる．サフィックスとなりうる枝ラベル集合と属性集合をそれぞれ $\mathcal{L}_F$ と $\mathcal{A}_T$ する．サフィックス枝刈りは経路パターン長を伸ばす際の候補生成削減に有効である．

**候補生成の最適化.** 非単調性性質とサフィックス枝刈りにより，(1) プリフィックスが頻出ではない経路パターン，(2) 支配している経路が頻出ではない経路パターン，および (3) 頻出経路パターンに係らないサフィックスによって，経路パターン候補を削減できる．最適化された候補生成では，二つの方法により経路パターンを拡張する．垂直型拡張は，ベースラインと同様に経路パターンの経路長を長くする拡張であり，水平型拡張は経路パターン内の属性集合を大きくする拡張である．候補から枝刈りされた経路パターンは頻出とはなりえないため，精度が劣化することは無い．

提案アルゴリズムを候補生成の最適化に基づいて拡張する．頻出属性集合発見において，$\mathcal{L}_F$ と $\mathcal{A}_T$ を求める．$\mathcal{L}_F$ と $\mathcal{A}_T$ は，全ての枝ラベルに対して $\mathcal{A}_T$，その後 $\mathcal{L}_F$，最後に $\mathcal{A}_T$ を $\mathcal{L}_F$ により再計算することで求める．単純経路パターン発見では，長さ 1 の単位経路パターンを発見し，そこから $\mathcal{L}_F$ と $\mathcal{A}_F$ をサフィックスに追加することで垂直型拡張を行う．垂直型拡張では二つの頻出経路パターンを統合し，経路パターン内の属性を増やしていく．接続性経路パターンでも同様に，補題 2 を用いて水平型拡張を実施する．

ルール発見では，補題 1 と補題 2 を用いて単位経路パターンからなるルールをまず発見し，頻出ルール内の経路パターンを垂直型および水平型拡張する．具体的には，$p \Rightarrow p'$ が頻出であるならば，$p_v \Rightarrow p'$, $p_h \Rightarrow p'$, $p \Rightarrow p'_v$, および $p \Rightarrow p'_h$ をルールの候補として逐次生成していく．$p_v$ と $p_h$ はそれぞれ $p$ から垂直型と水平型拡張したものを表す．

## 5 近似手法

提案アルゴリズムは候補集合を効果的に削減することができるためベースラインと比べて高速な実行が可能である．しかし，候補集合は未だ膨大となるため，大規模グラフに対してスケーラブルとはいえない．そこで，近似手法として，候補削減と節点サンプリングを開発する．

### 5.1 候補削減

最適化手法は補題 3 に基づいた候補枝刈りにより候補集合を削減できる．サフィックス削減は精度の低下しないことを目的としてるため，頻出集合に入らないサフィックスも多く残ってしまう．スケーラブルなマイニングのために，頻出集合に含まれない可能性が高いサフィックスを削減する技術を提案する．

補題 3 に基づいたサフィックス候補を近似化する．候補削減パラメータ $\psi$ $(0.0 \leqq \psi \leqq 1.0)$ が与えられたとき，もし $|\mathcal{L}(A,\ell)| \cdot (d_m)^{\psi(n-1)}$ が $\theta$ を超えないなら，サフィックス $\langle \ell, A \rangle$ を候補から削除する．多くの節点は最大次数より小さい次数であるため，精度への影響を小さく削減することができる．

**定理 3** 頻出経路集合 $p$ が与えられたとき, $p$ が欠損する確率は $P\left(\theta \geqq \frac{|\mathcal{V}(p)|(d_m)^{\psi(n-1)}}{P_m(d_p)^{n-1}}\right)$ である．$P_m$ は $p$ が節点に適合する確率，$d_p$ は $p$ に適合する経路における経路上の節点の平均出次数を表す．

この定理により，$|\mathcal{V}(p)|$ が大きい，最大次数と実際に経路パターン上の節点との差が大きい場合に，欠損する確率が下がる．

大規模グラフはこのような特徴をもつことが多いため効果的である．この近似は偽陽性なしの近似化であるため，非頻出な経路ルールがマイニング結果に入ることはない．

### 5.2 節点サンプリング

サンプリングはデータマイニングの近似化に頻繁に使われる技術である [9, 33–35]．提案する近似化手法は，経路相関ルールマイニングの特徴に合わせて，始点として計算する節点をサンプリングすることで，計算コストを削減する．

サンプリングとして，節点の属性集合に応じた層化サンプリングを用いる [36]．まず頻出属性をもたない節点は排除する．次に，節点を属性毎にグループ化し，サンプリング率 $\rho$ でグループ毎に節点を選択する．経路パターン $p$ は下記の式で推論する．

$$\widetilde{|\mathcal{V}(p)|} = \frac{|\mathcal{V}_s(p)|}{\rho}. \tag{1}$$

$\mathcal{V}_s$ はサンプリングされた節点集合を表す．

候補削減と比べて，サンプリングは偽陽性を含む可能性がある．一方で，候補削減が機能しない場合でも計算コストを削減することができる．

## 6 並 列 化

提案アルゴリズムは並列化により高速化可能である．我々の並列化は，スレッド毎に計算コストが均等となるように節点を分配し，スレッド間の計算コストの均等化を目指す．各節点は属集合と枝をもち，それぞれが頻出なものと頻出ではないものが存在する．もし多くの頻出な属性と枝をもつ節点が偏ったスレッドに配置された場合，そのスレッドの計算コストが大きくなり，全体として実行時間が長くなってしまう．

コストベースのパーティショニングとして，スレッド数 $N$ が与えられたときに，節点毎に計算コストを見積もり，計算コストの合計値が等しくなるように $N$ グループに節点を分割する．コストの見積もりは下記の式で行う．

$$C(v) = d_F(v) \cdot |A_F(v)| \tag{2}$$

$d_F(v)$ および $|A_F(v)|$ は節点 $v$ のそれぞれ頻出な枝ラベルの出次数および頻出な属性数を表す．

節点の分割では貪欲法を用いて計算する．節点をコスト毎に降順で並び替え，順に割当てられた節点のコストが最も小さいスレッドに割り当てていく．この並列化では，コストの見積もりに $O(|\mathcal{V}|)$，節点の分割に $O(N|\mathcal{V}|\log|\mathcal{V}|)$ 係る．計算コストが非常に小さいため，大規模グラフにおいて有効である．

## 7 評 価 実 験

評価実験では，提案アルゴリズムの効率性，近似手法の精度，および経路相関ルールマイニングの効果を検証する．全てのアルゴリズムは C++で実装し，512GB のメモリおよび Intel(R) Xeon(R) CPU E5-2699v3 を搭載する Linux 計算機にて実験を行う．

表 2: データ統計

| | $\lvert\mathcal{V}\rvert$ | $\lvert\mathcal{E}\rvert$ | $\lvert\mathcal{L}\rvert$ | $\lvert\mathcal{A}\rvert$ | Avg. $\lvert A(v)\rvert$ |
|---|---|---|---|---|---|
| Nell | 46,682 | 231,634 | 821 | 266 | 1.5 |
| DBpedia | 1,477,796 | 2,920,168 | 504 | 239 | 2.7 |
| Pokec | 1,666,426 | 34,817,514 | 9 | 36302 | 1.1 |

### 7.1 実 験 設 定

**データセット**．枝ラベルと節点属性を含む 3 つのグラフデータ Nell [37][2], DBpedia [38][3], および Pokec[4]を用いる．表 2 にデータの統計を示す．

**比較手法**．*Baseline* は最適化手法を用いないアルゴリズム，*Ours* は近似手法を用いない提案アルゴリズムを示す．*Ours w/ CR* および *Ours w/ SA* は候補削減と節点サンプリングをそれぞれの近似手法用いた提案手法，*Ours w/ CR+SA* は両方の近似手法を用いた提案手法を示す．全ての手法で並列化手法を用いて並列実行する．ここで，既存手法は経路相関ルールマイニングに適用できないため比較手法として用いない [6, 7, 9, 22]．

**パラメータ**．評価実験におけるデフォルトパラメータを述べる．最小支持度として，絶対支持度を用いて Nell，DBpedia，および Pokec において，それぞれ 1,000，120,000，および 45,000 とする．近似手法のパラメータである $\psi$ と $\rho$ を 0.4 とする．最大経路長 $k$ は 2 とし，32 スレッドを用いて並列化する．絶対支持度，相対支持度，確信度，およびリフトを指標として計算する．

### 7.2 効 率 性

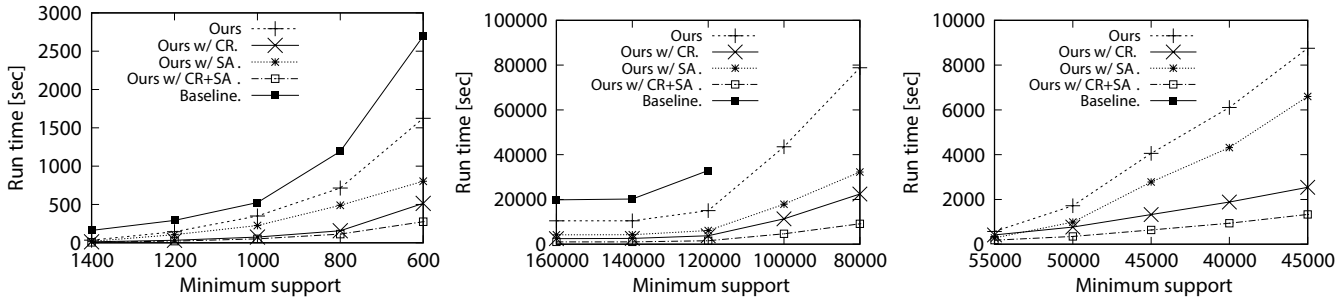
まず，パラメータを変化させ，提案手法の効率性を評価する．

**最小支持度 $\theta$ の影響**．最小支持度は発見するルールに影響する．最小支持度が減少すると発見ルール数が増加し，結果として計算コストが大きくなる．図 2 にそれぞれのデータにおいて最小支持度を変化させた場合の実行時間を示す．最小支持度が減少するにつれて，全ての手法で実行時間が長くなる．提案手法はベースラインと比較して高速である．特に，Pokec において，属性数が多いためベースラインは支持度が大きくても実行が 24 時間以内に終了しなかった．経路パターンの候補を効果的に削減することにより，提案手法が効率化されていることがわかる．

近似手法はさらに計算コストを削減できる．近似手法において，候補削減は候補サイズの削減，節点サンプリングは計算対象となる節点数の削減という，時間計算量において異なる項目を削減する．そのため，両方を用いた場合に片方を用いる場合よりも十分に計算コストを削減できる．それぞれの近似手法を比べると，候補削減の方が効果的である．これは，経路パターン候補の方が節点に比べて大きいためである．

**スレッド数の影響**．提案アルゴリズムは並列処理可能であるため，スレッド数が多くなればなるほど実行時間が減少する．表 3 にスレッド数を変化させた場合の実行時間を示す．厳密およ

2：https://github.com/GemsLab/KGist/blob/master/data/nell.zip
3：https://github.com/GemsLab/KGist/blob/master/data/dbpedia.zip
4：https://snap.stanford.edu/data/soc-pokec.html




(a) Nell (b) DBpedia (c) Pokec

図 2: 最小支持度 $\theta$ の実行時間への影響．24 時間以内に終了した点のみをプロットする．

表 3: スレッド数の実行時間への影響 [sec]

| Dataset | Method | Number of threads | | |
|---|---|---|---|---|
| | | 8 | 16 | 32 |
| Nell | Ours | 754.6 | 496.2 | 348.7 |
| | Ours w/ CR+SA | 74.0 | 63.2 | 51.7 |
| DBpedia | Ours | 59,603 | 29,971 | 15,028 |
| | Ours w/ CR+SA | 6,037 | 3,042 | 1,529 |
| Pokec | Ours | 10,039 | 6,197 | 4,038 |
| | Ours w/ CR+SA | 1,981 | 1,103 | 634 |

表 4: メモリ使用量 [GB]

| | Nell | DBpedia | Pokec |
|---|---|---|---|
| Ours | 11.1 | 23.9 | 6.4 |
| Ours w/ CR | 4.2 | 21.5 | 6.4 |
| Ours w/ SA | 5.6 | 23.8 | 6.1 |
| Ours w/ CR+SA | 2.4 | 21.5 | 6.1 |

表 5: 精度

| | Nell | | DBpedia | | Pokec | |
|---|---|---|---|---|---|---|
| | 再現率 | 適合率 | 再現率 | 適合率 | 再現率 | 適合率 |
| CR | 0.233 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| SA | 0.9999 | 0.999 | 1.0 | 1.0 | 0.967 | 0.983 |
| CR+SA | 0.233 | 0.998 | 1.0 | 1.0 | 0.967 | 0.983 |

び近似アルゴリズムの両方で，線形に実行時間が減少していることがわかる．これより，提案アルゴリズムはスケーラブルに経路相関ルールを発見できることが確認できる．

**メモリ使用量**．表 4 に提案アルゴリズムのメモリ使用量を示す．DBpedia が最大のメモリ使用量，Pokec が最小のメモリ使用量となっている．このことから，グラフサイズより候補集合のサイズや，発見した頻出経路パターン数とルール数がメモリ使用量に大きく影響することがわかる．近似手法はメモリ使用量も削減できている．候補削減では候補数を，サンプリングでは経路パターンを格納する節点数が削減している．Nell のように経路パターン数が大きいグラフにおいて，近似手法がメモリ使用量の観点で有効であることがわかる．

### 7.3 精　　度

近似手法は実行時間が短くなるが，精度が低くなりすぎる場合，効果的とはいえない．そこで，近似手法の精度を評価する．精度の評価では，厳密解と比較して再現率と適合率を用いる．

表 5 にそれぞれのデータセットにおける近似手法の再現率と適合率を示す．DBpedia と Pokec では，近似制度が非常に高い一方で，Nell では候補削減の再現率が低い．Nell では，最大次数と経路パターンに関連する次数の差が小さいため，候補削減における欠損する確率が高い．提案する近似手法はグラフサイズが大きい場合において特に有効であることがわかる．

### 7.4 有 用 性

最後に，経路相関ルールの有用性について検証する．経路相関ルールは特定の応用タスクを想定しないため，定量的な評価として Pokec のバイアス評価を示す．

経路相関ルールを用いて Pokec 内のバイアスとして友人の学歴が性別間で異なるかを検証する．具体的には，$\langle\{\texttt{gender}\}\rangle \Rightarrow \langle\{\}, \texttt{follows}, \{\texttt{education:A}\}\rangle$ のルールにおいて，男性と女性間で検出される数が異なるかを調べる．`gender` と `education:A` は性別および学歴に関する節点属性を表す．このテンプレートに関する経路相関ルールを 767 件検出した．男性/女性毎の絶対支持度の総和はそれぞれ 464,725 と 466,725 であり，確信度の総和はそれぞれ 2.445 と 2.064 である．男性と女性のそれぞれの節点数はそれぞれ 464,725 と 466,725 であり，女性の方が多い．それにも関わらず，絶対支持度がほぼ同等かつ確信度が男性の方が大きいため，男性の友人の方がより学歴を登録していることが多いといえる．そのため，Pokec には性別間で学歴に関するバイアスがあるといえる．

## 8 ま と め

本稿では，新しいコンセプトとして経路相関ルールマイニングを提案した．経路相関ルールマイニングでは，ひとつの大規模グラフから経路間の規則の発見を可能とする．さらに，効率かつスケーラブルなアルゴリズムを開発した．評価実験において，3 つの実グラフを用いて経路相関ルールの有用であること，および提案アルゴリズムが効率的かつ高精度であることを確認した．

本研究はいくつかの拡張の方向性を残している．まず一つ目は，頻出なルール検出ではなく，経路パターンの特徴を捉えることができる新たな評価指標を定義し，上位 $k$ 件のルールを発

見するように拡張することである．次に，経路パターン自体を拡張し，例えば，属性が空の経路や終点節点が同一であるようなルールの発見が考えられる．最後に，提案アルゴリズムは非頻出な経路パターンの計算が必要な指標（例えば，$\chi^2$）を計算することができないため，後処理を追加することでより多様な指標に対応することが考えられる．

## 謝　　辞

本研究は JSPS 科学研究費 JP17H06099 および JP20H00584 の支援によって行われた．

## 文　　献


[1] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *SIGMOD*, pp. 207–216, 1993.

[2] Qiankun Zhao and Sourav S Bhowmick. *Association rule mining: A survey*. Nanyang Technological University, 2003.

[3] Manpreet Kaur and Shivani Kang. Market basket analysis: Identify the changing trends of market data using association rule mining. *Procedia computer science*, Vol. 85, pp. 78–85, 2016.

[4] C-H Lee, Y-H Kim, and P-K Rhee. Web personalization expert with combining collaborative filtering and association rule mining technique. *Expert Systems with Applications*, Vol. 21, No. 3, pp. 131–137, 2001.

[5] Saurav Mallik, Anirban Mukhopadhyay, and Ujjwal Maulik. Ranwar: rank-based weighted association rule mining from gene expression and methylation data. *IEEE transactions on nanobioscience*, Vol. 14, No. 1, pp. 59–66, 2014.

[6] Wenfei Fan, Xin Wang, Yinghui Wu, and Jingbo Xu. Association rules with graph patterns. *PVLDB*, Vol. 8, No. 12, pp. 1502–1513, 2015.

[7] Xin Wang, Yang Xu, and Huayi Zhan. Extending association rules with graph patterns. *Expert Systems with Applications*, Vol. 141, p. 112897, 2020.

[8] Wenfei Fan, Yinghui Wu, and Jingbo Xu. Adding counting quantifiers to graph patterns. In *SIGMOD*, pp. 1215–1230, 2016.

[9] Wenfei Fan, Wenzhi Fu, Ruochun Jin, Ping Lu, and Chao Tian. Discovering association rules from big graphs. *PVLDB*, Vol. 15, No. 7, pp. 1479–1492, 2022.

[10] James S House, Karl R Landis, and Debra Umberson. Social relationships and health. *Science*, Vol. 241, No. 4865, pp. 540–545, 1988.

[11] Max Haller and Markus Hadler. How social relations and structures can produce happiness and unhappiness: An international comparative analysis. *Social indicators research*, Vol. 75, No. 2, pp. 169–216, 2006.

[12] Jeffrey Dastin. Rpt-insight-amazon scraps secret ai recruiting tool that showed bias against women. *Reuters*, 2018.

[13] Joseph Fisher, Dave Palfrey, Christos Christodoulopoulos, and Arpit Mittal. Measuring social bias in knowledge graph embeddings. *arXiv preprint arXiv:1912.02761*, 2019.

[14] Natalia Vanetik, Ehud Gudes, and Solomon Eyal Shimony. Computing frequent graph patterns from semistructured data. In *ICDM*, pp. 458–465, 2002.

[15] Björn Bringmann and Siegfried Nijssen. What is frequent in a single graph? In *PAKDD*, pp. 858–863, 2008.

[16] Toon Calders, Jan Ramon, and Dries Van Dyck. Anti-monotonic overlap-graph support measures. In *ICDM*, pp. 73–82, 2008.

[17] Jinghan Meng and Yi-cheng Tu. Flexible and feasible support measures for mining frequent patterns in large labeled graphs. In *SIGMOD*, pp. 391–402, 2017.

[18] Akihiro Inokuchi, Takashi Washio, and Hiroshi Motoda. An apriori-based algorithm for mining frequent substructures from graph data. In *PKDD*, pp. 13–23.

[19] Yiping Ke, James Cheng, and Jeffrey Xu Yu. Efficient discovery of frequent correlated subgraph pairs. In *ICDM*, pp. 239–248, 2009.

[20] Xifeng Yan and Jiawei Han. gspan: Graph-based substructure pattern mining. In *ICDM*, pp. 721–724, 2002.

[21] Md Samiullah, Chowdhury Farhan Ahmed, Anna Fariha, Md Rafiqul Islam, and Nicolas Lachiche. Mining frequent correlated graphs with a new measure. *Expert systems with applications*, Vol. 41, No. 4, pp. 1847–1863, 2014.

[22] Wenfei Fan and Chunming Hu. Big graph analyses: From queries to dependencies and association rules. *Data Science and Engineering*, Vol. 2, No. 1, pp. 36–55, 2017.

[23] Mohammad Hossein Namaki, Yinghui Wu, Qi Song, Peng Lin, and Tingjian Ge. Discovering graph temporal association rules. In *CIKM*, pp. 1697–1706, 2017.

[24] Wenfei Fan, Chunming Hu, Xueli Liu, and Ping Lu. Discovering graph functional dependencies. *TODS*, Vol. 45, No. 3, pp. 1–42, 2020.

[25] Frank Manola, Eric Miller, Brian McBride, et al. Rdf primer. *W3C recommendation*, Vol. 10, No. 1-107, p. 6, 2004.

[26] Christoph Schmitz, Andreas Hotho, Robert Jäschke, and Gerd Stumme. Mining association rules in folksonomies. In *Data science and classification*, pp. 261–270. 2006.

[27] Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In *Proceedings of the eleventh international conference on data engineering*, pp. 3–14, 1995.

[28] Mohammed Javeed Zaki, Neal Lesh, and Mitsunori Ogihara. Planmine: Sequence mining for plan failures. In *KDD*, pp. 369–373, 1998.

[29] Sebastian Nowozin, Gokhan Bakir, and Koji Tsuda. Discriminative subsequence mining for action classification. In *ICCV*, pp. 1–8, 2007.

[30] Philippe Fournier-Viger, Cheng-Wei Wu, Vincent S Tseng, Longbing Cao, and Roger Nkambou. Mining partially-ordered sequential rules common to multiple sequences. *TKDE*, Vol. 27, No. 8, pp. 2203–2216, 2015.

[31] Philippe Fournier-Viger, Ted Gueniche, Souleymane Zida, and Vincent S Tseng. Erminer: sequential rule mining using equivalence classes. In *International Symposium on Intelligent Data Analysis*, pp. 108–119, 2014.

[32] Guizhen Yang. The complexity of mining maximal frequent itemsets and maximal frequent patterns. In *KDD*, pp. 344–353, 2004.

[33] Sau Dan Lee, David W Cheung, and Ben Kao. Is sampling useful in data mining? a case in the maintenance of discovered association rules. *Data Mining and Knowledge Discovery*, Vol. 2, No. 3, pp. 233–262, 1998.

[34] Hannu Toivonen, et al. Sampling large databases for association rules. In *VLDB*, Vol. 96, pp. 134–145, 1996.

[35] Wenfei Fan, Ziyan Han, Yaoshu Wang, and Min Xie. Parallel rule discovery from large datasets by sampling. In *SIGMOD*, pp. 384–398, 2022.

[36] S Thompson. *Sampling*. Wiley, 2002.

[37] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka, and Tom M Mitchell. Toward an architecture for never-ending language learning. In *AAAI*, pp. 1306–1313, 2010.

[38] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pp. 722–735. 2007.