

量的データを表す属性の同一性判定

藤岡 周平[†] 加藤 誠^{††} 吉川 正俊^{†††}

[†] 京都大学工学部情報学科

^{††} 京都大学国際高等教育院データ科学イノベーション教育研究センター

^{†††} 京都大学大学院情報学研究科

E-mail: [†]fujioaka@db.soc.i.kyoto-u.ac.jp, ^{††}kato@dl.kuis.kyoto-u.ac.jp, ^{†††}yoshikawa@i.kyoto-u.ac.jp

あらまし 本論文では2つの表とその表中の各1つの量的データを表す属性を入力とし、それら2つの属性が同一か否かを判定する研究に取り組む。既存の質的データに対するスキーママッチング手法とは異なり、提案手法では量的データを表す属性を対象とするために、下記の3つの改良を行った: (1) 異なる表において同一の実体を含むタプルを特定し比較, (2) 値の重複を利用して同一性を判定する際に数値の誤差の許容, (3) 入力された2つの表以外の量的データを含む表を用いた予測による同一性判定。Webから得られた量的データを含むテーブルに対する実験の結果、提案手法が同一の実態を表すタプルの割合が低い場合において、ベースライン手法を精度の面で大きく上回ることが示された。

キーワード 量的データ, 情報統合, スキーママッチング

1. はじめに

近年、官公庁や企業などによって「オープンデータ」が推進されており、研究や調査、マーケティング等の種々の目的のために、多様な統計データが利用可能になってきている。例えば、政府統計の総合窓口 (e-Stat)^(注1) では、国勢調査や労働力調査といった統計調査の結果や、農林業センサスや工業統計調査といった産業統計等を CSV や Excel, PDF 等の形式で公開している。また、国外においても London Datastore^(注2) や Eurostat^(注3) といった様々な統計情報サイトが存在している。これに加えて、統計データを利用した調査も盛んに行われている。例えば、米国の PredPol^(注4) では、過去の犯罪発生データをもとにして調査を行い、予測モデルを構築することで、次に犯罪がどこで発生しそうかを地図上に表示し、より高い精度の犯罪予測を実現している。また、e-Stat で公開されている統計データを引用している論文を Google Scholar で検索すると、3,560 件もの検索結果が見つかることから、論文や分析における Web 上のデータの有益性も見て取れる。

しかしながら、統計データはオープンデータとして公開されているものの、統一的なオントロジーで記述されていない。オープンデータにおいては、データ間の関係性を記述するために RDF の利用が推奨されているが、我々が e-Stat で公開されている統計データファイルを無作為に 3,233 件分調べたところ、多くのデータ (調べたファイルの約 75%) は Excel ファイルによってまとめられていた (ただし、複数のファイル形式公開による重複を含む)。また、同種のデータが異なる Web サイトにて公開されていることも問題である。このため、同じ属性が異

表1 トヨタの車種ごとの燃費

車種	燃費
アクア	34.4
スベイド	22.2
タンク	21.8
パッソ	28.0
...	...

表2 ホンダの車種ごとの燃費

車名	燃料あたりの走行距離
アコード	31.6
ヴェゼル	27.0
NSX	12.4
オデッセイ	26.0
...	...

なる言語表現によって記述されている場合があり、複数 Web サイトに散在するデータを集約することが困難である。この問題は、各々の企業や地方自治体、民間組織などによって集計された統計データを扱う際に顕著である。データを収集・公開する組織は、データを自組織の Web サイトに公開することが多く、組織が異なれば、自然と公開 Web サイトも異なり属性表現の一貫性が保たれなくなる。表1、表2に同じ属性が異なる言語表現によって記述されている場合の例を示す。表1はトヨタ、表2はホンダの車種ごとの燃費を表す。どちらも車種の燃費という同一のデータを表しているが、燃費を表すカラムが表1では「燃費」、表2では「燃料あたりの走行距離」となっており、カラム名が統一されていない。

そこで、本論文では、2つの表とその表中の各1つの量的データ属性を入力とし、それら2つの属性の同一性判定に取り組む。ここで、量的データ属性とは、気温や体重といった間隔尺度または比例尺度によって測られる数量を表す属性を指す。この問題を解決することで、異なる Web サイトに存在する表でも、同じ属性を特定できるようになる。これにより、同一属性のカラムをマージし、異なる Web サイトに散在するデータに対して集約操作を行うことができる。この問題はスキーママッチングの問題として広く知られており、多くの既存研究が存在する [3, 4, 7, 8, 11]。既存の属性の同一性判定手法では、質的データ (血液型やアンケートの段階評価といった名義尺度と順序尺度によって測られる値) を表す属性を対象とし、属性内の値を手がかりとして、その属性が何であるかを推定し属性の同一性

(注1) : <https://www.e-stat.go.jp/>

(注2) : <https://data.london.gov.uk/>

(注3) : <http://ec.europa.eu/eurostat>

(注4) : <http://www.predpol.com/>

判定を行っている。例えば、「東京」や「北京」などの値が含まれる属性であれば、知識ベースなどを利用することで、その属性が「首都」だと推定することができる。しかしながら、量的データの属性を対象とした場合、属性内の値を手がかりにして属性を推定することは困難である。例えば、表1の「34.4」や表2の「31.6」などの値から、その属性が「燃費」であることを推測するためには、事前に属性「燃費」の値を全て知っておく必要があるが、そのような場合には、そもそもスキーママッチングを行う必要がない。上記の理由から、既存の質的データに対する属性の同一性判定手法は適しておらず、量的データ属性を対象とした属性の同一性判定手法は我々の知る限りにおいて提案されていない。

提案手法では、量的データを表す属性を対象とするために、既存手法に対して下記の3つの改良を行った：(1)異なる表において同一の実体を含むタプルを特定し比較(2)値の重複を利用して同一性を判定する際に数値の誤差の許容(3)入力された2つの表以外の量的データを含む表を用いた予測による同一性判定。(1)は、実体を一意に特定できるような属性集合、すなわち、候補キーの値が一致したタプルが存在すれば、それらは同一の実体に対応するはずであり、同一性判定の対象となる属性の値が一致するかを調べることで同一性判定が可能になる。(2)は、値が同一であるか否かを判定する際に、完全な一致のみを考えるのではなく、ある程度の値の差異を許容する。(3)は、前述のように、入力は2つの表とその表中の各1つの量的データ属性であるが、それに加えて他の表の量的データ情報を利用して入力された量的データ属性の値を予測する関数を作成し、その関数による予測値と実際の値を比較する。前提として、ある量的データの値は必ずある質的データの値が一意に特定する実体に関連し、実体によって区別可能であるとする。同じ質的データの値を含むタプルにおいて、同一性判定を行いたい属性の値 v が別の属性の値を表すベクトル $\mathbf{w}$ と高い相関を持つと仮定した場合、ベクトル $\mathbf{w}$ から値 v をタプルごとに推定することが可能となる。その推定値と同一性判定を行う属性中の実際の値をタプルごとに比較することで、同一性判定を行うことができる。

実験では、Web上から取得した国と日本の都道府県に関する統計データを用い、属性の同一性を正確に判定できるかを評価した。これらの表をランダムに分割し、分割された2つの表のある属性ペアが同一であるかどうかを判定した。もし2つの表が元々同じ表であった場合は同一と判定されるべきであり、一方、別の表であった場合は同一でないとして判定されるべきである。提案手法に加えて、いくつかのベースライン手法を用い、それぞれの精度を比較した。その結果、同じ実体のタプルの割合が低い場合において、提案手法が他のベースライン手法を精度の面で大きく上回ることが示されたため、提案手法は量的データに対して有効であると考えられる。

この論文により(1)量的データ属性を対象とした、属性同一性を判定する問題を提案(2)予測関数の利用を提案(3)Web上から取得した統計データに対する実験で、量的データのスキーママッチングにおける属性同一性の判定タスクにおいて予

測関数利用の有効性の実証という貢献がなされた。

本論文の構成は以下の通りである。2節ではWebからの表抽出、および、表の解釈に関する関連研究について述べる。3節では問題設定と既存の属性の同一性判定手法について述べた後、提案手法である量的データに特化した同一性判定手法、および、そのアルゴリズムを説明する。4節では実験とその結果を示す。最後に、5節では今後の課題と共に本論文の結論を述べる。

2. 関連研究

本研究には、Webからの表抽出、および、表の解釈についての関連研究が存在する。また、スキーママッチングの問題も本研究に関連する。スキーママッチングの主な手法として、言語的マッチング、補助情報利用マッチング、インスタンスベースマッチング、構造ベースマッチング、制約ベースマッチング、ルールベースマッチングという6つの方法がある[1]。量的データ属性に対しては、質的データ属性に対する手法をそのまま適用しても、マッチングの手がかりとして利用できるカラム名となるため、精度が低くなると予想される。そこで、我々は、量的データ属性のカラム名だけでなく、値も利用してマッチングを行えば、精度を改善できるのではないかと考えた。ゆえに、本研究では、量的データ属性の値の部分を手掛かりとするマッチング手法であるインスタンスベースマッチングを研究の対象とする。以降では、Webからの表抽出、および、表の解釈に関する関連研究について述べる。

2.1 Webからの表抽出

Web情報の中から、高精度な表の抽出と表からの情報抽出を行う研究として、Gatterbauerらによる研究が挙げられる[5]。彼らの手法では、Web上のページから、表の格子などの外見的特徴、および、あらかじめ定めておいた表の定義を利用して表の情報を抽出し、その後ヒューリスティックによる解釈を行っている。実際にWeb上の表を対象として我々の手法を実行する場合には、この手法を利用できる。

また、Chenらは、スプレッドシートからユーザの労力を最小化しつつ正確な関係メタデータを抽出する二段階半自動システムを提案している[2]。彼らの提案したシステムでは、まず、スプレッドシートの図表形式に基づき関係メタデータを復元する。次に、その復元結果を利用して、別のスプレッドシート内における似たような領域を特定することで、ユーザ側の単一の修正で多くの抽出エラーをカバー可能にする。彼らの手法では、スプレッドシート形式の量的データを、関係モデルへと変換している。この手法を利用すれば、Web上のスプレッドシート形式のデータに対しても、同一性判定を実行できる。

2.2 表の解釈

Yakoutらの研究は、Webに存在するHTMLの表において、セルの値が空欄になっている表から、属性名やそのカラム内の値に対応する実体を利用して、正確に情報を収集することを目的とするものである[9]。論文の中では、この目的を達成するために、(1)表の属性名を利用した表の空欄部分の補完(2)表中で与えられた実体を利用した表の空欄部分の補完(3)表中の実体をもとにした関係する属性の発見という3つの操作を提案してい

る。である。表の属性、および、表のカラム内の値を考慮するという点でこのタスクは我々のタスクと類似する。我々のタスクで扱う量的データは、その属性値自体から特定される実体がほぼ全て「数字」になってしまうため、これらの研究で提案される手法はそのままでは利用できないと思われる。

Zhang らの研究は、HTML を利用した属性情報の自動抽出において、数量的な、そして時間依存な属性に対しても、正確に表の情報を抽出することであった [10]。よって、彼らは、(i) ユニット、スケール、そしてタイムスタンプと、カラムがラベルつけられているあるいは、(ii) 同じ数値的属性が異なるユニット内あるいはスケール内で表現されているような場合でも、属性間の意味的なマッチを実行するような意味的グラフを構築した。量的データの値を予測するという点でこのタスクは我々のタスクと類似する。この手法も、セルの値が空欄になっている表において欠損部分を補完するという形で、我々の手法の前処理として利用できると思われる。

Limaye らの研究では、ある表に対し、そこで言及されている可能性のあるエンティティ、属性内のセルをもとにしてエンティティを取り出すことになる表のカラムの型、そして表の属性のペアが表現する関係をもちいて、表のセルに対応するエンティティを割り当てる機械学習手法を提案している [6]。手法のアルゴリズムなどの詳細は 3.2 節で述べるが、表の各カラム名に対応する属性が何であるか判別可能にするという点で、このタスクは我々のタスクと類似する。彼らの手法では、主に表の質的データ属性を対象として、エンティティの割り当てを行っている。しかし、セルの内容が量的データの場合、前述の関連研究と同様に対応エンティティを直接的に見つけることは困難である。よって、彼らの手法を、量的データを含む表に対して実行したとしても、カラム名のみしか手がかりとして使えないため、精度が悪化してしまうと思われる。

3. 手 法

本節では、2 つの表とその表中の各 1 つの量的データ属性を入力とし、その 2 つの属性の同一性判定を行う問題について説明する。その後、既存手法の説明を行い、量的データに特化した同一性判定手法、および、そのアルゴリズムを述べる。

3.1 問題設定

ある表を T_i とする。また、 T_i は、属性集合 $A_i = \{a_{i1}, a_{i2}, \dots\}$ 、および、関係 $R_i \subseteq D_{i1} \times D_{i2} \times \dots$ により構成される。なお、 D_{ik} は属性 a_{ik} の定義域である。加えて、各属性 a_{ik} はその言語表現であるカラム名 c_{ik} によって表現される。さらに、表 T_i における属性 a_{ik} の値集合を $V_{ik} = \{t[a_{ik}] | t \in R_i\}$ (ただし、 $t[a_{ik}]$ はタプル t の属性 a_{ik} の値) と表現する。

我々の目的は、ある表 T_i の属性集合 A_i 中の量的データを表す属性 a_{ik} が別の表 T_j の量的データを表す属性 a_{jl} と同じであるか否かを判定することである。すなわち、 $a_{ik} = a_{jl}$ かどうかを判定する問題に取り組む。

3.2 既存の属性の同一性判定手法

前述のように、Limaye らは、表に対し、エンティティ、属性の型、関係を利用することによって、表のセルへの注釈付けを

試みている [6]。この注釈付けを行うことによって、属性の型が明らかとなり、属性の同一性を判定することが可能になる。なお、エンティティとは何らかの識別名によって指し示された、独立した存在として認識され、一意に識別されるものである。また、型とはエンティティの種類を表現するものであり、例として、国、首都、日本の都市、果実といったものがあげられる。手法の基本的なアルゴリズムを以下に示す。

(1) 属性にある型を割り当てる。

(2) 属性内の各値に 1 で割り当てられた型に属するエンティティを割り当てる。

(3) 1, 2 を属性に割り当てる型を変えて繰り返す。

(4) 属性名と型名の類似度が高く、なおかつ、その型に属するエンティティが多くの値に割り当てられた型を、最終的に属性へ割り当てる型とする。

以下、アルゴリズムの流れの詳細を説明する。

まず、変数 x に全ての型の集合 X の最初の要素を代入する。この x の値が型の推定を行いたい属性に現在割り当てられている型を表す。この x の値にもとづいて以下の処理を行う。

- カラム名と、型 x の名前間の類似度を計算する。

- 属性内の各セルに対して、セルの内容が指し示すエンティティの候補を挙げる。

候補の中から、セルと名前の類似度が高く、型 x に属するエンティティを選択する。同様にしてセルごとに対応するエンティティを決定していく。この割り当てを x の値、すなわち、カラムに対応する型を変えながら実行し、最終的に最もセル-エンティティとカラム名-型の類似度が高く、かつ、対応するエンティティの多くが割り当てられた型に属するような型を採用する。このように各カラムに型を割り当てていき、その型が一致する場合、2 つのカラムに対応する属性は同じであると判定する

3.3 提案手法

1 つ目の前提として、「同じ実体の同じ属性の量的データは誤差を許容して一致する」というものをおく。この前提より、「もし同じ実体の 2 つの属性の各量的データの値が誤差を許容して一致すれば、その 2 つの属性は同一である」という 1 つ目の仮定が導かれる。また、2 つ目の前提として、「量的データが他の量的データと高い相関を持つ（他の量的データに基づく予測値の精度が十分に高い）」というものをおく。これら 2 つの前提より、「もし同じ実体の 2 つの属性の各量的データの予測値と実測値が一致すれば、その 2 つの属性は同一である」という 2 つ目の仮定が導かれる。この 2 つの仮定が本論文で提案する手法の基本的なアイデアである。

図 1 では、提案手法の流れが具体例を用いて表されており、処理 1「2 つの表の間における実体が重複するタプルの割合を計算」、処理 2「同じ実体のタプルのうち誤差を許容して量的データの値が一致するものの割合を計算」、処理 3「別の表の情報で予測関数を学習」、処理 4「学習された関数による予測値と実測値との比較」という 4 つの処理が表現されている。

図 1 の処理 1 では、表 T_i のカラム名が「都道府県」である属性と、表 T_j のカラム名が「県名」である属性において、 T_i または T_j に含まれる県のうち T_i と T_j の両方に含まれる県の割合を

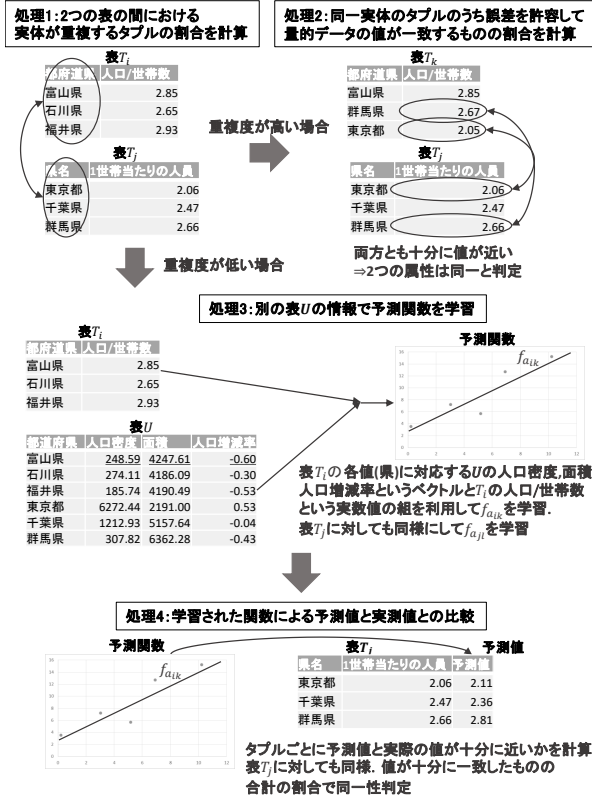


図1 量的データを表す属性の同一性判定手法

計算している. この割合が十分に大きい場合, それらのタブルにおける量的データの値の重複を利用する同一性判定の方が, 予測値を利用する同一性判定よりも精度が高くなることが期待されるため, 処理2を行う. そうでない場合は処理3を行う.

また, 図1の処理2では, 表 T_k と T_j の両方の表に含まれている県である「群馬県」, 「東京都」のタブルにおいて, 同じ県に対する量的データ属性「人口/世帯数」, および, 「1世帯当たりの人員」の値を比較し, 十分に値が近かったタブルのペアの割合に応じてこれら2つの属性の同一性を判定している. なお, ここでは処理1の割合計算を, T_k と T_j に対して行ったものとする. 「もし同じ実体の2つの属性の量的データの値が誤差を許容して一致すれば, その2つの属性は同一である」という仮定に基づき, 同じ実体の2つの属性の量的データの値が誤差を許容して一致するタブルの割合が十分に高ければ, その2つの属性は同一であると判定する. その場合, 両方の表に「群馬県」, 「東京都」が含まれているため, 処理1の計算結果の割合が高いものとなる.

図1の処理3では, 表 T_i と U の両方の表に含まれている「富山県」, 「石川県」, 「福井県」のタブルにおいて, 同じ県ごとに表 U の「都道府県」以外の量的データ属性「人口密度」, 「面積」, 「人口増減率」の値をまとめたベクトルと表 T_i の量的データ属性「人口/世帯数」の値を組にし, それらを訓練データとして与えることで予測関数を学習させている. 同様のことを表 T_j に対しても行う. そして, 図1の処理4では, 表 T_i と U によって学習された予測関数を用い, 表 T_j の県ごとに量的データ属性「人口/世帯数」の予測値と実際の値を比較し, 十分に値が近

Algorithm 1 CheckEquality(a_{ik} , a_{jl} , T_i , T_j)

Require: 属性 $a_{ik} \in A_i$, 属性 $a_{jl} \in A_j$, 表 T_i , 表 T_j

Ensure: a_{ik} と a_{jl} が同一か否かを表すブール値 S

```

1: CCK ← GetCommonCandidateKeys( $a_{ik}$ ,  $a_{jl}$ ,  $T_i$ ,  $T_j$ ) // CCK に  $T_i$  と  $T_j$  それぞれの候補キー集合の共通部分を代入
2:  $S \leftarrow \text{False}$  //  $S$  には  $a_{ik}$ ,  $a_{jl}$  が同一なら True を, 異なるなら False を最終的に代入
3: for  $K \in \text{CCK}$  do
4:    $D \leftarrow \{t \mid t \in T_i \wedge t' \in T_j \wedge t[K] = t'[K]\}$  //  $D$  に  $T_i$  と  $T_j$  の両方に含まれる候補キー  $K$  の値集合, すなわち, 実体集合を代入
5:   if  $|D[K]| / |T_i[K] \cup T_j[K]| > \alpha$  then // 実体の重複の割合を計算
6:      $S \leftarrow \text{CheckEqualityByDuplication}(a_{ik}, a_{jl}, T_i, T_j)$  // 重複するタブルの比較による判定
7:   else
8:      $S \leftarrow \text{CheckEqualityByRegression}(a_{ik}, a_{jl}, T_i, T_j)$  // 回帰直線による判定
9:   end if
10:  if  $S$  then
11:    break
12:  end if
13: end for

```

いタブルの合計の割合に応じてこれら2つの属性の同一性を判定する. 「もし同じ実体の2つの属性の量的データの予測値と実測値が一致すれば, その2つの属性は同一である」という仮定に基づき, 同じ実体の2つの属性の量的データの予測値と実測値が一致するタブルの割合が十分に高ければ, その2つの属性は同一であると判定する. 同様のことを表 T_i に対しても行う. その後, 予測値と実測値が十分に近かったタブルの割合を計算し, 属性 a_{ik} , a_{jl} の同一性を判定する.

3.4 提案アルゴリズム

量的データを表す属性の同一性判定のために提案する手法のアルゴリズムについて説明する.

提案手法の主なアルゴリズムは CheckEquality である. このアルゴリズムは, 引数として属性 $a_{ik} \in A_i$, 属性 $a_{jl} \in A_j$, 表 T_i , 表 T_j をとり, a_{ik} と a_{jl} が同一の属性であるか否かを返す. また, このアルゴリズムでは, 以下の処理を行う.

Algorithm 1-1 (Algorithm 1: Line 1)

T_i と T_j で同じ実体を表すタブルを見つけるために, それぞれの候補キー集合の共通部分 CCK を取得する関数 GetCommonCandidateKeys を実行する.

Algorithm 1-2 (Algorithm 1: Lines 3-5)

候補キー $K \in \text{CCK}$ について, T_i と T_j で同じ実体を表すタブルの割合を計算する.

Algorithm 1-3 (Algorithm 1: Lines 5-9)

Algorithm 1-2 の計算結果に応じて以下の処理を実行する.

- 両表において同じ実体を表すタブルの割合が十分に大きい場合は, それらのタブルにおける量的データの値の重複を利用する同一性判定の方が, 予測値を利用する同一性判定よりも精度が高くなることが期待される. そのため, 両表における同じ実体を表すタブルのうち量的データの値が一致する

Algorithm 2 GetCommonCandidateKeys(a_{ik}, a_{jl}, T_i, T_j)

Require: 属性 $a_{ik} \in A_i$, 属性 $a_{jl} \in A_j$, 表 T_i , 表 T_j

Ensure: T_i と T_j それぞれの候補キー集合の共通部分 CK

- 1: $QltA_i \leftarrow \text{GetQualityAttribute}(T_i)$
 - 2: $QltA_j \leftarrow \text{GetQualityAttribute}(T_j)$
 - 3: $\mathcal{K}_i \leftarrow \text{GetCandidateKey}(T_i, QltA_i)$
 - 4: $\mathcal{K}_j \leftarrow \text{GetCandidateKey}(T_j, QltA_j)$
 - 5: $CCK \leftarrow \mathcal{K}_i \cap \mathcal{K}_j$
-

ものの割合を計算することで、属性の同一性判定を行う関数 CheckEqualityByDuplication を実行する。

- そうでない場合は、量的データの値を予測する回帰モデルを学習し、その上で両表における同じ実体を表すタブルのうち量的データの値が予測値と実測値との間で一致するものの割合を計算することで、属性の同一性判定を行う関数 CheckEqualityByRegression を実行する。

Algorithm 1-4 (Algorithm 1: Lines 10-12)

Algorithm 1-3 の結果で、 $S = \text{True}$ となった場合は終了する。そうでない場合は、 K の値を変えて **Algorithm 1-2** へ戻る。 $\exists K \in CCK$ において、 $S = \text{False}$ となった場合は終了する。

関数 GetCommonCandidateKeys, CheckEqualityByDuplication, CheckEqualityByRegression の内容は後述で説明するとして、ここでは **Algorithm 1-2**, **Algorithm 1-3** を主に説明する。これらのアルゴリズムでは、同じ実体を表すタブルの割合を計算する。すなわち、上記で得られた CCK の要素 K について、表 T_i と表 T_j の属性集合 K の値が同じになるタブルを列挙し、その割合を求める。すなわち、

$$D = \{t \mid t \in T_i \wedge t' \in T_j \wedge t[K] = t'[K]\} \quad (1)$$

として、

$$|D[K]| / |T_i[K] \cup T_j[K]| > \alpha \quad (2)$$

であるかを調べる。ここで、 K は $K = \{k_1, k_2, \dots\}$ のような属性集合であるとする。このとき、タブル $t \in T_i$ に関して、 $t[K] = (t[k_1], t[k_2], \dots)$ と定義する。ただし、 $t[k]$ はタブル t の属性 k の値を表す。また、表 T に関して $T[K] = \{t[K] \mid t \in T_h\}$ と定義する。表 T_i と表 T_j の属性集合 K の値が同じになるタブルの割合が定めておいた閾値 α ($0 \leq \alpha \leq 1$) より大きいなら重複があると考え、CheckEqualityByDuplication を行う。そうでなければ、重複がないと考え、CheckEqualityByRegression を行う。以降、関数 GetCommonCandidateKeys, CheckEqualityByDuplication, CheckEqualityByRegression の内容を説明する。

3.4.1 GetCommonCandidateKeys

このアルゴリズムは、引数として属性 $a_{ik} \in A_i$, 属性 $a_{jl} \in A_j$, 表 T_i , 表 T_j をとり、 T_i と T_j の各候補キー集合を求め、その共通部分 CCK を返す。また、このアルゴリズムでは、以下の処理を行う。

Algorithm 2-1 (Algorithm 2: Lines 1-2)

T_i と T_j の各質的データ属性を取得する。なお、関数 GetQualityAttribute(T_i) は表 T_i を引数にとり、 T_i の質的デー

Algorithm 3 CheckEqualityByDuplication(a_{ik}, a_{jl}, T_i, T_j)

Require: 属性 $a_{ik} \in A_i$, 属性 $a_{jl} \in A_j$, 表 T_i , 表 T_j

Ensure: a_{ik} と a_{jl} が同一か否かを表すブール値 S

- 1: $\text{Tuple} \leftarrow 0$
 - 2: $\text{Match} \leftarrow 0$
 - 3: **for** $t \in T_i, t' \in T_j$ **do**
 - 4: **if** $t[K] = t'[K]$ **then**
 - 5: $\text{Tuple} \leftarrow \text{Tuple} + 1$
 - 6: **if** $\text{dif}(t[a_{ik}], t'[a_{jl}]) < \beta$ **then**
 - 7: $\text{Match} \leftarrow \text{Match} + 1$
 - 8: **end if**
 - 9: **end if**
 - 10: **end for**
 - 11: **if** $\text{Match}/\text{Tuple} > \gamma$ **then**
 - 12: $S \leftarrow \text{True}$
 - 13: **else**
 - 14: $S \leftarrow \text{False}$
 - 15: **end if**
-

タを含むカラムの属性集合 $QltA_i$ を見つけだし、その $QltA_i$ を返す。

Algorithm 2-2 (Algorithm 2: Lines 3-4)

Algorithm 2-1 で得られた質的データ属性の情報と、入力された表の情報から各表の候補キー集合を取得する。なお、関数 GetCandidateKey($T_i, QltA_i$) は表 T_i と質的データ属性の集合 $QltA_i$ を引数にとり、 T_i の(質的データを含む)属性集合から候補キーを検索し、その候補キー集合を返す。

Algorithm 2-3 (Algorithm 2: Line 5)

Algorithm 2-2 で得られた各表の候補キー集合の共通部分を求め、CCK に代入し終了する。

Algorithm 2-1 では、量的データ属性を含む各表に対し、どのデータが質的データ属性かを判定する。

Algorithm 2-2 では、質的データの組み合わせごとに同一性を判定する。すなわち、表 T_i, T_j について質的データの属性集合をそれぞれ $QltA_i, QltA_j$ とすると、 $\forall q_{ik} \in QltA_i, \forall q_{jl} \in QltA_j$ について、既存の手法で $q_{ik} = q_{jl}$ か否かを調べる。例えば、Limaye らの手法を用いることができる。これにより、表間においてどの質的データ属性のペアが同一であるか判明する。

Algorithm 2-3 では、両表からそれぞれ候補キーを列挙し、その共通部分を求める。ここで、 $t, t' \in T$ において $t[K] = t'[K] \Rightarrow t = t'$ となる属性集合 K を T のスーパーキーといい、 $\forall K' \subset K$ がスーパーキーでない K を候補キーであるという。表 T_i, T_j について、それらの候補キー集合を見つけ、それぞれを $\mathcal{K}_i, \mathcal{K}_j$ とする。そして、それらの共通部分集合を求め、それを $CCK = \mathcal{K}_i \cap \mathcal{K}_j$ とする。この処理を行うことによって、同じ実体を見つけるために、2つの表における量的データに対応している同一の実体を見つけることができる。

3.4.2 CheckEqualityByDuplication

このアルゴリズムは、引数として属性 $a_{ik} \in A_i$, 属性 $a_{jl} \in A_j$, 表 T_i , 表 T_j をとり、両表において候補キーの値が一致するタブルの割合を計算した結果、 a_{ik} と a_{jl} が同一か否かを返す。ま

Algorithm 4 CheckEqualityByRegression(a_{ik}, a_{jl}, T_i, T_j)

Require: 属性 $a_{ik} \in A_i$, 属性 $a_{jl} \in A_j$, 表 T_i , 表 T_j

Ensure: a_{ik} と a_{jl} が同一か否かを表すブール値 S

```
1: Tuplei ← TrainAndCalculate( $a_{ik}, a_{jl}, T_i, T_j, U$ )
2: Tuplej ← TrainAndCalculate( $a_{jl}, a_{ik}, T_j, T_i, U$ )
3: if (Tuplei + Tuplej) / ((|Ti|) + |Tj|) >  $\varepsilon$  then
4:   S ← True
5: else
6:   S ← False
7: end if
```

た, このアルゴリズムでは, 以下の処理を行う.

Algorithm 3-1 (Algorithm 3: Lines 3-4)

T_i と T_j 両表における同じ実体を表すタブルのペアを探す.

Algorithm 3-2 (Algorithm 3: Lines 4-10)

Algorithm 3-1 で見つけたペアに対し, 各タブルの量的データ属性 a_{ik} , および, a_{jl} の値が誤差を許容して一致するかを判定する. すなわち, $t[K] = t'[K]$ となる $\forall t \in T_i, t' \in T_j$ について,

$$\text{dif}(t[a_{ik}], t'[a_{jl}]) < \beta \quad (3)$$

かを判定する. $t[a_{ik}]$ と $t'[a_{jl}]$ の誤差を表す関数 $\text{dif}(t[a_{ik}], t'[a_{jl}])$ の値が β ($0 \leq \beta \leq 1$) よりも小さければ, $t[a_{ik}]$ と $t'[a_{jl}]$ の値は十分に近いとする. なお, 関数 $\text{dif}(x, y)$ は, $\text{dif}(x, y) = (|x| - |y|) / (|x| + |y|)$ と定義する.

Algorithm 3-3 (Algorithm 3: Line 11)

Algorithm 3-2 で値が一致すると判定されたタブルのペアが, 同じ実体を表すタブルのペア全体に占める割合を計算する.

Algorithm 3-4 (Algorithm 3: Lines 11-15)

Algorithm 3-3 で計算された割合が γ ($0 \leq \gamma \leq 1$) よりも大きければ, $S = \text{True}$ として終了する. そうでない場合は, $S = \text{False}$ として終了する.

3.4.3 CheckEqualityByRegression

このアルゴリズムは, 引数として属性 $a_{ik} \in A_i$, 属性 $a_{jl} \in A_j$, 表 T_i , 表 T_j をとり, 両表において予測値と実測値が一致するタブルの割合を計算した結果, a_{ik} と a_{jl} が同一か否かを返す. また, このアルゴリズムでは, 以下の処理を行う.

Algorithm 4-1 (Algorithm 4: Lines 1-2)

T_i と T_j のそれぞれに対し, 別の表 U を用いて量的データ属性の値を予測する関数を学習させ, その予測関数による予測値と実測値が誤差を許容して一致したタブルの数を返す関数 TrainAndCalculate を実行する

Algorithm 4-2 (Algorithm 4: Line 3)

Algorithm 4-1 で値が一致すると判定されたタブルの合計が, T_i と T_j のタブルの合計に占める割合を計算する.

Algorithm 4-3 (Algorithm 4: Lines 3-7)

Algorithm 4-2 で計算された割合が十分に大きければ, $S = \text{True}$ として終了する. そうでない場合は, $S = \text{False}$ として終了する. という処理を行う.

まず, 関数 TrainAndCalculate の内容を説明する. このアルゴリズムは, 引数として属性 $a_{ik} \in A_i$, 属性 $a_{jl} \in A_j$, 表 T_i , 表 T_j , 別の表 U をとり, T_j のタブルのうち, U の量的データ属性

Algorithm 5 TrainAndCalculate($a_{ik}, a_{jl}, T_i, T_j, U$)

Require: 属性 $a_{ik} \in A_i$, 属性 $a_{jl} \in A_j$, 表 T_i , 表 T_j , 別の表 U

Ensure: a_{jl} の予測値と実測値の値が十分に近いタブルの数 Match

```
1: Trdi ← {(V, t[aik]) | u ∈ U ∧ V = (u[q1], u[q2], ..., u[qm]) ∧ t ∈ Ti ∧ u[K] = t[K]}
2: faik ← Train(Trdi)
3: Match ← 0
4: for t ∈ Tj do
5:   for u ∈ U ∧ u[K] = t[K] do
6:     Qt ← (u[q1], u[q2], ..., u[qm])
7:     if dif(faik(Qt), t[ajl]) <  $\delta$  then
8:       Match ← Match + 1
9:     end if
10:   end for
11: end for
```

の値ベクトル集合と T_i の a_{ik} の値集合によって学習された a_{jl} の予測関数による予測値と実測値の値が十分に近いタブルの数 Match を返す. また, このアルゴリズムでは, 以下の処理を行う.

Algorithm 5-1 (Algorithm 5: Line 1)

U における量的データ属性の値ベクトル集合と, T_i における a_{ik} の値集合を組にして訓練用データを作成する.

Algorithm 5-2 (Algorithm 5: Line 2)

Algorithm 5-1 で作成された訓練用データを用い, a_{jl} の予測関数を学習する. なお, 関数 FunctionMaker(Trd_i) は訓練用のデータ Trd_i を引数にとり, 予測関数を学習し, その関数を返す.

Algorithm 5-3 (Algorithm 5: Lines 4-7)

Algorithm 5-2 で学習された関数と, U における量的データ属性の値を用いて, T_j の各タブルにおける a_{jl} の値を予測する.

Algorithm 5-4 (Algorithm 5: Lines 8-13)

T_j のタブルのうち, **Algorithm 5-3** で予測された値と. 実際の値が誤差を許容して一致したタブルの数を Match に代入する.

処理を詳しく説明すると, K を属性として持つ別の表 U の属性のうち量的データのものを $q_1, q_2, \dots, q_m$ とし, この量的データの属性データと $t[a_{ik}] (t \in T_i)$ を用いて, 関数 $f_{a_{ik}}$ を学習する. 学習のために与えるデータは

$$\{(V, t[a_{ik}]) | u \in U \wedge V = (u[q_1], u[q_2], \dots, u[q_m]) \wedge t \in T_i \wedge u[K] = t[K]\} \quad (4)$$

となる. なお, 関数 $f_{a_{ik}}$ はある実体の量的データを表す属性の値を, 別の量的データを表す属性の値を利用して予測する関数であり, $f_{a_{ik}}: \mathbb{R}^m \rightarrow \mathbb{R}$ である. その後, $f_{a_{ik}}$ を用いて, 表 T_j の量的データの属性 a_{jl} における, その予測値と実際の値との比較を行う. すなわち, $t \in T_j$ について,

$$\text{dif}(f_{a_{ik}}(Q_t), t[a_{jl}]) < \delta \quad (5)$$

となるタブルの数を計算する. なお, u は表 U に含まれるタブル, Q は表 U に含まれる量的データを表す属性の値ベクトルを表す. すなわち, $Q_t = (u[q_1], u[q_2], \dots, u[q_m])$ である. ただし, $u \in U \wedge u[K] = t[K]$ とする. $f_{a_{ik}}(Q_t)$ と $t[a_{jl}]$ の誤差を表す

関数 $\text{dif}(f_{a_{ik}}(Q_t), t[a_{jl}])$ の値が δ ($0 \leq \delta \leq 1$) よりも小さければ, $f_{a_{ik}}(Q_t)$ と $t[a_{jl}]$ の値は十分に近いとする.

次に, CheckEqualityByRegression の **Algorithm 4-2, Algorithm 4-3** の部分を説明する. ここでは, 関数 TrainAndCalculate によって, T_i, T_j それぞれで求められた, 予測値と実測値が誤差を許容して一致するタプルの合計が, T_i, T_j のタプルの合計に占める割合を計算し, 割合が ε ($0 \leq \varepsilon \leq 1$) より大きければ, $a_{ik} = a_{jl}$ と判定する.

4. 実験

まずデータセットの作成の概略について説明する. その後ベースライン手法を含む実験設定について紹介し最後に実験結果を示す.

4.1 データセット

Web や雑誌から自動で, 2 クラスのエンティティに対する様々なエンティティを抽出したものを加工した結果をデータセットとして扱う. これら 2 クラスは国, 都道府県である. このようにして得られたエンティティの集合を含む表から, 属性「国名」, および, 「都道府県名」と特定の量的データの属性の値を組にして取り出す. なお, 量的データの属性として国の場合は「1 人あたりの GDP」, 「一人当たりの医療費」, 「人口 10 万人あたりの自殺者数」, 「消費電力量」, 「穀物生産量」, 都道府県の場合は「人口/世帯数」, 「出生率」, 「女性平均寿命」, 「平均給与」, 「耕地面積」を用いる. 得られた各表のタプルをランダムに分割し, 5 つに分ける. ここで, この分割された表全体の集合を O とする. O の要素のうち, 表 T_i を 5 つに分割した表を含む集合を O_i とする. さらに, O_i の要素を O_{ik} とする. 次に, $O \setminus O_i$ の要素からランダムに 4 つの表を含む集合を E_i とする. ここで, $O_i \cup E_i$ をブロック i と呼ぶ. この結果, 各属性で 1 ブロック, 計 5 ブロックできる. ブロック i には分割された表が計 9 個含まれ, そのうち O_i に含まれる表は同じ表由来のものとなる (当然属性は同じ). このブロック 5 つを実験のためのデータセットとして用いる.

4.2 実験設定

上述のデータセット 5 ブロックのうち, 1 ブロックを選択する. これをテストデータセットとする. そして, 選択されなかった 4 ブロックを訓練データセットとしてパラメータ $\alpha, \beta, \gamma, \delta, \varepsilon$ を決める. まず, 各訓練データセットに対し, O_i に含まれる表 O_{ik} を 1 つ選んだ後, O_i に含まれる O_{ik} 以外の表を同一性判定の対象として, 2 つの表における質的データ属性の値の重複度に関わらず, 重複していると判定された各タプルの量的データ属性の値の類似度, および, 類似しているタプルの割合を計算するアルゴリズムを実行する. なお, 重複度とは, 同じ実体をあらわすタプルの割合のことである. この際, パラメータ β, γ の値を 0.2 刻みで動かし, 各々の値の組み合わせにおける精度を各訓練データセットで計 25 通り計算しておく. これを O_i のそれぞれの分割された表に対して行い, それぞれで求めた精度を平均する. この値を ac とする. そして, 求められた ac の平均が最も大きくなる値の組み合わせをパラメータ β, γ の値として採用する. 次に各訓練データセットに対し,

O_i に含まれる分割された表を 1 つ選び, 自身以外の分割された表を同一性判定の対象として, 2 つの表における質的データ属性の値の重複度に関わらず, 予測関数による予測値と実際の値との類似度, および, 類似しているタプルの割合を計算するアルゴリズムを実行する. なお, 予測関数を学習させるためのデータは, scikit-learn^(注5) の MinMaxScaler によって, 最大値が 1, 最小値が 0 になるように属性値を正規化したものを与える. また, 予測関数の学習には, scikit-learn の SVR (サポートベクタマシン) を利用する. この際, パラメータ δ, ε の値を 0.2 刻みで動かし, 各々の値の組み合わせにおける精度を各訓練データセットで計 25 通り計算しておく. これを同様に O_i のそれぞれの分割された表に対して行い, ac の平均が最も大きくなる値の組み合わせをパラメータ δ, ε の値として採用する. その後, 各訓練データセットに対し, O_i に含まれる分割された表を 1 つ選び, 自身以外の分割された表を比較対象として, 手法のアルゴリズムをそのまま実行する. この際, パラメータ α の値を 0.2 刻みで動かし, 各々の値の組み合わせにおける精度を各訓練データセットの計 5 通り計算しておく. これも O_i のそれぞれの分割された表に対して行い, ac の平均が最も大きくなる値をパラメータ α の値として採用する. このパラメータの値を用い, テストデータに対して, O_i に含まれる分割された表を 1 つ選び, 自身以外の分割された表を比較対象として, 手法のアルゴリズムを実行し, ac , および, ac の平均を求める. この ac の平均を $avgac$ とする. これをテストデータを変えて計 5 通り繰り返し, 求められた $avgac$ の平均を最終的な精度とする. これが実験の手順である.

ここで, 本研究の提案手法を手法 Reg とする. また, 実験においてベースラインとなる他の手法を, 手法 Rand から順に説明する. これは, 二つの属性が同一であるか否かをランダムで決定する手法である. 次に, 手法 Dup について説明する. これは, 予測を用いず, 表における同じ実体を表すタプルのうち量的データの値が一致するものの割合を計算する手法である. すなわち, **Algorithm 1-3** にて, 同じ実体を表すタプルの割合による分岐の際に必ず, CheckEqualityByDuplication を実行する手法である. なお, CheckEqualityByDuplication における値の一致の判定の際は誤差を許容するものとする. 続いて, 手法 PDup について説明する. これも, Dup と同様に, 表における同じ実体を表すタプルのうち量的データの値が一致するものの割合のみを計算する手法であるが, CheckEqualityByDuplication における値の一致の判定の際は誤差を許容せず, 完全一致のみをものとする. そして, 手法 Avg について説明する. これは, 関数学習結果を, 各表における量的データ属性の値の平均値を返す関数としてあらかじめ与え, その上で表における同じ実体を表すタプルのうち量的データの値が予測値と実測値との間で一致するものの割合を計算する手法である. すなわち, **Algorithm 5-2** にて, 予測関数の学習結果を必ず $f_{a_{ik}}(Q_t) = (\sum_{t' \in T_i} t'[a_{ik}]) / |T_i|$ とする手法である. なお, この手法をベースライン手法とする理由として, 訓練データとして与える他の表の属性値集合が予

(注5) : <http://scikit-learn.org/>

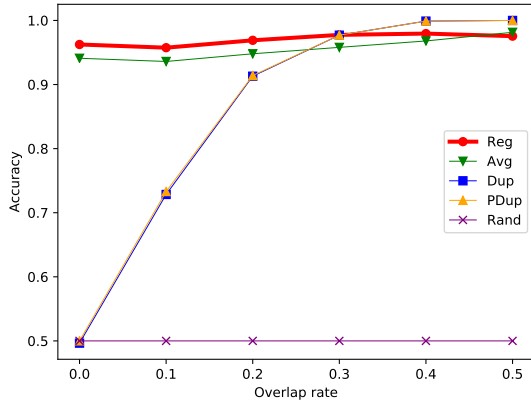


図2 各手法の精度

測する属性値と何の相関もなかった場合、関数学習結果である最も分散が小さくなる関数は、量的データ属性の値の平均値を与える関数となるためである。これらの手法の結果を用いて提案手法の有効性を示す。

4.3 実験結果

図2で実験の結果得られた各手法の精度を示す。横軸は2つの表の間の同じ実体をあらわすタブルの割合、縦軸は最終的な精度となる値 $avgac$ を表す。また、各線が異なる手法の精度を表している。

まず、タブルの重複度によらず、ランダムで2つの属性が同一か否かを判定する Rand では、実測値ではなく、理論値として全ての場合において、0.5 という精度を扱う。次に、実体が重複するタブル間の値の一致を利用する Dup では、重複が全くない場合の精度は Rand と同等で非常に低いが、重複度の上昇につれ精度は直線的に上昇し、重複度が0.4付近でほぼ飽和する。値が同じであるか否かを判定する際に誤差を許容しない PDup においても、ほとんど Dup と同様の挙動を示す。Avg も比較的高い精度を記録したが、ほとんどの場合において、Reg の精度を下回っている。そして、本論文の提案手法である重複タブル間の値の比較と、予測値と実測値の比較の両方を利用する Reg では、全ての状況において、安定して高い精度を維持している。特に、他の手法では Avg を除いて非常に低い精度となる重複度0.0の場合でも、Reg では0.95を超える精度を記録した。ゆえに、Reg の値の予測に基づく同一性判定は、重複度が低い場合において、十分に有効であることが分かる。

一方で、ある程度重複度が高くなっても手法 Reg の精度はあまり上昇しない。図2において、Reg の精度は重複度が0.3のあたりで Dup, PDup とほぼ同等になり、重複度が0.4以降は Dup, PDup の精度を多少下回ってしまっている。今回の実験結果を鑑みるに、Dup では全ての場合において必ず、両表において候補キーの値が一致するタブルの割合を計算する判定方法を選択するが、Reg ではあくまで重複度の大小に基づく分岐が行われるため、必然的に重複度が低くなる $t \in O$ と $t' \in E$ を判定する場合に、両表において予測値と実測値が一致するタブルの割合を計算する判定方法が選択され、そこで判定誤りが積み重なったためと思われる。

このように、提案手法 Reg は、同一の実体を表すタブルの割合が高い場合において、重複度のみを考える手法を多少下回る点はあるものの、同一の実体を表すタブルの割合が低い場合において、他の手法を非常に大きく上回る精度を記録したため、本論文の主旨である「属性の値が量的データを表す属性の同一性判定」に対する有効性は十分に示されたと言える。

5. まとめ

入力として2つの表とその表中の各1つの量的データ属性が与えられた場合に、その2つの属性の同一性を判定する問題に本論文で取り組んだ。提案手法において、異なる表において同一の実体を含むタブルを特定し比較する点、値の重複を利用した同一性判定の際に数値の誤差を許容する点、そして入力された2つの表以外の量的データを含む表の情報に基づく予測値を扱う点が既存手法と異なる特長である。実験では、Web から得られた量的データを含む表をランダムに分割し、分割された2つの表のある属性ペアが同一の属性か否かを判定した。実験の結果、提案手法がタブルの重複度が低い場合において、ベースラインとなる他の手法を精度の面で大きく上回ることが示された。ゆえに、提案手法の特長である値の予測は量的データを表す属性に対する同一性判定において、効果的であることが示されたと言える。今後の課題としては、同一の実体を表すタブルの割合が低い場合でも、他のベースライン手法を上回る精度を実現できるように提案手法 Reg を改良することが挙げられる。

謝辞 本研究は JSPS 科研費 JP25240050, JP15H01718, JP26700009 の助成を受けたものです。ここに記して謝意を表します。

文 献

- [1] P. A. Bernstein, J. Madhavan, and E. Rahm. Generic schema matching, ten years later. *VLDB*, 4(11):695–701, 2011.
- [2] Z. Chen and M. Cafarella. Integrating spreadsheet data via accurate and low-effort extraction. In *SIGKDD*, pages 1126–1135. ACM, 2014.
- [3] A. Doan, P. M. Domingos, and A. Y. Levy. Learning source description for data integration. In *WebDB*, pages 81–86, 2000.
- [4] A. Doan, J. Madhavan, R. Dhamankar, P. Domingos, and A. Halevy. Learning to match ontologies on the semantic web. *VLDB*, 12(4):303–319, 2003.
- [5] W. Gatterbauer, P. Bohunsky, M. Herzog, B. Krüpl, and B. Pollak. Towards domain-independent information extraction from web tables. In *WWW*, pages 71–80. ACM, 2007.
- [6] G. Limaye, S. Sarawagi, and S. Chakrabarti. Annotating and searching web tables using entities, types and relationships. *VLDB*, 3(1-2):1338–1347, 2010.
- [7] E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. *VLDB*, 10(4):334–350, 2001.
- [8] P. Shvaiko and J. Euzenat. A survey of schema-based matching approaches. *Journal on data semantics IV*, pages 146–171, 2005.
- [9] M. Yakout, K. Ganjam, K. Chakrabarti, and S. Chaudhuri. Infogather: entity augmentation and attribute discovery by holistic matching with web tables. In *SIGMOD*, pages 97–108, 2012.
- [10] M. Zhang and K. Chakrabarti. Infogather+: semantic matching and annotation of numeric and time-varying attributes in web tables. In *SIGMOD*, pages 145–156, 2013.
- [11] Z. Zhang. Towards efficient and effective semantic table interpretation. In *ISWC*, pages 487–502. Springer, 2014.