データベースのスキーマ情報を活用した機械学習

志村　薫† 杉浦　健人† 石川　佳治†

† 名古屋大学大学院情報学研究科　〒 464–8603 愛知県名古屋市千種区不老町
E-mail: {shimura,sugiura}@db.is.i.nagoya-u.ac.jp, ishikawa@i.nagoya-u.ac.jp

あらまし　ビッグデータ時代の今日，企業などのデータベースには膨大なデータが蓄積されている．これらデータベース中のデータから知見を獲得しビジネスなどに活用するために，データベース中のデータを用いた機械学習の需要が高まっている．しかし，一般にデータベース中のデータは正規化もしくはデータウェアハウスに基づく解析処理のために複数のテーブルに分割されており，それらを学習用データとして利用するために一つのテーブルへ結合するには高いコストがかかる．そこで，本研究では結合の削減を目的とし，キー制約や従属性などのスキーマ情報を活用した特徴選択による結合削減手法を提案する．本手法では，分割されたデータを一つに結合する前に，スキーマ情報を活用して相互情報量を求めることで学習に不要な特徴量を判定し，結合を削減する．

キーワード　データベース技術，機械学習，メタデータ

1 序　　論

1.1 背　　景

ビッグデータ時代の今日，企業などのデータベースには膨大なデータが蓄積されている．これらデータベース中のデータから知見を獲得しビジネスなどに活用するために，データベース中のデータを用いた機械学習の需要が高まっている．

しかし，一般にデータベース中のデータは正規化もしくはデータウェアハウスに基づく解析処理のために複数のテーブルに分割されており，それらを学習用データとして利用するために一つのテーブルへ結合するには高いコストがかかる．また，高いコストをかけ結合したとしても，特徴選択の結果次第では結合した属性が学習に利用されないこともあるため，結合処理そのものが無駄となることもある．例えば，図 1 は履修のデータセットであり，(a) 履修テーブル，(b) 学生テーブル，(c) 科目テーブル，(d) 教員テーブルの 4 つに分割されている．このデータセットを用いて成績の合否を機械学習で予測するためには，最初に (履修テーブル．学籍番号 = 学生テーブル．学籍番号 AND 履修テーブル.科目番号 = 科目テーブル.科目番号 AND 科目テーブル.教員番号 = 教員テーブル.教員番号) という条件で結合し，一つの巨大なテーブルにする必要がある．しかし，その後の特徴選択で必ずしもすべてのテーブルから属性が選択されるわけではなく，教員テーブルから 1 つの属性も選択されないというような場合も考えられる．この場合，結果的に教員テーブルの結合は不要であったことになり，結合処理にかかったコストが無駄となる．

1.2 目　　的

本研究では結合の削減を目的とし，データベースのキー制約や従属性などのスキーマ情報を活用した特徴選択による結合削減手法を提案する．本手法では，分割されたデータを一つに結合する前に，スキーマ情報を活用して相互情報量を求めることで学習に不要な特徴量を判定し，結合を削減する．図 1 の例では，(a) 履修テーブル，(b) 学生テーブル，(c) 科目テーブル，(d) 教員テーブルの 4 つに分割されている段階で特徴選択を行うことで，教員テーブルの属性が学習に利用されないような場合は，その結合を削減することが可能となる．本研究での提案手法のイメージを図 2 に示す．学習結果の品質に大きな影響を与えずに処理コストを大幅に削減することが目標となる．

1.3 構　　成

本論文の構成は以下のとおりである．まず，2 章ではデータベースのキー制約や従属性などのスキーマ情報の活用に関する関連研究を紹介する．次に，3 章では本論文で提案するスキーマ情報を活用した特徴選択による結合削減手法について説明する．そして，4 章では実験により提案手法の妥当性を評価する．最後に，5 章では本論文のまとめと今後の課題について述べる．

2 関 連 研 究

機械学習の対象となるデータのうち，本研究ではデータベースに格納されたデータに着目する．ビッグデータ時代の今日では，データベース中のデータも膨大となっており，高品質かつ効率的な機械学習を実現することが求められる [1]．特に効率の面に着目した場合，データベース中のデータをすべてファイルに出力し，そのファイルを機械学習のソフトウェアの入力とするアプローチは処理コストが大きい．このため，データベースに格納されているデータに関する統計情報の活用や，データベースの問合せ処理技術の活用などが重要となる．本章では，提案手法の関連研究としてタプル比を基準とした結合削減および確率的条件付き関数従属性について述べる．

2.1 タプル比を基準とした結合削減

Kumar らはタプル比 (タプル数の比) を基準とした結合削減手法を提案した [2, 3]．この手法ではスタースキーマに従ったテーブル群を前提としており，ディメンションテーブルに対す

(a) 履修テーブル

履修番号	成績	学籍番号	科目番号
#1	合格	#1	#1
#2	不合格	#2	#1
#3	合格	#3	#3
-	-	-	-
#1000000	合格	#10000	#1000

(b) 学生テーブル

学籍番号	氏名	学部	学年
#1	吉田○○	情報	1
#2	伊藤○○	情報	2
#3	山本○○	工	1
-	-	-	-
#10000	加藤○○	経済	4

(c) 科目テーブル

科目番号	科目名	単位数	曜日	時限	教員番号
#1	線形代数	2	月	1	#1
#2	人工知能	2	月	2	#2
#3	確率統計	2	火	4	#2
-	-	-	-	-	-
#1000	英語	3	金	1	#700

(d) 教員テーブル

教員番号	氏名	勤続年数
#1	鈴木○○	5
#2	高橋○○	9
#3	山田○○	15
-	-	-
#700	佐藤○○	5

図 1　履修のデータセット

るファクトテーブルのタプル比を用いて，結合を削減した場合の学習モデルの汎化性能の低下具合を見積もる．タプル比がしきい値以上の場合は結合の削減が可能であると判断し，ファクトテーブルの外部キーを対応するディメンションテーブルの全属性の代わりに学習の特徴量とすることで，結合を削減する．

例えば，スタースキーマに従い，図 1(a) 履修テーブル，(b) 学生テーブル，(c) 科目テーブル (教員番号を除く) を用いて成績の合否を機械学習で予測することを考える．従来では，(a), (b), (c) すべてのテーブルを結合し，(氏名，学部，学年，科目名，単位数，曜日，時限) を特徴量として用いる．一方，Kumar らの手法では仮にしきい値を 500 とすると，学生テーブルのタプル比は $\frac{1000000}{10000} = 100 < 500$，科目テーブルのタプル比は $\frac{1000000}{1000} = 1000 > 500$ であるため，(氏名，学部，学年，科目番号) を特徴量とし，科目テーブルの結合を削減する．

Kumar らの手法は，データベース中のデータを用いた機械学習における結合削減手法の先駆的な事例となっているが，そのアプローチには問題が残っている．一つには，理論的導出に一部誤りが含まれており，最終的に得られたタプル比に関する規則の妥当性が明確でないことである．また，データベースで利用可能な種々の統計情報などを十分活用できていないという点も問題である．

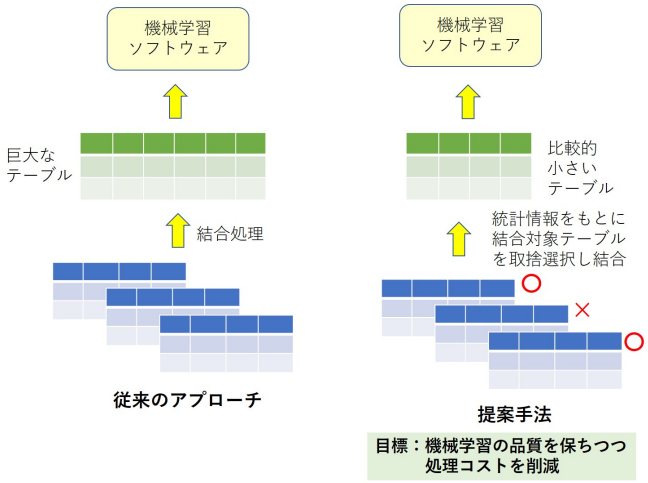

図 2　提案手法のイメージ

2.2　確率的条件付き関数従属性

近年，関数従属性を確率的に拡張した確率的条件付き関数従属性が着目されている [4–8]．そもそも，関数従属性とは，ある属性 X の値が決まると別の属性 Y の値が一意に定まる性質のことであり，$X \to Y$ のように表記する．関数従属性は属性間の関係を表す有用な情報であるが，一方でデータベース中の全タプルが従属性の条件を満たさなければならないという厳しい条件を持ち，一部の例外的なタプルによって関数従属性が成り立たないケースも起こりうる．そこで，確率的条件付き関数従属性では，条件ごとに従属性が成り立つ割合を確率で表すことで，大部分のタプルに共通する属性間の関係を利用できるようにする．例えば，図 1 において，曜日が「月」ならば 9 割の確率で成績が「合格」であるとする．この場合，関数従属性を $\langle$ 曜日 $=$ 月 $\to$ 成績 $=$ 合格 : 0.9 $\rangle$ のように確率的に表現する．このように，確率的条件付き関数従属性をデータベースのメタデータとして管理することで，属性間の関係を表す指標として活用できる．

3　スキーマ情報を活用した特徴選択

本章では，キー制約や従属性などのスキーマ情報を活用した特徴選択による結合削減手法について説明する．本手法では，分割されたデータを一つに結合する前に，スキーマ情報を活用して相互情報量を求めることで学習に不要な特徴量を判定し，結合を削減する．

3.1 準　　備

本節では，提案手法の基本方針を説明する上で必要となる前提条件や予備知識について述べる．

3.1.1 対象とするスキーマ

本手法で対象とするスキーマは木構造のスキーマとする．木構造のスキーマとは，テーブル間の参照関係について，参照元のテーブルを親，参照先のテーブルを子と見なすと，参照関係が木構造となるスキーマのことである．例えば，図 1 のテーブル群では，(a) 履修テーブルが (b) 学生テーブルおよび (c) 科目テーブルを参照し，(c) 科目テーブルが (d) 教員テーブルを参照している．ここで，(a) を (b)，(c) の親，(c) を (d) の親と見なすと参照関係が木構造となるため，図 1 のテーブル群は木構造のスキーマに従っているといえる．

3.1.2 特徴選択の評価指標

本手法で用いる特徴選択の評価指標は mRMR(minimal redundancy maximal relevance) とする．mRMR の定義は，

$$mRMR = \frac{1}{|S|}\sum_{x_i \in S} I(x_i; C) - \frac{1}{|S|^2}\sum_{x_i \in S}\sum_{x_j \in S} I(x_i; x_j) \quad (1)$$

である [9,10]．なお，S は特徴量の全体集合，x は S に含まれる特徴量，C はクラスラベル，$I(\cdot)$ は相互情報量を表す．mRMR の第一項は目的変数と説明変数の相互情報量であるため，大きいほど目的変数と説明変数の相関が強いことを意味する．また，第二項は説明変数間の相互情報量であるため，小さいほど説明変数間の冗長性が低いことを意味する．したがって，mRMR 全体が大きくなるよう特徴選択することで，目的変数との相関が強く，冗長性の低い学習に適した説明変数を特徴量として得ることができる．

3.1.3 クロス集計表からの相互情報量の導出

式 (1) から明らかなように，mRMR を求めるためには任意の二属性間の相互情報量が必要となることを踏まえ，本項では相互情報量がクロス集計表から導出可能であることを示す．

まず，相互情報量の定義は

$$I(X;Y) = \sum_{x \in X}\sum_{y \in Y} P_{X,Y}(x,y) \log \frac{P_{X,Y}(x,y)}{P_X(x)P_Y(y)} \quad (2)$$

であり，確率変数 X, Y の同時確率および周辺確率によって構成されている．そして，これら確率変数 X, Y の同時確率および周辺確率はクロス集計表から容易に導出可能である．

例えば，確率変数 X, Y に曜日，成績という属性を対応させた場合を考えると，クロス集計表は図 3 のようになる．なお，クロス集計表は一般には図 3(a) のような形式を取るが，データベースの観点からは図 3(b) のようなビューの形式で表現できる．ここで，フィールド中の $N_{\text{月, 合格}}$ とは曜日が「月」かつ成績が「合格」となるタプル数のことであり，他のフィールドについても同様に定義する．また，クロス集計表を周辺化した値についても定義しておく．まず，行の周辺化について，$N_{\text{月}}$ を曜日が「月」である行に関して周辺化した値，つまり，

$$N_{\text{月}} = N_{\text{月, 合格}} + N_{\text{月, 不合格}} \quad (3)$$

と定義し，他の行についても同様に定義する．続いて，列の周辺化について，$N_{\text{合格}}$ を成績が「合格」である列に関して周辺化した値，つまり，

$$N_{\text{合格}} = N_{\text{月, 合格}} + N_{\text{火, 合格}} + N_{\text{水, 合格}} + N_{\text{木, 合格}} + N_{\text{金, 合格}} \quad (4)$$

と定義し，他の列についても同様に定義する．最後に，N をすべてのフィールドの総和，つまり

$$\begin{aligned} N = {} & N_{\text{月, 合格}} + N_{\text{月, 不合格}} \\ & + N_{\text{火, 合格}} + N_{\text{火, 不合格}} \\ & + N_{\text{水, 合格}} + N_{\text{水, 不合格}} \\ & + N_{\text{木, 合格}} + N_{\text{木, 不合格}} \\ & + N_{\text{金, 合格}} + N_{\text{金, 不合格}} \end{aligned} \quad (5)$$

と定義する．これらの定義を用いると，曜日と成績の同時確率や周辺確率は

$$P_{\text{曜日, 成績}}(\text{月}, \text{合格}) = \frac{N_{\text{月, 合格}}}{N} \quad (6)$$

$$P_{\text{曜日}}(\text{月}) = \frac{N_{\text{月}}}{N} \quad (7)$$

$$P_{\text{成績}}(\text{合格}) = \frac{N_{\text{合格}}}{N} \quad (8)$$

のように表すことができる．ここで，曜日が「月」，成績が「合格」に対応する相互情報量の項について考えると，

$$\begin{aligned} & P_{\text{曜日, 成績}}(\text{月}, \text{合格}) \log \frac{P_{\text{曜日, 成績}}(\text{月}, \text{合格})}{P_{\text{曜日}}(\text{月})P_{\text{成績}}(\text{合格})} \\ = {} & \frac{N_{\text{月, 合格}}}{N} \log \frac{\frac{N_{\text{月, 合格}}}{N}}{\frac{N_{\text{月}}}{N}\frac{N_{\text{合格}}}{N}} \\ = {} & \frac{N_{\text{月, 合格}}}{N} \log \frac{N N_{\text{月, 合格}}}{N_{\text{月}} N_{\text{合格}}} \end{aligned} \quad (9)$$

のように，式変形することができる．他の項についても同様の式変形を行うと，曜日と成績の相互情報量は

$$\begin{aligned} & I(\text{曜日}; \text{成績}) \\ = {} & \frac{N_{\text{月, 合格}}}{N} \log \frac{N N_{\text{月, 合格}}}{N_{\text{月}} N_{\text{合格}}} + \frac{N_{\text{月, 不合格}}}{N} \log \frac{N N_{\text{月, 不合格}}}{N_{\text{月}} N_{\text{不合格}}} \\ + {} & \frac{N_{\text{火, 合格}}}{N} \log \frac{N N_{\text{火, 合格}}}{N_{\text{火}} N_{\text{合格}}} + \frac{N_{\text{火, 不合格}}}{N} \log \frac{N N_{\text{火, 不合格}}}{N_{\text{火}} N_{\text{不合格}}} \\ + {} & \frac{N_{\text{水, 合格}}}{N} \log \frac{N N_{\text{水, 合格}}}{N_{\text{水}} N_{\text{合格}}} + \frac{N_{\text{水, 不合格}}}{N} \log \frac{N N_{\text{水, 不合格}}}{N_{\text{水}} N_{\text{不合格}}} \\ + {} & \frac{N_{\text{木, 合格}}}{N} \log \frac{N N_{\text{木, 合格}}}{N_{\text{木}} N_{\text{合格}}} + \frac{N_{\text{木, 不合格}}}{N} \log \frac{N N_{\text{木, 不合格}}}{N_{\text{木}} N_{\text{不合格}}} \\ + {} & \frac{N_{\text{金, 合格}}}{N} \log \frac{N N_{\text{金, 合格}}}{N_{\text{金}} N_{\text{合格}}} + \frac{N_{\text{金, 不合格}}}{N} \log \frac{N N_{\text{金, 不合格}}}{N_{\text{金}} N_{\text{不合格}}} \end{aligned} \quad (10)$$

と表すことができるため，相互情報量はクロス集計表から容易に導出可能である．

(a) 一般的な形式

		成績	
		合格	不合格
曜日	月	$N_{月,\ 合格}$	$N_{月,\ 不合格}$
	火	$N_{火,\ 合格}$	$N_{火,\ 不合格}$
	水	$N_{水,\ 合格}$	$N_{水,\ 不合格}$
	木	$N_{木,\ 合格}$	$N_{木,\ 不合格}$
	金	$N_{金,\ 合格}$	$N_{金,\ 不合格}$

(b) ビュー形式

曜日	成績	集計値
月	合格	$N_{月,\ 合格}$
月	不合格	$N_{月,\ 不合格}$
火	合格	$N_{火,\ 合格}$
火	不合格	$N_{火,\ 不合格}$
水	合格	$N_{水,\ 合格}$
水	不合格	$N_{水,\ 不合格}$
木	合格	$N_{木,\ 合格}$
木	不合格	$N_{木,\ 不合格}$
金	合格	$N_{金,\ 合格}$
金	不合格	$N_{金,\ 不合格}$

図 3　曜日と成績のクロス集計表

3.2 基本方針

前節を踏まえると，分割されたデータを一つに結合する前に，任意の二属性についてクロス集計表に相当するビューを求めることで，本研究の目的である結合の削減が達成されるといえる．そこで，本節ではこれを実現するための提案手法について，基本方針を説明する．

本手法は以下の 3 つのステップで構成されている．

ステップ 1: 導出処理　外部キーを含むクロス集計表を導出
ステップ 2: 変換処理　外部キーを従属する別の属性に変換
ステップ 3: 抽出処理　同項目を集約

これらの手順を踏むことで，分割されたデータを一つに結合する前に，あるクロス集計表から別のクロス集計表を連鎖的に導出することが可能となる．ここでは，各ステップについて図 1 を用いて説明する．

まず，ステップ 1 では木構造のスキーマの根に相当するテーブル，つまり，図 1(a) 履修テーブルから外部キーを含むクロス集計表を導出する．例えば，図 4 科目番号と成績のクロス集計表は以下の問合せによって導出可能である．

```
CREATE MATERIARIZED VIEW 科目番号_成績 AS
SELECT 科目番号, 成績, COUNT(*) AS 集計値
FROM 履修
GROUP BY 科目番号, 成績
```

続いて，ステップ 2 ではステップ 1 で導出したクロス集計表の外部キーを従属する別の属性に変換する．例えば，科目番号 → 曜日 という関数従属性を活用することで，図 4 科目番号と成績のクロス集計表を図 5(a) 曜日と成績のクロス集計表のように変換する．

最後に，ステップ 3 ではステップ 2 で変換したクロス集計表について同項目を集約する．図 5(a) は同一の曜日が複数行にわたって存在しているため，図 5(b) のように集約する．なお，ステップ 2，ステップ 3 は以下の問合せによって実装可能である．

```
CREATE MATERIARIZED VIEW 曜日_成績 AS
SELECT 曜日, 成績, SUM(集計値) AS 集計値,
FROM 科目 NATURAL JOIN 科目番号_成績
GROUP BY 曜日, 成績
```

また，今回 科目番号 → 曜日 という関数従属性を活用したが，科目番号 → 教員番号 という関数従属性を活用すると，同様に以下の問合せによって，図 4 科目番号と成績のクロス集計表から図 6(a) を経由し，図 6(b) 教員番号と成績のクロス集計表を導出することができる．

```
CREATE MATERIARIZED VIEW 教員番号_成績 AS
SELECT 教員番号, 成績, SUM(集計値) AS 集計値
FROM 科目 NATURAL JOIN 科目番号_成績
GROUP BY 教員番号, 成績
```

さらに，この図 6(b) に対して，例えば，教員番号 → 勤続年数 という関数従属性を活用すると，同様に以下の問合せによって，図 7(a) を経由し，図 7(b) 勤続年数と成績のクロス集計表を導出することができる．

```
CREATE MATERIARIZED VIEW 勤続年数_成績 AS
SELECT 勤続年数, 成績, SUM(集計値) AS 集計値
FROM 教員 NATURAL JOIN 教員番号_成績
GROUP BY 勤続年数, 成績
```

このように，本手法では分割されたデータを一つに結合する前に，キー制約や従属性などのスキーマ情報を活用することで，木構造のスキーマの根から葉にかけて連鎖的に任意の二属性についてのクロス集計表を導出する．そして，導出したクロス集計表から式 (10) のように相互情報量を求め，mRMR によって特徴選択を行うことで結合を削減する．

4 実　　験

本章では，実世界のデータセットを利用した実験を行い，データベース中のスキーマ情報や統計情報，問合せ機能を活用した特徴選択の有用性や妥当性を確認する．

科目番号	成績	集計値
#1	合格	46
#1	不合格	3
#2	合格	54
#2	不合格	7
#3	合格	30
#3	不合格	1
-	-	-
#1000	合格	92
#1000	不合格	14

図 4　科目番号と成績のクロス集計表

(a) 集約前

曜日	成績	集計値
月	合格	46
月	不合格	3
月	合格	54
月	不合格	7
火	合格	30
火	不合格	1
-	-	-
金	合格	92
金	不合格	14

(b) 集約後

曜日	成績	集計値
月	合格	46 + 54 + ...
月	不合格	3 + 7 + ...
火	合格	30 + ...
火	不合格	1 + ...
水	合格	-
水	不合格	-
木	合格	-
木	不合格	-
金	合格	92 + ...
金	不合格	14 + ...

図 5　曜日と成績のクロス集計表

4.1　データセット

データセットは Instacart という食料品を即日配達するインターネットサービスの注文履歴 [11] を用いる．図 8 にデータセットのスキーマを示す．このデータセットは注文と商品の組となるテーブルを根とした木構造のスキーマに従っているため，提案手法に対応している．

また，本実験におけるデータセットの分析シナリオとしては，

(a) 集約前

教員番号	成績	集計値
#1	合格	46
#1	不合格	3
#2	合格	54
#2	不合格	7
#2	合格	30
#2	不合格	1
-	-	-
#700	合格	92
#700	不合格	14

(b) 集約後

教員番号	成績	集計値
#1	合格	46 + ...
#1	不合格	3 + ...
#2	合格	54 + 30 + ...
#2	不合格	7 + 1 + ...
-	-	-
#700	合格	92 + ...
#700	不合格	14 + ...

図 6　教員番号と成績のクロス集計表

(a) 集約前

勤続年数	成績	集計値
5	合格	46 + ...
5	不合格	3 + ...
9	合格	54 + 30 + ...
9	不合格	7 + 1 + ...
-	-	-
5	合格	92 + ...
5	不合格	14 + ...

(b) 集約後

勤続年数	成績	集計値
-	-	-
5	合格	46 + 92 + ...
5	不合格	3 + 14 + ...
-	-	-
9	合格	54 + 30 + ...
9	不合格	7 + 1 + ...
-	-	-

図 7　勤続年数と成績のクロス集計表

どのような場合に商品が再注文されるかを知るため，再注文フラグに関する決定木を生成するというものを想定している．

4.2 実験方法

まず，複数に分割されたテーブルを結合する前に提案手法による特徴選択を行い，特徴量の選択順を確認する．その後，選択順の結果に基づき，一部の属性のみを特徴量とした場合と，すべての属性を特徴量とした場合について，結合時間・学習時間・正解率を比較する．なお，すべての学習において，3 分割交差検証を行う．また，実装には PostgreSQL 11.6 [12]，および Python 3.6.10 を使用する．

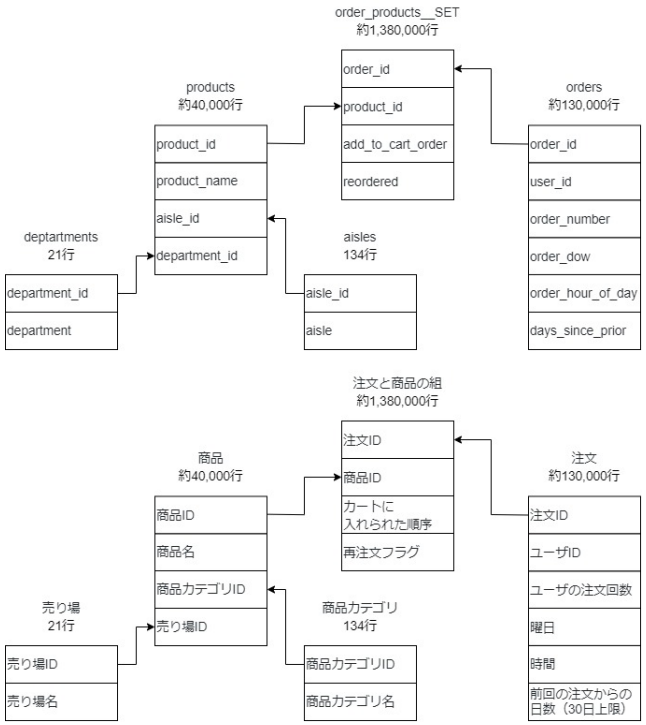

図 8　データセットのスキーマ

4.3 実験結果

■ 特徴量の選択順

図 9 に mRMR に基づいた特徴量の選択順を示す．なお，番号が若いほど，先に選択された学習に有用な特徴量となっている．

図 9 のとおり，注文 ID という属性が最初に選択されるという結果となった．この結果は，注文した商品すべてが再注文である，もしくは，再注文でないというデータセットの傾向を表している．つまりは，新規顧客は当然，注文した商品すべてが再注文でない商品である一方で，既存顧客はリピート商品のみの注文が多いという傾向があると考えられる．

また，カートに入れられた順序という属性が 2 番目に選択されるという結果となった．この結果は，カートに入れられた順序によって，再注文である，もしくは，再注文でないということを決定しやすい傾向にあることを表している．つまりは，リピート商品は先にカートに入れられ，初購入の商品はリピート商品の後からついでにカートに入れられる傾向があると考えられる．

これらの結果や，結合するテーブル数を考慮し，結合時間・学習時間・正解率については，選択数 3（注文と商品の組テーブル・注文テーブルを結合し，再注文フラグ・注文 ID・カートに入れられた順序・時間を射影），選択数 6（注文と商品の組テーブル・注文テーブル・商品テーブル・売り場テーブルを結合し，再注文フラグ・注文 ID・カートに入れられた順序・時間・曜日・売り場 ID・ユーザの注文回数を射影）および選択数 13（すべてのテーブルを結合し，すべての属性を射影）を比較する．

■ 結合時間・学習時間・正解率

表 1 に選択数 3・選択数 6・選択数 13（すべて）についての結合時間・学習時間・正解率を示す．

まず，学習時間・正解率に関しては，選択数 6 は選択数 13（すべて）と比べて，学習時間は半分以下となり，正解率も下がらなかった．この結果から，データベース中のすべての属性を特徴量とすることなく，一部の属性のみを特徴量とすることは妥当であると考えられる．

一方，結合時間に関しては，選択数 3・選択数 6・選択数 13 のいずれの場合も大きな差は現れなかった．これは，選択数 3・選択数 6 のいずれの場合も，注文と商品の組テーブル（約 1,380,000 行）と注文テーブル（約 130,000 行）という行数の多いテーブル間の結合を削減できておらず，比較的行数の少ないテーブルについてのみ，結合が削減されているためであると考えられる．特に，選択数 6 に関しては，売り場テーブル（21 行）の売り場 ID を特徴量として利用するために，自身の属性は特徴量として利用されない商品テーブル（約 40,000 行）までも結合しているため，選択数 13（すべて）との間で大きな差が現れなかったと考えられる．

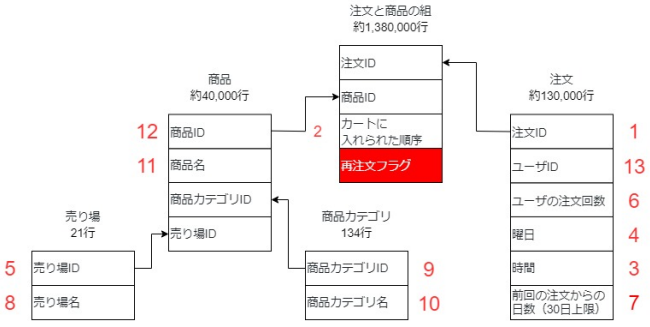

図 9　mRMR に基づいた特徴量の選択順

表 1　選択数ごとの結合時間・学習時間・正解率

選択数	結合時間（s）	学習時間（s）	正解率
3	1.68	33.1	57.0%
6	2.19	20.0	62.7%
13（すべて）	2.53	50.6	61.9%

5 結　　論

5.1 ま と め

ビッグデータ時代の今日では，データベース中のデータも膨大となっており，高品質かつ効率的な機械学習が求められる．これを実現するため，データベースが持つスキーマ情報や統計情報，各種機能などを活用するというアプローチが存在する．これを踏まえ，本研究では，キー制約や従属性，問合せ機能を活用した特徴選択を行うことで，学習の質を保ちつつ，処理コストを抑えた．

5.2 今後の課題

■ テーブル数の多いデータセットの利用

本論文の実験で用いたデータセットは，5 つのテーブルから構成されているが，実際の企業などのデータベースは何十，何百ものテーブルから構成されている．そのため，本論文の実験では，結合時間が分析全体に与える影響は小さかったが，実際の現場では結合時間が分析のボトルネックとなることも考えられる．そこで，今後の課題として，テーブル数が多いデータセットを利用した実験を行うことで，より実情に即した結合時間の影響を調査したい．

■ 分析系データベースの利用

本論文の実験では，PostgreSQL という基幹系データベースを用いたが，より分析に特化した分析系データベースを用いることで，提案手法を最大限に活かすことができると考えられる．例えば，C-Store [13] や MonetDB [14] は列指向データベースとも呼ばれ，一般的な行単位ではなく，列単位でデータが格納されており，列単位でのデータの集約などを効率的に行うことができる．また，Microsoft SQL Server [15] では PIVOT 演算を用いることで，クロス集計表を効率的に導出することができる．そこで，今後の課題として，これらの分析系データベースを利用した実験を行うことで，さらなる処理コストの抑制を目指したい．

謝　　辞

本研究の一部は，科研費 16H01722 および 19K21530 による．

文　　献

[1] M. Boehm, A. Kumar, and J. Yang, *Data Management in Machine Learning Systems*. Morgan & Claypool, 2019.

[2] A. Kumar, J. Naughton, J. M. Patel, and X. Zhu, “To join or not to join?: Thinking twice about joins before feature selection,” in *Proc. SIGMOD*, pp. 19–34, 2016.

[3] V. Shah, A. Kumar, and X. Zhu, “Are key-foreign key joins safe to avoid when learning high-capacity classifiers?,” *Proc. VLDB*, vol. 11, no. 3, pp. 366–379, 2017.

[4] Z. Abedjan, L. Golab, and F. Naumann, “Profiling relational data: A survey,” *VLDB J.*, vol. 24, no. 4, pp. 557–581, 2015.

[5] J. Liu, J. Li, C. Liu, and Y. Chen, “Discover dependencies from data — a review,” *IEEE TKDE*, vol. 24, no. 2, pp. 251–264, 2010.

[6] S. Ma, L. Duan, W. Fan, C. Hu, and W. Chen, “Extending conditional dependencies with built-in predicates,” *IEEE TKDE*, vol. 27, no. 12, pp. 3274–3288, 2015.

[7] W. Fan, F. Geerts, J. Li, and M. Xiong, “Discovering conditional functional dependencies,” *IEEE TKDE*, vol. 23, no. 5, pp. 683–698, 2010.

[8] G. Cormode, L. Golab, F. Korn, A. McGregor, D. Srivastava, and X. Zhang, “Estimating the confidence of conditional functional dependencies,” in *Proc. SIGMOD*, pp. 469–482, 2009.

[9] H. Peng, F. Long, and C. Ding, “Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy,” *IEEE TPAMI*, no. 8, pp. 1226–1238, 2005.

[10] C. Ding and H. Peng, “Minimum redundancy feature selection from microarray gene expression data,” *Journal of bioinformatics and computational biology*, vol. 3, no. 02, pp. 185–205, 2005.

[11] The Instacart Online Grocery Shopping Dataset 2017 `https://www.instacart.com/datasets/grocery-shopping-2017` (accessed: February 12, 2020).

[12] PostgreSQL: The world’s most advanced open source relational database `https://www.postgresql.org/` (accessed: February 12, 2020).

[13] M. Stonebraker, D. J. Abadi, A. Batkin, X. Chen, M. Cherniack, M. Ferreira, E. Lau, A. Lin, S. Madden, E. O’Neil, P. O’Neil, A. Rasin, N. Tran, and S. Zdonik, “C-store: A column-oriented DBMS,” in *Proc. VLDB*, pp. 553–564, 2005.

[14] MonetDB: Home: `https://www.monetdb.org/` (accessed: February 12, 2020).

[15] SQL Server 2019 | Microsoft: `https://www.microsoft.com/ja-jp/sql-server/sql-server-2019` (accessed: February 12, 2020).