「マルチメディア，分散，協調とモバイル
(DICOMO2019) シンポジウム」 令和元年7月

外部キー制約を考慮した特徴量削減手法

長 裕敏[1] 山室 健[1] 内山 寛之[1]

概要：機械学習において，特徴選択は学習モデルの精度を向上させるために必要不可欠な処理であるが，特徴量数が増えると最適な組み合わせを選択するには膨大な処理時間を要する．先行研究では，DBMS から学習データを取得する際，外部キー制約を持つテーブル間のレコード数比が十分に大きい場合に，学習モデルの精度への影響を抑えながら特徴選択を行う前に特徴量数を削減する手法が提案されている．しかし，レコード数比が小さい実データも多く存在するため適用範囲が限定的という課題がある．そこで本研究では，上記の既存手法が適用できない条件下において，学習モデルの精度への影響を抑えながら外部キーが参照するテーブルの全特徴量を一次元の特徴量に変換し，結果の特徴量数を削減する手法を提案する．性能測定実験において，既存手法では条件を満たせず特徴量数を削減できなかった 7 個のテーブルのうち，提案手法では 3 個のテーブルに対して新たに特徴量数を削減可能なことを示し，学習モデルの精度への影響を抑えながら特徴選択を高速化可能なことを確認した．

A Feature Reduction Method Under Foreign Key Constraints

Hirotoshi Cho[1] Takeshi Yamamuro[1] Hiroyuki Uchiyama[1]

1. 背景

企業データを活用した意思決定やサービス改善では機械学習によるデータ分析を行うことが多くなり，DBMS(Database Management System) 分野と ML(Machine Learning) 分野における横断的な最適化技術への関心が高まっている [5,6,7]．一般的に企業が管理するデータは，主キーと外部キーの依存関係を持つ DBMS 上の正規化された複数のテーブルである．一方で，既存の ML ツールに実装された学習アルゴリズムの多くは入力データとして単一のテーブルを想定している．そのためデータ分析者は DBMS 上に格納された複数のテーブルを結合[*1]することで単一のテーブルで構成される学習データに事前に生成する必要がある．

学習データから有用な特徴量の組み合わせを抽出する特徴選択 [3,4,12] は，ML 分野において学習モデルの精度を改善するための重要な前処理である．特徴選択には，学習モデルを構築して精度比較をすることで有用な特徴量の部分集合を計算する手法や，学習モデル自体に組み込まれている手法など様々ある．しかし特徴量が x 個あるとき $2^x - 1$ 個の部分集合があるため，特徴量数が多くなると計算が遅くなる問題がある[*2]．そこで本研究では，格納されたデータのスキーマ情報や統計情報を活用して，学習モデルの精度への影響を抑えながら，特徴量数を削減可能な手法を検討する．

先行研究では，スキーマで定義された外部キー制約を持つテーブル間の結合を省略して，外部キーが参照するテーブルの特徴量の集合を削除した場合に，学習モデルのバリアンスが大きくなることを抑えるための様々な条件を分析している [1,2]．結果として，図 1 の概要で示すようにレコード数比が十分に大きい場合に，学習モデルの精度への影響を抑えながら結合を省略する手法を提案している．図 1 のテーブル S，R_1，R_2 は DBMS 上の正規化されたテーブル群 (レコード数をそれぞれ n_S, n_{R_1}, n_{R_2} とする) であり，S-R_1 間と S-R_2 間は主キーと外部キーの依存関係がある．この手法ではテーブル S，R_1，R_2 を結合して単一のテーブル T を生成する際に，レコード数比が十分に大きいテーブル S-R_1 間の結合（$n_S \gg n_{R_1}$）を省略することで特

[1] NTT ソフトウェアイノベーションセンタ
NTT Software Innovation Center

[*1] SQL における JOIN 操作のこと．

[*2] 例えば前者の特徴選択手法に単純ベイズを用いた場合，レコード数を n としたときの計算量は $O(x^2 n)$ である．

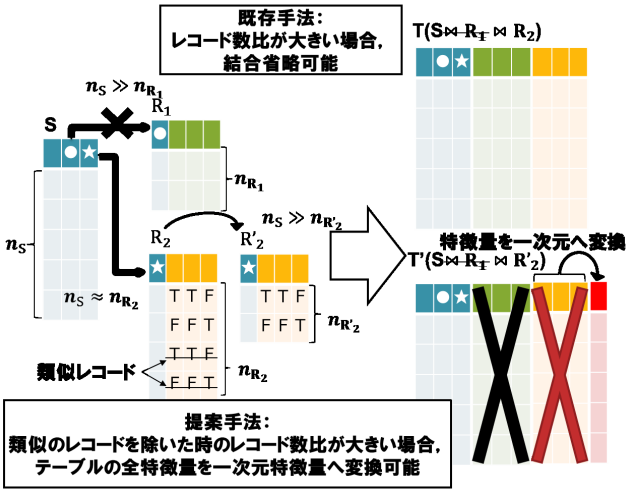

図 1　既存手法と提案手法の概要図

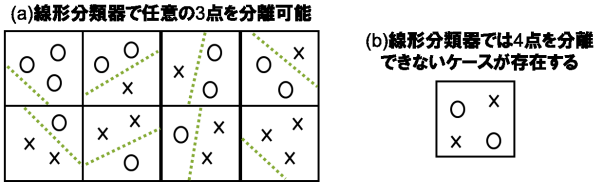

図 2　2 次元特徴空間における線形分類器の VC 次元

徴量数を削減（図 1 右上の緑色のカラム群）することができる．しかし，テーブル S-R_2 間のようにレコード数比が十分に大きくない場合 ($n_S \approx n_{R_2}$) は，学習モデルの精度への影響が大きいと判断され，テーブル R_2 が持つ特徴量（図 1 右上の黄色のカラム群）を削減することは出来ない．この手法はレコード数比しか考慮に入れないため適用は容易だが，レコード数比が小さい結合も多くあるため適用範囲が限定的である．

そこで本研究では上記の既存手法が適用できない条件下において，学習モデルの精度への影響を抑えながら外部キーが参照するテーブルの全特徴量を一次元の特徴量に変換することで，特徴選択によって得られる特徴量数を削減する手法を提案する．本提案手法ではテーブル間のレコード数比の代わりに，外部キーが指すテーブル（図 1 のテーブル R_2）において，類似性に基づくレコード削減を行う。削減後のレコード数と、外部キーを含むテーブルのレコード数との比を判定指標に用いる．類似のレコードを除去することでレコード数比を大きくすることができるため，既存手法より多くのテーブルに対して特徴量削減が可能となる．判定指標の条件を満たした場合（図 1 で類似レコードを除いたテーブル R_2 をテーブル R'_2 とすると，$n_S \gg n_{R'_2}$），図 1 のテーブル T' が示すように，外部キーが参照しているテーブル R'_2 の全特徴量を一次元の特徴量（図 1 右下の赤色のカラム）に変換することで学習モデルのバリアンスが大きくなることを抑えながら特徴量数を削減できる．

本研究における貢献は以下の通りである．

- 外部キーが参照するテーブルの類似レコードを取り除くことでテーブル間のレコード数比を十分に大きく出来る場合に，学習モデルの精度への影響を抑えながら特徴量数を削減する手法を提案した．
- 提案する特徴量数削減の判定指標が，常に既存手法の判定指標より適用条件が広いことを分析した．
- 既存手法と同様の性能評価を行い，既存手法では条件を満たせず全特徴量を利用しなければならなかった 7 件のデータセットのうち，提案手法では 3 件を新たに特徴量削減することが可能となり，学習モデルの精度への影響を抑えながら特徴選択の計算時間を全特徴量を使用した場合と比較して FFS での特徴選択時に平均 9.0 倍, BFS での特徴選択時に平均 121.2 倍高速化した．

本論文の構成は以下の通りである．まず 2 章で本論文が扱う問題設定と既存手法について述べる．3 章では提案する一次元特徴変換による特徴量削減手法について詳述し，4 章で既存手法と提案手法の比較実験の結果を示す．5 章では関連研究を，最後の 6 章でまとめについて述べる．

2. 準備

2.1 問題設定と前提

外部キー制約を持つ k 個のテーブルから学習データを抽出し，特徴量変換及び特徴選択をした上で学習モデルを作成する場合を考える．本論文では [1, 2] と同様の表記を用いる．外部キーを持つテーブルを S, 外部キーで参照されるテーブルを $R_i (i = 1, ..., k)$ とする．図 1 では $k = 2$ の例を示している．S のスキーマは $S(Y, X_S, FK_1, \cdots, FK_k)$, R_i のスキーマは $R_i(RID_i, X_{R_i})$ で構成されるテーブルである．括弧内の各要素は属性を示す．S と R_i のレコード数をそれぞれ n_S と n_{R_i} とする．Y を目的変数として，X_S と各 X_{R_i} はそれぞれのテーブルに属する特徴量の集合とする．FK_i は R_i に対する外部キーであり，RID_i は R_i の主キーと定義し，FK_i と RID_i 間は外部キー制約 ($|D_{FK_i}| \leq |D_{RID_i}|$) を持つものとする ($D_x$ は x のとりうる値 e.g., x : 性別の時，D_x={ 男性, 女性 } であり $|D_x| = 2$)．S と各 R_i に対して FK_i と RID_i で結合処理 ($S \bowtie_{FK_i=RID_i} R_i$) を行い，単一のテーブル $T(Y, X_S, FK_1, \cdots, FK_k, X_{R_1}, \cdots, X_{R_i})$ を抽出し，学習モデルを構築する．線形モデルを前提とし，4 章では単純ベイズ [10,11] を用いて評価を行う．また，説明を簡単にするために性能評価時を除いて R_i は単一 ($k = 1$) であると想定して議論を進め，$i = 1$ の時 R_i を R と表記する ($k \geq 2$ の時は $k = 1$ と同様の手法を繰り返すことで対応する)．

2.2 VC 次元

VC 次元は汎化誤差を評価するために導入された仮説空

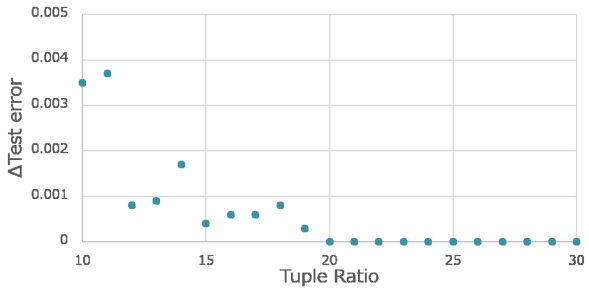

図 3　単純ベイズを用いて，全特徴量を使用した場合と特徴量削減した場合でモデル構築した時の汎化誤差の上昇量

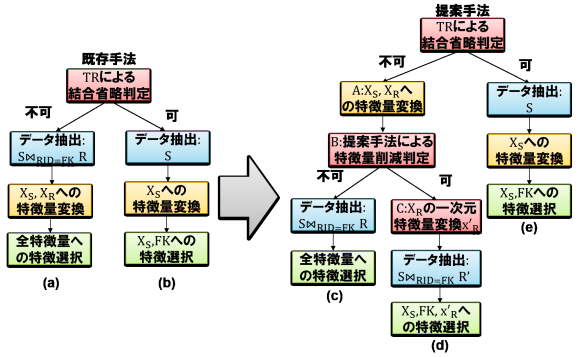

図 4　既存手法と提案手法の処理の流れ

間[*3]の複雑さの度合いを表す指標である．一般的に VC 次元が高いほど訓練誤差は下がりやすいが過学習しやすい傾向がある．Y について 2 値の分類を行う分類器の集合 F を考える時，VC 次元は F に属する分類器によって完全に分離できる最大の点の数である [15]．

具体例として入力データを 2 次元平面上の点とし，線形分類器を用いて Y を 2 クラス分類する場合を考える．点の数が 3 つの場合，Y の組み合わせは図 2(a) が示すように 8 通り存在するが，いずれの組み合わせでも Y を分離できる分類器が存在するのが分かる．しかしながら，点の数が 4 つの場合には図 2(b) が示すように Y が交互に異なる値を示す組み合わせの場合に Y を分離することができないため，VC 次元は 3 となる．以降の説明で用いる単純ベイズの VC 次元は [1] と同様に各特徴を離散化させた X を入力する特徴群とした時に，$1+\sum_{x\in X}(|D_x|-1)$ で表される．

VC 次元を v, ML に用いる訓練データ数を n, 危険率を $\sigma \in (0,1)$ として $n > v$ の時に訓練誤差と汎化誤差との差に関して以下の定理が成り立つ．

$$|Test\ error - Train\ error| \leq \frac{4+\sqrt{v\log\left(\frac{2en}{v}\right)}}{\sigma\sqrt{2n}} \quad (1)$$

2.3　外部キー制約を用いた特徴量削減手法

先行研究として [1,2] では結合時に，精度に大きな影響を及ぼさない結合について省略して外部キーが指すテーブルの全特徴量を削減する手法を提案している．

既存手法では定理 (1) を用いて結合省略によって学習モデル精度に与える影響を定量化する指標を導入している．結合省略して X_R を除いた時の VC 次元を v_S, 結合して全特徴量を用いる時の VC 次元を v_{all} とすると結合省略することで生じるリスク ROR(Risk of Representation) [1] は以下で表される．

$$ROR \leq \frac{\sqrt{v_S\log\left(\frac{2en_S}{v_S}\right)}-\sqrt{v_{all}\log\left(\frac{2en_S}{v_{all}}\right)}}{\sigma\sqrt{2n_S}}$$

ROR が極小であれば結合を省略して特徴量数を削減しても学習モデルのバリアンスの変化は極小であるため，結合省略を行う判定指標とできる．既存手法では ROR に対して強い相関を持つ TR (Tuple Ratio) というテーブル間のレコード数比のみで適用確認が可能な判定指標に ROR を代替している．TR $= \frac{n_S}{n_R}$ で表される．既存手法では実験的に TR ≥ 20 ならば結合を省略しても精度への影響は極小に収まることを示している．図 3 に予備実験結果を示す．学習モデルに単純ベイズを用いて以下の条件で実験を行った．テーブルは S と単一の R を用い，Y, X_S, X_R は全て $\{0,1\}$ をとる．R の単一特徴量 $x_R \in X_R$ を真の分布とし $P(Y=0|x_R=0)=P(Y=1|x_R=1)=0.9$ とする．X_S と X_R の特徴量数を 4，$n_S=10000$ として，n_R を可変値とした．図 3 の横軸は TR を示し，縦軸は結合を省略した場合のモデルの汎化誤差の上昇量を示している．図 3 が示すように TR≥ 20 ならば結合を省略しても精度に影響がないため，X_R を DBMS のスキーマ情報のみで削減することができる．

図 4(a)(b) に既存手法の処理の流れを示す．TR の判定条件を満たした場合，図 4 の (b) のフローが示すように R に対する結合処理を省略し S のみを抽出することで，X_R に対する処理が不要となるので特徴選択の計算時間を大きく削減できる．しかしながら，TR が十分に大きくないと精度に影響があるため適用範囲が限定的であり，多くの場合図 4 の (a) のフローにあるように全特徴量で特徴選択するため計算が遅くなる課題がある．

本論では，DBMS 上の外部キー制約や統計情報を活用し，既存手法が適用できない条件下において，学習モデルの精度への影響を抑えながら特徴選択を行う前に特徴量数を削減可能な手法を示すことで，特徴選択時にかかる膨大な処理時間の問題を解決する．

3.　提案手法

3.1　外部キーで参照される特徴量 X_R の削減手法

本論文では X_R が持つレコード間の冗長性に着目し，類似のレコードが同様の離散値を持つように，X_R を一次元の特徴量に変換することで特徴量数を削減する手法を提案

[*3] 入力空間から出力空間へ変換する関数の集合．

する．直感的に，Y と X_R に関連性が高い場合に，X_R が類似した値を示すならば Y は等しい値を示す可能性が高いことが考えられる．例えば，X_R が趣味や年齢，年収といったユーザ情報を表し，Y を商品を購入するか否かで 2 値分類予測する場合を考えた時，趣味や年齢，年収が近しいユーザは同一の Y を示すことが考えられる．既存手法では X_R に高い類似性がある場合でも，X_R と FK の間にある関連性が考慮されていないという課題がある．

そこで我々はこの課題に対して，類似レコードをまとめることで，既存手法では適用できない範囲においても，特徴量を削減可能な手法を考案した．本手法は，X_R の持つ情報量をレコードの類似性に着目して抽出し，当該情報量に基づくテーブル間のレコード数比を算出する．

図 4(c-e) に提案する特徴量削減手法の処理の流れを示す．TR の条件を満たす場合は図 4 の (e) のフローが示すように既存手法と同様の処理を行う．提案手法では既存手法の限定的な適用範囲の問題を解決するために，TR が適用できない条件下において，TR より緩い条件で特徴量削減を行う判定指標を導入する．

既存手法ではテーブル間のレコード数比で特徴量削減判定を行うが，本提案手法では R に対して類似レコードを除去した時のレコード数 n'_R と n_S の比を判定指標 UTR(Unique Tuple Ratio) として用いる (図 4 の B)．UTR $= \frac{n_S}{n'_R}$ で表し，3.2 節で詳細を述べる．

類似レコードの除去を効率的に行うために，判定処理を行う前に binning [10] を行い類似のレコードが同様の離散値を持つように特徴量変換を行う (図 4 の A)．UTR の条件を満たした場合，X_R を一次元の特徴量に変換を行う (図 4 の C)．これにより特徴選択を行う前に X_R の特徴量を削減できるため，特徴選択の計算時間を大きく削減できる．X_R の一次元の特徴量への変換については 3.3 節で詳細に述べる．

3.2 特徴量削減の判定指標

本節では学習モデル精度に影響を与えずに X_R を一次元の特徴量 x'_R に変換可能か判定する UTR を導出し，TR より緩い条件で特徴量数を削減可能なことを示す．本提案においても定理 (1) を用いて，X_R を一次元の特徴量に変更することで生じるリスクを定量化する．X_R を一次元の特徴量に変換したときの VC 次元を v_{one} とした時，X_R を一次元の特徴量に変換することで生じるリスク ROR' は以下で表される．

$$ROR' \leq \frac{\sqrt{v_{one}\log\left(\frac{2en_S}{v_{one}}\right)} - \sqrt{v_{all}\log\left(\frac{2en_S}{v_{all}}\right)}}{\sigma\sqrt{2n_S}}$$

ROR' は ROR と同様に特徴量削減を行った場合と全特徴を使用した場合の VC 次元における定理 (1) の差分を表しているため，ROR' が極小ならば学習モデルのバリアンスへの影響を抑えつつ特徴量を削減することができる．x'_R を離散化した X_R を表すカテゴリ変数となるように一次元の特徴量へ変換すると，v_{one} は $v_{one} = \sum_{F\in U_S}(|D_F|-1)+|D_{x'_R}|$ に置き換えられる．この時，ROR' に対して [1] と同様の条件で式変形を行うことで，以下のように近似できる．

$$ROR' \approx 1/\sqrt{n_S/|D_{x'_R}|} \tag{2}$$

また [1] より ROR は以下のように近似できる．

$$ROR \approx 1/\sqrt{n_S/|D_{FK}|} \tag{3}$$

この時 [1] の仮定より $|D_{FK}| = n_R \leq n_S$ かつ，$D_{x'_R}$ は D_{FK} から X_R が等しい FK を除去した集合なので，$n'_R = |D_{x'_R}| \leq |D_{FK}| \leq n_S$ である．この条件下において式 (2)(3) より ROR' $\leq$ ROR なので，既存手法より特徴量数を削減することで生じるリスクは小さいと言える．従って，ROR' が極小，すなわち UTR $= \frac{n_S}{n'_R}$ が十分に大きければ精度への影響を抑えながら X_R を一次元の特徴量 x'_R に変換できる．また，$n'_R \leq n_R(= |D_{FK}|)$ より TR $\leq$ UTR であるため，TR の判定指標より緩い条件で特徴量削減判定を行うことが可能である．

n'_R については，図 4 の提案手法の A における特徴量変換で X_R に対して binning 処理を行い離散化を行った上で，集約・カウントを行うことで求めることができる．計算量は $O(n_R)$ であり，全特徴量を用いた時の特徴選択にかかる計算量と比較すると非常に小さいので，UTR の判定条件を満たせずに図 4 の (e) のフローで実行される場合でも処理時間に与える影響は極小に抑えることができる．

3.3 X_R の一次元特徴変換

UTR の判定条件を満たした場合，X_R に対して一次元特徴変換を行うことで，特徴選択の前に特徴量を削減する．X_R の一次元特徴変換は SQL1 で行う．SQL1 では window 関数を用いることで，X_R が同じ値を示すレコードごとにグルーピングし，グループごとに同一の番号を付与することで一次元の特徴量 x'_R を生成し，X_R の代替として用いる．具体例として図 5 にユーザ情報を格納した R に対して一次元特徴変換を行う例を示す．図 5 中の R は (UserID) が RID, (Sex, Age, Income) が X_R である．PARTITION BY によって X_R が同一のデータでグループ分けを行い，FIRST_VALUE 関数でグループ内の初頭の RID を x'_R とすることで，x'_R の値は X_R を表すカテゴリ変数となる．図 5 の例だと $x'_R = 1$ ならば (F, 10 代, 400 万) を表し，$x'_R = 3$ ならば (M, 20 代, 500 万) を表す．これにより，X_R を一次元の特徴量に削減可能なので特徴選択にかかる時間を削減することが可能となる．

SQL 1

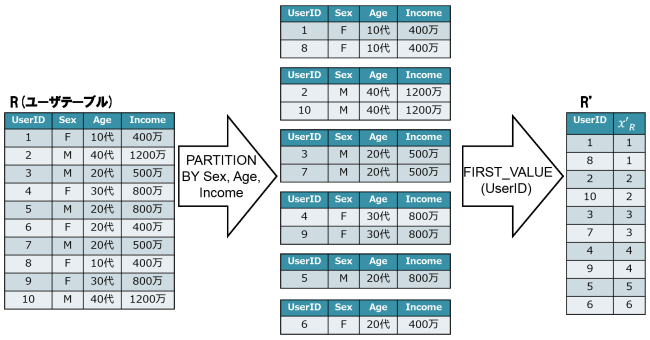

図 5　X_R の一次元特徴変換例

```
SELECT RID, FIRST_VALUE(RID)
OVER (PARTITION BY X_R) AS x'_R FROM R
```

4. 実験評価

提案手法の評価のために実データセットに対して提案手法を適用した上で性能測定を行う．性能評価によって，既存手法より適用条件が緩和されているか確認し，精度に影響を与えることなくデータ分析を高速化できるかどうか検証する．

4.1 実験セットアップ

実験環境は Intel(R) Xeon(R) CPU E5-2676 v3 @ 2.40GHz, 32GB, R(versin 3.4.4) で行った．実験データセットには外部キー制約を持つ 6 つの公開データセットを用いた．データセットは Kaggle, GroupLens, openflights.org, mtg.upf.edu/node/1671 から取得できる．以下に各データセットと分析内容について説明する．

MovieLens1M：過去の映画の評価情報を保持した S, 映画のジャンルや公開年度といった映画情報を格納した R_1, 職業や性別といったユーザ情報を格納した R_2 を使用して映画の評価を 5 段階評価で予測する．

Yelp：過去のお店の評価情報を保持した S，立地やお店のジャンルといったお店情報を格納した R_1, 性別や過去の評価回数といったユーザ情報を格納した R_2 を使用して企業の評価を 5 段階評価で予測する．

Walmart：過去の売上高に関する情報を保持した S, 平均気温や平均燃料価格といった天気・経済指標に関する情報を格納した R_1, お店の種別や大きさなどの店舗情報を格納した R_2 を利用して売上高を 7 段階評価で予測する．

LastFM：ユーザの歌手ごとの音楽再生数に関する情報を保持した S, 手がける曲のジャンルや LastFM アプリでの再生数などの歌手情報を格納した R_1, 年齢や出身などのユーザ情報を格納した R_2 を使用して音楽再生数を 5 段階評価で予測する．

BookCrossing：過去の書籍に関する評価情報を保持した S, 年齢や出身といったユーザ情報を格納した R_1, 出版社や出版年度といった書籍情報を格納した R_2 を使用して書籍の評価を 5 段階評価で予測する．

Flights：飛行機の過去のフライト情報を保持した S, 航空会社の情報を格納した R_1, 出発する空港の情報を格納した R_2, 到着する空港の情報を格納した R_3 を利用してフライトがコードシェア便か否かの 2 値予測を行う．

表 1　実験データセットの統計量

Dataset	n_S	$\lvert X_S \rvert$	R	n_R	$\lvert X_R \rvert$	TR判定 (既存手法)	n'_R	UTR判定 (提案手法)
MovieLens1M	1,000,209	0	User	6,040	4	○	5,789	○
			Movie	3,706	21	○	2,354	○
Yelp	215,879	0	Business	11,537	32	×	10,369	×
			User	43,873	6	×	1,483	○
Walmart	421,570	1	Indicator	2,340	9	○	913	○
			Store	45	2	○	8	○
LastFM	343,747	0	Artist	4,999	7	○	147	○
			User	50,000	4	×	3,845	○
BookCrossing	253,120	0	User	27,876	2	×	356	○
			Book	49,772	4	×	26,030	×
Flight	66,548	20	Airlines	540	5	○	374	○
			SrcAirport	3,182	6	×	3,111	×
			DestAirPort	3,182	6	×	3,111	×

Flights：飛行機の過去のフライト情報を保持した S, 航空会社の情報を格納した R_1, 出発する空港の情報を格納した R_2, 到着する空港の情報を格納した R_3 を利用してフライトがコードシェア便か否かの 2 値予測を行う．

表 1 にデータセットの統計を示す．既存手法 [1] との比較のために各種データセットにおける年齢などの数値特徴は既存手法と同様に等長ヒストグラムで binning を行った．

実験では 3 つのアプローチを比較する．1. 手法を適用せず全てのテーブルの特徴量を対象とする場合．2. 既存手法を適用し全特徴量から TR の判定条件を満たしたテーブルの特徴量を除去する場合．3. 提案手法を適用し TR の判定条件を満たしたテーブルの特徴量を除去した上で UTR の判定条件を満たしたテーブルの特徴量を一次元特徴変換する場合．それぞれのアプローチについて特徴選択までにかかる時間と単純ベイズで生成したモデルの精度を比較する．特徴選択は Forward Feature Selection(FFS) と Backward Feature Selection(BFS) の 2 種類 [3,4] で検証を行った．学習モデル精度は 5 分割の交差検証を行うことで求め，2 値予測の Flights は正答率を比較し，それ以外のデータセットに対しては平均二乗誤差 (RMSE) の比較を行う．

4.2 実験結果

精度と適用範囲 表 1 にデータセットの統計量から TR 及び UTR の判定条件を満たしたか否かを示し，表 2 に各アプローチでの BFS と FFS を用いて特徴選択した特徴群で構築した学習モデルの汎化誤差を示す．TR 及び UTR の判定の閾値はともに 20 に設定した．表 1 が示す通り，既存手法では 13 個のテーブルに対して 6 個のテーブル (MovieLens の $R_{1,2}$, Walmart の $R_{1,2}$, LastFM の R_1, Flight の R_1) が TR の条件を満たしたので，該当テーブルの特徴量を全て除去し，提案手法ではさらに 3 個のテーブル (Yelp の R_2, LastFM の R_2, BookCrossing の R_1) が UTR の条件を満たしたため，該当テーブル特徴量について一次元特徴量変換した上で特徴選択を行った．

表 2 の結果が示す通り，事前の特徴量削減によって汎化誤差が大きく増大しないことが確認できる．BookCrossing

表 2　各アプローチでの学習モデル精度

	MovieLens1M(RMSE)			Yelp(RMSE)			Walmart(RMSE)			LastFM(RMSE)			BookCrossing(RMSE)			Flight(正答率)		
アプローチ	全特徴	TR	TR+UTR	全特徴	TR	TR+UTR	全特徴	TR	TR+UTR	全特徴	TR	TR+UTR	全特徴	TR	TR+UTR	全特徴	TR	TR+UTR
FFS精度	1.0661	1.0655	1.0655	1.1356	1.1356	1.1355	0.9041	0.9015	0.9015	0.9989	0.9981	0.9981	1.4170	1.4170	1.4170	0.8632	0.8634	0.8634
BFS精度	1.0672	1.0655	1.0655	1.1331	1.1331	1.1330	0.9041	0.9015	0.9015	1.0190	0.9981	0.9981	1.4278	1.4278	1.4278	0.8616	0.8618	0.8618

表 3　各処理に要する実行時間

	MovieLens1M			Yelp			Walmart			LastFM			BookCrossing			Flight		
アプローチ	全特徴	TR	TR+UTR	全特徴	TR	TR+UTR	全特徴	TR	TR+UTR	全特徴	TR	TR+UTR	全特徴	TR	TR+UTR	全特徴	TR	TR+UTR
UTR判定時間(s)						0.1117						0.0264			0.0864			0.0347
特徴変換時間(s)						0.0623						0.0693			0.0887			
結合時間(s)	0.877			0.257	0.257	0.257	0.2231			0.2734	0.191	0.2734	0.198	0.198	0.198	0.1266	0.092	0.092
特徴選択FFS時間(s)	706.889	17.528	17.528	305.559	305.559	206.15	168.987	18.998	18.998	63.645	39.672	31.028	43.324	43.324	39.009	301.867	292.396	292.396
特徴選択BFS時間(s)	10846.4	17.345	17.345	8483.932	8483.932	5575.339	572.753	6.806	6.806	482.115	84.794	31.762	96.443	96.443	69.688	2432.094	1701.382	1701.382

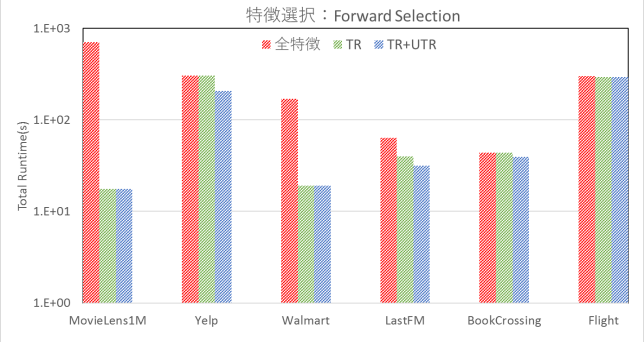

図 6　特徴選択に FFS を用いた時の全体の処理時間

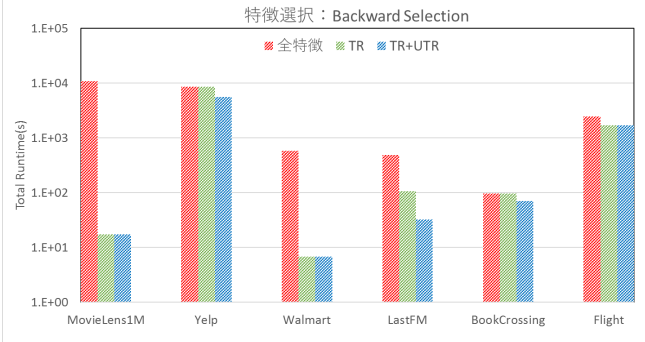

図 7　特徴選択に BFS を用いた時の全体の処理時間

については全アプローチにて同一の特徴群を選択したため等しい精度を示した．MovieLens1M, Walmart, LastFM, Flight については TR の判定条件を満たした時点で特徴量を削減した方が精度が向上している (特に LastFM の BFS)．これらは特徴量の過多によって特徴選択時に局所最適解に陥ってしまい，過学習を起こすことを回避しているためだと考えられる．また，LastFM の X_{R_2} については TR 適用時点で選択されず，一次元特徴変換しても選択されなかったため TR のみのアプローチと同一精度となっている．Yelp については全特徴量使用時に X_{R_2} の一部を選択しており，UTR の適用によって x'_{R_2} を X_{R_2} の一部の代替として選択したが，精度の変化は極小 (0.0001 の改善) に抑えられていることが確認できる．従って，学習モデルへの影響を抑えながら既存手法より多くのデータセットに対して特徴量削減可能なことを確認できた．

実行時間 表 3 に各処理に要した実行時間を示す．UTR による条件判定や一次元特徴変換に要する時間は 0.2s 以下と極小であり，FFS, BFS による特徴選択に要する時間より 100 倍以上高速なので，UTR が適用できない場合でも全体の処理時間にほとんど影響を与えないことが分かる．

図 6 に特徴選択に FFS を用いた時の判定条件確認から特徴選択終了までに要した処理時間，図 7 に特徴選択に FFS を用いた時の判定条件確認から特徴選択終了までに要した処理時間を示す．MovieLens1M, Walmart については TR の条件判定時点で R_1, R_2 ともに省略できていたので既存手法と同等の実行時間であり，全特徴量を使用する場合と比較して大きく高速化されている (FFS については MovieLens : 40.3 倍, Walmart : 8.9 倍, BFS については MovieLens : 625.3 倍, Walmart : 84.1 倍)．UTR の条件を満たした Yelp, LastFM, BookCrossing については全アプローチの中で処理を最短で実行できた．全特徴量を使用する場合と比較してそれぞれ FFS の時に 1.5 倍, 2.0 倍, 1.1 倍．BFS の時に 1.5 倍, 15.1 倍, 1.4 倍高速化した．Flight については UTR の条件判定を満たすことができなかったが UTR の条件判定に要した時間が極小なので既存手法とほぼ同一時間で実行できている．全特徴量使用時と比較して FFS では平均 9.0 倍, BFS では平均 121.2 倍高速化できており，データ分析を高速化可能なことを確認できた．

5. 関連研究

5.1 DBMS 分野と ML 分野における統合最適化

本論文で対象としている DBMS 分野と ML 分野における統合的な最適化手法は近年関心が高い研究分野である [16-25]．本節では実行処理系との統合によるモデル構築の高速化，宣言的言語を用いて実行プランを最適化する手法を提案している研究について述べる．

5.1.1 実行処理系最適化

DBMS 分野と ML 分野を組み合わせた実行処理系の最適化は数多く提案されている [15,16,23,24,25]．BlinkML [24] は学習データ数が精度に与える影響について定量化しており，一定の精度損失を許容する代わりに学習データ数を必要最小限にサンプリングすることで処理の高速化を実現する手法を提案している．M. Schleich [16] らは一般化線形モデルについて，モデル構築で行う計算の一部を DBMS 上

の処理に組み込むことでモデル構築時間を削減している．一方で，本研究では DBMS のスキーマ情報を活用して特徴量数を削減することで特徴選択及びモデル構築にかかる処理時間の高速化を実現している．

5.1.2 宣言的言語とプラン最適化による高速化

DBMS の宣言型言語とプラン最適化技術を用いて，特定の学習アルゴリズムに対して最適な実行プランを自動的に選択するフレームワーク (e.g., MLBase [18], TuPAQ [19], SystemML [20]) が提案されている．同様に MADlib[21] は PostgreSQL 上で ML アルゴリズムを実行できるライブラリを提供しており，DBMS 内でのデータ分析を可能にしている．これらは DBMS から抽出する学習データの出力結果は変わらないことを条件に処理順などの最適化を行うが，本研究では学習モデル精度に与える影響が小さいことを条件に学習データから抽出する特徴量数を自動的に削減することで処理を高速化している．

5.2 特徴抽出との統合

学習モデル構築の際には過学習を避けるために主成分分析や t-sne といった特徴量削減手法 [8,22] や特徴選択 [3,4,12,13] を用いることが一般的である．特に特徴選択は学習モデルの精度向上に大きく寄与するので，ML によるデータ分析を行う際に特徴量削減手法と組み合わせて広く用いられている．代表的な特徴選択の手法には本実験でも用いた FFS や BFS といったヒューリスティックな手法 [3,12] や Gini 係数や相互情報量を用いたフィルター法 [26] が存在する．本研究では DBMS 上のスキーマ情報を活用して精度への影響を抑えながら特徴量を事前に削減することで特徴選択の計算量を削減しているため，既存の特徴選択手法は，いずれの手法に対しても適用可能である．

6. まとめ

本論では，TR 指標を用いた結合省略による特徴量削減手法が適用できない条件下において，TR 指標より適用条件の広い UTR 指標が条件を満たす場合に，外部キーが参照するテーブルの全特徴量を一次元の特徴量に変換することで，学習モデルの精度への影響を抑えながら特徴量数を削減する手法を提案した．既存手法と同様の性能比較を行った結果，既存手法では条件を満たせずに全特徴量を利用する必要があった 7 つのテーブルのうち，提案手法では 3 つのテーブルを新たに特徴量削減が可能となり，学習モデルの精度を保ちながら特徴選択の高速化を達成した．

参考文献

[1] A. Kumar, J. Naughton, J. M. Patel, et al. To join or not to join?: Thinking twice about joins before feature selection. SIGMOD, pp. 19-34 (2016).

[2] V. Shah, A.kumar, X.Zhu. Are key-foreign key joins safe to avoid when learning high-capacity classifiers? PVLDB, pp. 366-379, (2017).

[3] C. Zhang, A.kumar, C.Re. Materialization Optimizations for Feature Selection Workloads. SIGMOD, pp. 265-276 (2014).

[4] I. Guyon, S. Gunn, M. Nikravesh,et al. Feature Extraction: Foundations and Applications. New York:Springer-Verlag (2001).

[5] T. Kraska, A.Talwalkar, J. Duchi,et al. MLbase: A Distributed Machine-learning System. CIDR, pages 2-1, (2013).

[6] A. Ghoting, R.Krishnamurthy, E.Pednault, et al. SystemML: Declarative Machine Learning on MapReduce. ICDE, pp. 231-242 (2011).

[7] J. Hellerstein, C.Re, F.Schoppmann, et al. The MADlib Analytics Library or MAD Skills, the SQL. VLDB, Vol. 5, No.12, pp.1700-1711 (2012).

[8] J. Shlens. A tutorial on principal component analysis. arXiv preprint arXiv:1404.1100 (2014).

[9] R. Ramakrishnan and J. Gehrke. Database Management Systems. McGraw-Hill, Inc. (2003).

[10] T. M. Mitchell. Machine Learning. McGraw Hill (1997).

[11] J. Pearl. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers Inc. (1988).

[12] R. Kohavi and G. H. John. Wrappers for Feature Subset Selection. Artif. Intell., Vol.97,pp. 273-324 (1997).

[13] S. A. Zadeh, M. Ghadiri, Vahab Mirrokni, et al. Scalable feature selection via distributed diversity maximization. AAAI, pp. 2876-2883 (2017).

[14] S. Shalev-Shwartz and S. Ben-David. Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press (2014).

[15] 鈴木 大慈. 統計的学習理論概説. 日本応用数理学会論文誌, Vol. 23, No.3, pp. 537-561, (2013).

[16] M. Schleich, D. Olteanu, and R. Ciucanu. Learning linear regression models over factorized joins. SIGMOD, pp. 3-18 (2016).

[17] A. Kumar, J. Naughton, and J. M. Patel. Learning generalized linear models over normalized data. SIGMOD, pp. 1969-1984 (2015).

[18] T. Kraska, A. Talwalkar, J. C. Duchi, et al. Mlbase: A distributed machine-learning system. In CIDR, 2013.

[19] E. R. Sparks, A. Talwalkar, D. Haas, et al. Automating model search for large scale machine learning. SoCC, pp. 368380, (2015).

[20] M. Boehm, M. W. Dusenberry, D. Eriksson, et al. Systemml: Declarative machine learning on spark. PVLDB, pp. 1425-1436, (2016).

[21] J. M. Hellerstein, C. Re, F. Schoppmann, et al. The madlib analytics library: or mad skills, the sql. PVLDB, pp. 1700-1711, (2012).

[22] L.J.P. van der Maaten, G.E. Hinton. Visualizing data using t-SNE. Journal of Machine Learning Research, pp. 2431-2456 (2008).

[23] C. Zhang, A. Kumar, and C. Re. Materialization optimizations for feature selection workloads. SIGMOD, pp. 265-276, (2014).

[24] Y. Park, J. Qing, X. Shen. BlinkML: Efficient Maximum Likelihood Estimation with Probabilistic Guarantees. SIGMOD, (2019).

[25] S. Idreos, K. Zoumpatianos, B. Hentschel, et al.The Data Calculator : Data Structure Design and Cost Synthesis from First Principles and Learned Cost Models. SIGMOD, pp.535-550, (2018).