Vol.2015-DBS-162 No.5
2015/11/26

分析者による試行錯誤を促進するデータ分析ツールの提案と試作

加藤 大志[1,a)]　三木 清一[1,b)]

概要：大規模データを分析して規則や知識を発見するデータマイニングが注目されている．データマイニングではしばしば機械学習の手法を用いるが，学習パラメータの入力や学習結果の評価において分析者の役割が重要となる．よりよい分析結果を得るには分析者が試行錯誤を重ねることが重要で，そのためには分析実行から結果評価のサイクルを速く回すことが求められる．本稿では，試行錯誤を促進するために，学習に計算時間がかかる場合でも実行中の途中結果を可視化できるデータ分析ツールを提案する．機械学習の EM アルゴリズムを用いた一つの手法を対象に，提案を具体化したプロトタイプシステムを開発した．本プロトタイプシステムでは，アルゴリズムの途中結果を確認しつつアルゴリズムの終了を待つことなく分析を再実行できるようになり，試行錯誤の促進が期待できることが分かった．

キーワード：データ分析，機械学習，インタラクション，可視化

Designing and developing an interactive data minig tool for rapid repeating trials

Daishi Kato[1,a)]　Miki Kiyokazu[1,b)]

Abstract: Data mining has got attention for finding rules and knowledge out of big data. Machine learning technique is often used in data mining, and the role of data analysts is important to design input parameters and evaluate output results. To get better results, trials and errors by analysts are important. Hence, a method for repeating design/evaluation cycles rapidly is desired. We propose an interactive data mining tool which allows to visualize intermediate results from a time-consuming algorithm We developed a prototype tool with an EM algorithm based machine learning algorithm. With this tool, users can observe the intermediate results and stop the algorithm for another trial if necessary.

Keywords: Data Analysis, Machine Learning, HCI, Visualization

1. はじめに

大規模データを分析して有用な情報を見つけ出すデータ分析が注目されている．データ分析にはしばしば機械学習の手法が用いられる．機械学習の手法を用いると将来を予測する規則や未知の知識を見つけ出すことができる．一方，機械学習の手法を用いた分析アルゴリズムの多くは探索的なアルゴリズムであり，多くの計算時間がかかる．場合によっては計算時間が数時間以上かかることもある．

ところで，データ分析においては分析者の役割が重要である．分析アルゴリズムは与えられた入力に対して最適 (もしくはそれに近い) な解を見つけることはできるが，その入力は分析者が与えなければならないからである．ここで分析アルゴリズムの入力とは，分析対象のデータ (データのどの範囲を学習に用いるかなどの情報も含む) と分析アルゴリズムのパラメータ (分析アルゴリズム毎に固有のもの) である．しかし，分析者が分析を実行する前に適切なデータとパラメータを用意することは難しい．そこで，分析者は，分析を実行し，その分析結果を吟味し，データやパラメータを調整することで，適切なデータやパラメータに近付けていくことになる．すなわち，データ分析にお

[1] 日本電気株式会社
NEC Corporation
[a)] daishi@cb.jp.nec.com
[b)] k-miki@bq.jp.nec.com

いては分析者の試行錯誤が重要である．

本稿が対象とする問題は，機械学習の手法を用いた分析アルゴリズムは多くの計算時間がかかり，試行錯誤に向かないという問題である．このような分析アルゴリズムでは計算時間に数分から数時間かかる場合があり，その間分析者は待たされることになり試行錯誤の効率が悪い．効率だけでなく，分析者の思考が中断されると本来ひらめいたかもしれない考えが出てこないといった問題が生じている可能性がある．

本稿では，多くの計算時間がかかる分析アルゴリズムを用いた場合でも，分析アルゴリズムの処理途中の状態を可視化することで分析者の試行錯誤を促進するデータ分析ツールを提案する．

2. 試行錯誤を促進するデータ分析ツール

機械学習の手法を用いた分析アルゴリズムの多くは，繰り返し型のアルゴリズムである．分析アルゴリズムの出力を「モデル」と呼ぶと，典型的な繰り返し型のアルゴリズムは，モデルをある指標で評価し，その指標が適当に収束するまでモデルの改良を繰り返すという動作をする．

本稿では分析アルゴリズムの例として機械学習でしばしば用いられる EM アルゴリズムに関して議論する．EM アルゴリズムは，潜在変数を持つ確率モデルを最尤推定するアルゴリズムであり，繰り返し型である．他の繰り返し型アルゴリズムとしては，SVM やクラスタリングやニューラルネットワークなど様々なアルゴリズムがあり，本稿の議論が同様に適用できる．

本稿では分析アルゴリズムをソウトウエアとして実装したものを「分析エンジン」と呼ぶ．EM アルゴリズムを実装した分析エンジンは通常は収束した後にモデルを出力するが，内部的には 1 組の EM ステップ毎に更新されたモデルを途中状態として保持している．そのため，途中状態のモデルを EM ステップ毎に出力することは容易である．

提案するデータ分析ツールは，分析エンジンの実行中 (すなわち収束するまでの間) に途中状態のモデルを可視化する．その結果，分析者は分析エンジンの実行終了を待たなくして情報を得て，「学ぶ」ことができる．具体的には，どのような変数が特徴的かなどデータそのものに対する理解やモデルの収束のパターンなどアルゴリズムの動作に関する理解が得られることが期待できる．

分析者がデータやアルゴリズムについて学ぶことができると，データの加工やアルゴリズムパラメータの調整を行いやすくなり試行錯誤につながるという効果がある．また，パラメータを変更して分析を再実行する場合には，実行中の分析エンジンを中断することもできる．その結果，時間あたりの試行錯誤の回数を増やせるという効果がある．

図 1 に提案するデータ分析ツールのシステム構成を示す．分析者は分析エンジンを対話的に操作することになり，本

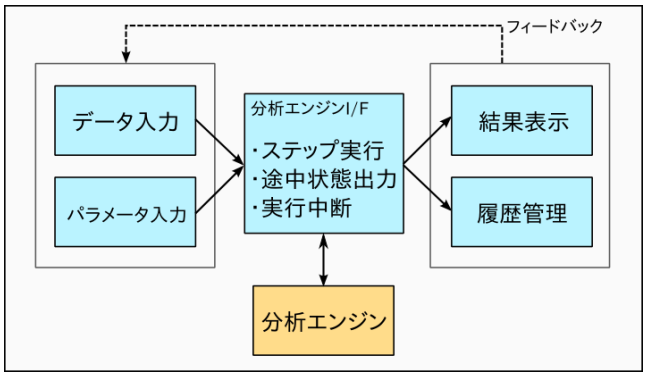

図 1　システム構成
Fig. 1　System overview

システムは一種のインタラクティブシステムである．図 1 の左側が分析エンジンへの入力に対応し，右側が出力に対応する．分析エンジンへの入力は分析対象のデータおよび分析アルゴリズムのパラメータである．中央の分析エンジン I/F はこれらの入力を用いて分析エンジンをステップ実行し，途中状態を出力する．分析エンジンが出力する途中状態のモデルは終了 (収束) 状態のモデルと同一形式である．そのため，モデルを表示する「結果表示」の機能は共通である．また，モデルを含む分析の結果は「履歴管理」する．図の上部の破線矢印は分析者が結果や履歴から得た知識をフィードバックして入力を変更し，再度分析エンジンを実行することを示す．この際，分析エンジンが実行中であっても中断し，新しい入力で再実行することができる．

3. FAB/HME を対象としたプロトタイプシステム

前節で述べたデータ分析ツールおよびそのシステム構成の実現可能性を検討するために，実際に動作するプロトタイプシステムを開発した．本節では，具体的な実装方法と機能を紹介し，最後に考察を述べる．

3.1 FAB/HME

本プロトタイプシステムでは，分析アルゴリズムとして FAB/HME [4] を採用した．FAB/HME は，混合モデルにおける正則化条件付きの線形回帰モデルを効率的に学習するアルゴリズムであり，EM アルゴリズムの一種である．学習するモデルは，一つの決定木と複数の線形回帰式で表現される．決定木は各中間ノードが一つの変数としきい値を分岐条件としデータを分割するもので，線形回帰式は分割されたのデータそれぞれについて目的変数を予測するものである．

本アルゴリズムが出力するモデルは一つの決定木と複数の疎な線形回帰式で表現されるため，分析者にとってモデルの解釈がしやすいという特徴がある．分析者がモデルを解釈できると分析が期待通りに進んでいるかを判断しやす

いため，試行錯誤に向いている．

3.2 実装方法

本プロトタイプシステムは Web アプリケーションシステムとして実装した．Web システムとすることで，クライアント側は任意の Web ブラウザを使うことができる．具体的に実装に使用したライブラリについて説明する．

サーバ側は Python で実装し FAB/HME の実装と連携した．Web サーバ機能は Python の twisted [*1] を用いて実装した．Web サーバとクライアント間の通信は，ファイル転送を除き sockjs [*2] を利用した．sockjs は WebSocket [5] のラッパーである．WebSocket を用いることにより分析エンジンの出力を遅延なくクライアント側に送信することができる．

クライアント側はフレームワークとして AngularJS [*3] を利用し，sockjs の通信による分析エンジンとの入力・出力を操作・表示する UI を実装した．これに加えて，各ライブラリを選定した．ライブラリの選定には，Web ブラウザで可能な限り大きなデータを扱えることを条件とした．分析対象のデータの表示および編集には，スプレッドシートの機能を提供する handsontable [*4] を利用した．handsontable は 100 万行までのデータを扱えると述べられている．また，出力結果のチャート表示には，時系列データの可視化を提供する dygraphs [*5]，散布図を含む様々なデータの可視化を提供する echarts [*6] を利用した．dygraphs は数百万のデータプロットが行えると述べられている．echarts も Big Data Mode で 20 万までのデータプロットが行えると述べられている．

3.3 構成機能

開発したプロトタイプシステムの機能について説明する．

Web ブラウザを開いてアプリケーションの URL を開くとデータアップロード画面が表示される．ファイル選択ダイアログからローカルの ARFF [*7] ファイルを選択してアップロードするとメインの画面に切り替わり分析が開始する．メインの画面は，データ，パラメータ，モデル (ツリービュー)，モデル (テーブルビュー)，チャート，実行履歴の 6 つのエリアから構成される．これらのエリアのについてそれぞれ説明する．

データエリア (図 2) は分析対象のデータをスプレッドシート形式で表示・編集する機能である．初めにどの変数 (スプレッドシート上の列に相当) を目的変数とするかを指定する．また本エリアでは，分析エンジンに入力するデータを選択できる．データ選択は「学習データ」「テストデータ」「両方」「使用しない」から選択でき，行番号範囲での設定や行毎の設定が可能である．必要であれば，handsontable のオプションで有効にすることでデータの各セルを手動で修正することも可能になる．

パラメータエリア (図 3) は FAB/HME のパラメータを設定する機能である．マウスによるドロップダウンリストや，属性のドラッグ&ドロップでパラメータの値を操作し，「再計算」ボタンを押すことで分析エンジンの実行を開始できる．パラメータとして実数を受け付けるような場合でも，典型的に用いられる数値をドロップダウンリストから選べるようにした．これにより，分析者はキーボードを用いずにマウスで簡単に操作できるだけでなく，リストの値に付記されている説明を参考にして値の意味を理解して選択できる．パラメータの値が自明で説明不要な場合はスライドバーを使うこともできる．

モデルの可視化はユーザインタフェース上最も重要な機能であり，本プロトタイプでは 2 種類用意した．モデル (ツリービュー) エリア (図 4) は FAB/HME が出力したモデルの決定木を主体として全体像を分かりやすく可視化する機能である．木構造を直接図示することでデータの分岐条件を把握しやすいという特徴がある．一方，モデル (テーブルビュー) エリア (図 5) はモデルの線形回帰式 (予測式) を中心に詳細に可視化する機能である．決定木 (選択基準) によって分割された予測式の全ての変数の係数，すなわちどの変数がどの程度寄与しているかを確認しやすいという特徴がある．

チャートエリア (図 6) は FAB/HME が出力したモデルによってテストデータを予測した結果を表示する機能である．一つのチャート (図 6 左) はデータ毎 (横軸) に目的変数の値 (縦軸) の実測値と予測値を比較表示する．このチャートでどのデータの予測がどれくらいずれているかが視覚的に分かる．通常，誤差率などの数値のみで比較することに比べて，このチャートを用いると全体の傾向だけでなくズーム機能を使って個別のデータのずれも詳細に確認することができる．もう一つのチャート (図 6 右) はデータ毎にプロットした散布図で，横軸に目的変数の実測値をとり，縦軸に目的変数の予測値をとる．また，プロット毎にどの予測式による予測であるかを色と記号で区別して表示している．このチャートも同様に予測のずれが視覚的に分かり，特に外れ値が見つけやすい．データ点をマウスでクリックすると，テーブルエリアの対応する行がハイライト表示されデータ要素を確認することができる．このチャートもズーム機能を備えているため，データ点が多い場合でも特徴的な部分を拡大して詳細に見ることができる．

実行履歴エリア (図 7) は分析エンジンを実行した履歴を表示する機能である．各履歴には重要な指標 (FIC [4],

[*1] https://twistedmatrix.com/
[*2] http://sockjs.org/
[*3] https://angularjs.org/
[*4] http://handsontable.com/
[*5] http://dygraphs.com/
[*6] http://ecomfe.github.io/echarts/index-en.html
[*7] http://weka.wikispaces.com/ARFF

図 2　データエリアの画面例
Fig. 2　Screenshot of data table

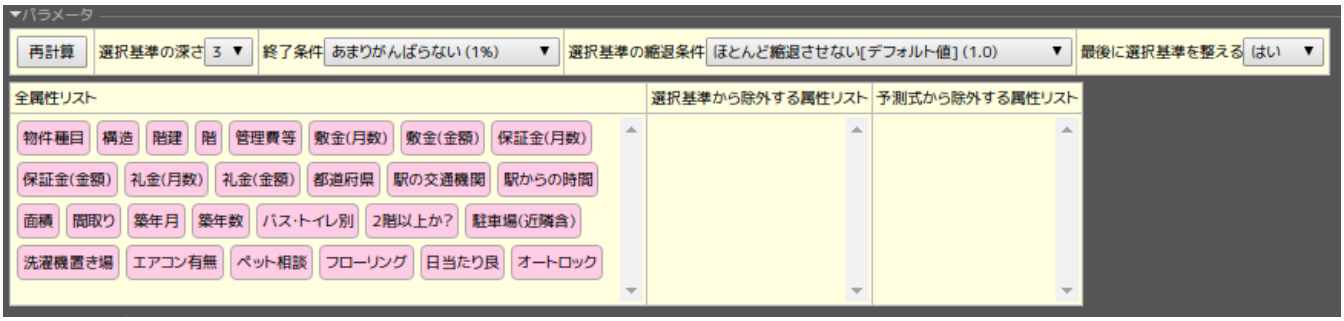

図 3　パラメータエリアの画面例
Fig. 3　Screenshot of prameters

RMSE=Root Mean Squared Error, MAE=Mean Absolute Error) が表示され，マウスで指定することでこれらの指標のステップ毎の変化をチャートに表示することができる．分析者が試行錯誤を重ねた結果，以前の実行を再現したい場合には履歴毎に設置されているボタンをクリックすることで，その時の用いたパラメータを再度読み込むことができる．

3.4　考察

分析者がどの程度分析エンジンを対話的に操作できるか，実際の分析動作を例に説明する．

分析対象のデータは Census database [*8] を加工したもので，属性数は 134，データ数は 22784 である．このデータを典型的なパラメータで分析実行したところ，全体で 176 秒かかった．この分析は 65 ステップで終了した．各ステップの実行時間は多少のばらつきはあるもののおよそ一定であり，すなわち 1 ステップにかかる平均時間は約 2.7 秒である．画面のモデルエリアはステップ毎に更新されるため約 2.7 秒で表示が変わるが，フェードイン，フェードアウト，ハイライトのアニメーション効果が 1 秒あるため，表示が変化しない状態は約 1.7 秒である．

176 秒と比較すると 1.7 秒は大幅に短いため，分析者は画面を見つつ次の試行について考えることができる．例えば，モデルで使われる変数に矛盾するような属性が選択されてしまった場合に，その属性を除外するようにパラメータを変更するなどができる．さらに，その場合に分析が終了していない場合は中断して新しいパラメータでの分析を即座に実行することができる．実際に本データで実行した場合でも，176 秒待つことは少なく，早い段階で決定木や回帰式に使われる変数がほとんど変化しなくなったため，途中で中断してパラメータを変更して再実行することが多かった．最終的に適切なパラメータを決定し，十分な時間をかけて分析を最後まで実行することで目的とするモデルを得る．

分析エンジンの途中状態を可視化することで，分析者のデータやパラメータに関する理解が進み，これまで気づかなかった視点での分析を設計することも期待できる．例えば，途中状態でモデルに表われていた変数がその後に消え

[*8] http://www.cs.toronto.edu/%7edelve/data/census-house/censusDetail.html

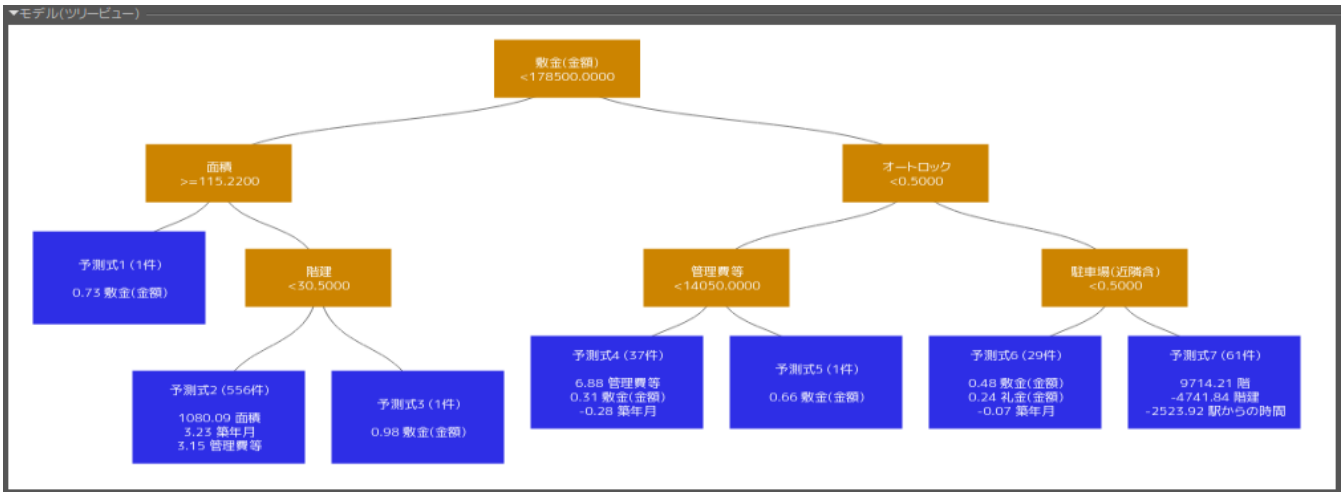

図 4　モデル (ツリービュー) エリアの画面例
Fig. 4　Screenshot of model tree

図 5　モデル (テーブルビュー) エリアの画面例
Fig. 5　Screenshot of model table

てしまうケースでその変数が分析の意味において重要である場合に，その変数をモデルに組み込むような制約を追加するなどがある．さらに，分析エンジンを中断・再実行することで，試行錯誤に全体にかかる時間を減らす効果が期待できる．

3.5　今後の拡張について

本プロトタイプシステムでは実行履歴機能は副次的であったが，履歴から知識を得ることは重要であり，今後拡張すべき機能である．

試行錯誤においては，過去に試行した結果から一番よいものを選んだり，それをさらに修正したりすることがある．手軽に試行ができると試行の数は膨大になるのため，履歴の管理が重要となる．

本プロトタイプシステムの実行履歴機能は，試行の履歴を記録し，パラメータを読み込むだけの機能であったが，少なくとも何らかの指標順に並べ替える仕組みは必要だろう．また，一つまたは複数の履歴から新しい試行のためのパラメータを容易に生成する仕組みも望まれる．このような仕組みを実現する履歴機能は自明ではなく，試行錯誤を促進するための履歴機能は今後の課題である．

4.　関連研究

本稿で提案したデータ分析ツールのようにインタラクティブシステムを提供している研究について述べる．以下に示すように様々な取り組みが行われており，データマイニングにおけるインタラクティブ性は重要視されている．一方で，EM アルゴリズムなどの繰り返し型アルゴリズムにおいて途中状態を可視化することでインタラクティブ性を実現する取り組みはない．

iPCA [6] は，主成分分析 (PCA) をインタラクティブに行うツールである．PCA のアルゴリズムは繰り返し型ではなく，本稿が対象とした計算時間が多くかかるという問題はない．

Jiang ら [7] は，様々なクラスタリングをインタラクティブに実行するツールを開発した．実行速度を上げるために，BIDMach [3] と呼ばれる GPU を最大限に利用するツールキットを用いており，本稿が対象とした計算時間が多くかかるという問題を回避している．

VINeM [1] [*9] は，k 近傍法などの計算を用いて手動でクラスタリングを行うツールである．k 近傍法は繰り返し型

[*9] http://adrem.uantwerpen.be/vinem

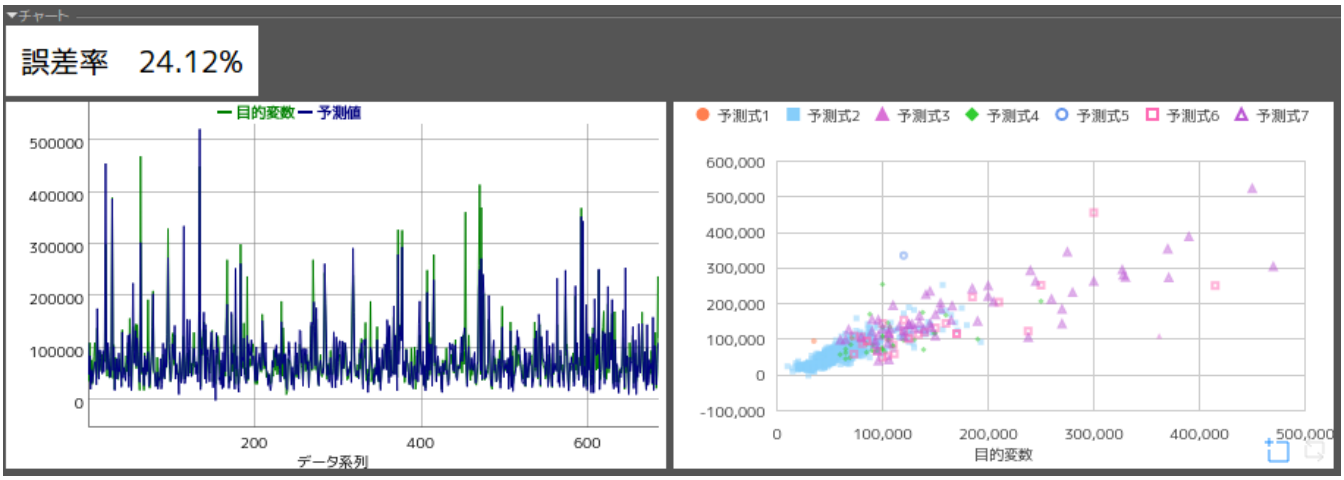

図 6　チャートエリアの画面例
Fig. 6　Screenshot of charts

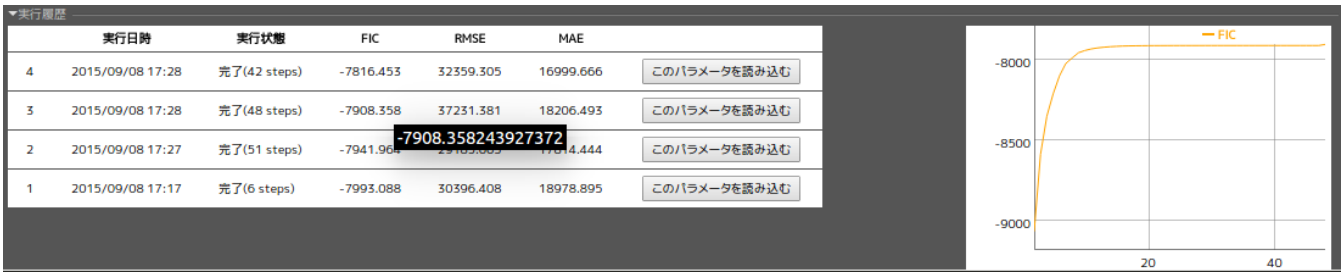

図 7　実行履歴エリアの画面例
Fig. 7　Screenshot of history

のアルゴリズムではなく，本稿が対象とした計算時間が多くかかるという問題はない．

GGvis [2] は，多次元尺度構成法 (MDS) をインタラクティブに操作できるようにするツールである．MDS のアルゴリズムは繰り返し型ではなく，本稿が対象とした計算時間が多くかかるという問題はない．

5. おわりに

本稿では，機械学習の手法を用いたデータ分析において分析者が試行錯誤を行いやすくするためのデータ分析ツールを提案した．多くの計算時間がかかる分析アルゴリズムの場合にステップ毎の途中状態を可視化することが特徴である．プロトタイプシステムの開発により動作を確認し，試行錯誤の促進が期待できることが分かった．

今後の課題としては，より実際的な評価を行うこと，他の分析アルゴリズムを対象とすることなどがある．

参考文献

[1] Aksehirli, E., Goethals, B. and Müller, E.: Visual Interactive Neighborhood Mining on High Dimensional Data, *Proceedings of the KDD 2015 workshop on Interactive Data Exploration and Analytics (IDEA)*, ACM (2015).
[2] Buja, A., Swayne, D. F., Littman, M. L., Dean, N., Hofmann, H. and Chen, L.: Data visualization with multidimensional scaling, *Journal of Computational and Graphical Statistics* (2008).
[3] Canny, J. and Zhao, H.: Big Data Analytics with Small Footprint: Squaring the Cloud, *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, New York, NY, USA, ACM, pp. 95–103 (online), DOI: 10.1145/2487575.2487677 (2013).
[4] Eto, R., Fujimaki, R., Morinaga, S. and Tamano, H.: Fully-Automatic Bayesian Piecewise Sparse Linear Models, *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, AISTATS 2014, Reykjavik, Iceland, April 22-25, 2014*, JMLR Proceedings, Vol. 33, JMLR.org, pp. 238–246 (online), available from ⟨http://jmlr.org/proceedings/papers/v33/eto14.html⟩ (2014).
[5] Fette, I. and Melnikov, A.: The WebSocket Protocol, RFC 6455 (Proposed Standard) (2011).
[6] Jeong, D. H., Ziemkiewicz, C., Fisher, B., Ribarsky, W. and Chang, R.: iPCA: An Interactive System for PCA-based Visual Analytics, *Proceedings of the 11th Eurographics / IEEE - VGTC Conference on Visualization*, EuroVis'09, Chichester, UK, The Eurographs Association & John Wiley & Sons, Ltd., pp. 767–774 (online), DOI: 10.1111/j.1467-8659.2009.01475.x (2009).
[7] Jiang, B. and Canny, J.: Interactive Clustering with a High-Performance ML Toolkit, *Proceedings of the KDD 2015 workshop on Interactive Data Exploration and Analytics (IDEA)*, ACM (2015).