2 データスカッシング
ー逆転の発想によるスケールダウン戦略ー

鈴木 英之進（横浜国立大学 大学院工学研究院）
suzuki@ynu.ac.jp

■データスカッシングとは？

できそうもない目標を達成するためには，逆転の発想が役に立つことがある．機械学習 (machine learning) は，学習という行為を通して性能を向上するソフトウェアを研究する分野であり，開発されているアルゴリズムは例 (＝事例，サンプル) 数が高々数万程度と仮定している．データマイニング (data mining) は，大量情報からの有用知識の発見を目的とし，例数がはるかに大きなデータを対象とする．多くの研究者は，このギャップを，機械学習アルゴリズムを大規模データに対応できるように改良するスケールアップ戦略で切りぬけようとした☆1．ところが，これとは逆に，大規模データを“つぶす” (squash) ことによって小さくし，機械学習アルゴリズムをそのまま使うスケールダウン戦略をとる研究者たちが現れた[2)]．この場合は，大規模データを適切に小さくするアルゴリズムが，データマイニング手法である．データスカッシングの動機を図-1に示す．

データスカッシング (data squashing) は，知識を高速に発見するために巨大なデータを適切に小さくする処理を表す．レモンをそのまま丸ごと食べるのは難しいが，レモネードにすればだいぶ楽になる☆2．同様に，巨大データをそのまますべて処理するのは難しいが，適切な形式の小さなデータに変換すればだいぶ楽になる．これは通常，データの一部を選択するサンプリングよりも高度な作業であり，例分布の推定である密度推定 (density estimation) と関連する．

■BIRCH：大量データの高速クラスタリング

銀行の顧客データから，年齢，資産額，および年収などのデータを抜き出し，各々に関してヒストグラムを描いてみる．顧客数が十分大きければ，年齢に関する分布は1つの山になるだろうが，資産額に関する分布や年収に関する分布は複数個の山になると予想される．これは，取り得る値の種類が多いという理由だけではなく，“富は偏在する”という性質に基づくためだと考えられる．この性質により，年齢，資産額，および年収に基づき，顧客たち$e_1, e_2, \ldots, e_n$を“似たもの同士”☆3に分けてみると，いくつかの“かたまり”すなわちクラスタ (cluster) $c_1, c_2, \ldots, c_m$に分かれそうである．ただし$c_1, c_2, \ldots, c_m$は$e_1, e_2, \ldots, e_n$の分割となっている．各クラスタc_iは典型的な顧客像に対応すると予想されるため，優良顧客の種類を知ったり，見過ごされがちだが注意が必要な顧客像を見つけることに役立ちそうである☆4．クラスタリングは，データを似たもの同士であるクラスタに分けるこ

☆1 機械学習アルゴリズムの中には，分類・回帰学習手法など，データマイニングのために使えるものが多い．
☆2 cf. レモンスカッシュ (lemon squash)．
☆3 2人の顧客が似ている度合は，目的に応じて与える必要がある．
☆4 クラスタの解釈は，最終的にはユーザに任されている．

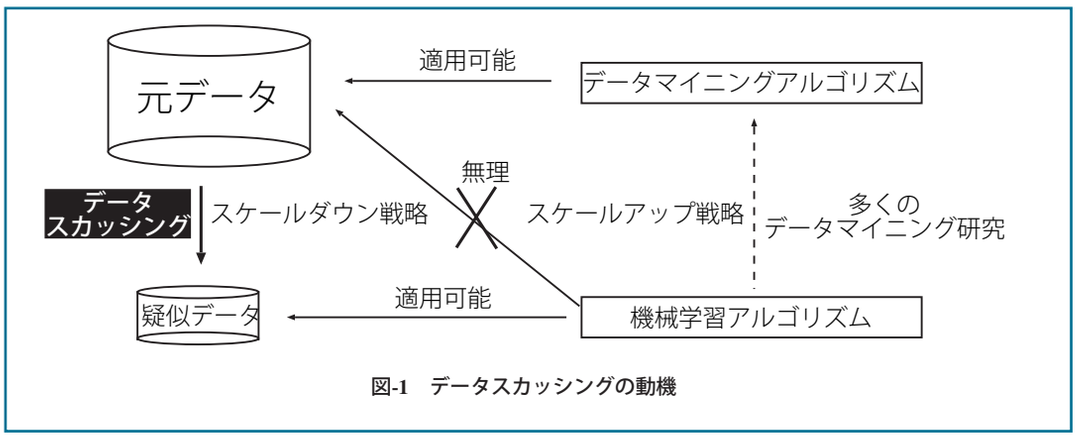

図-1　データスカッシングの動機

とにより，データをよりよく理解するための情報を得る学習・発見手法である．

CFベクトルを用いたクラスタリング

実質的には最初のデータスカッシング手法と言われるBIRCH (Balanced Iterative Reducing and Clustering using Hierarchies)[9] を紹介する．BIRCHは，数値属性で表される例集合$X=\{x_1, x_2, \ldots, x_n\}$をCF (clustering feature) ベクトルの集合$V=\{v_1, v_2, \ldots, v_m\}$に変換し，CFベクトルの集合をクラスタリングする．各CFベクトルvは，距離が近い複数個の例をまとめた抽象表現に相当し，まとめた例が$x'_1, x'_2, \ldots, x'_p$ (ただし$x'_i \in X$) の場合，(例数，線形和，2乗和) で表される．

$$v = \left(p, \sum_{i=1}^{p} x'_i, \sum_{i=1}^{p} \| x'_i \|^2 \right)$$

たとえば$x_1=(1, 1)$, $x_2=(2, 1)$, $x_3=(1, 2)$ をまとめたCFベクトルは$v_1=(3, (4, 4), 12)$ であり，同様に$x_4=(3, 3)$, $x_5=(4, 3)$, $x_6=(4, 4)$ をまとめたCFベクトルは$v_2=(3, (11, 10), 75)$ である．

正確なクラスタリングを行うためには，元の例集合Xが必要であるが，近似的な計算ならCFベクトルの集合Vで十分な場合もある．一部のクラスタリングアルゴリズムは，平均クラスタ間距離 (average inter-cluster distance) などの，2つの例集合間の距離に基づく．片方の例集合E_1に属す例を$x_1, x_2, \ldots, x_{n_1}$, もう片方の例集合$E_2$に属す例を$x_{n_1+1}, x_{n_1+2}, \ldots, x_{n_1+n_2}$とおくと，平均クラスタ間距離$D_2(E_1, E_2)$ は次で与えられる．

$$D_2(E_1, E_2) \equiv \sqrt{\frac{\sum_{i=1}^{n_1} \sum_{j=n_1+1}^{n_1+n_2} \| x_i - x_j \|^2}{n_1 n_2}}$$

たとえば，上記のv_1に該当する例集合と，v_2に該当する例集合間の平均クラスタ間距離は，$\sqrt{31/3}$ となる．ここで，この値は，元の例集合を用いなくても2つのCFベクトルだけから計算できることが分かる．実際，平均クラスタ間距離をはじめとする種々の例集合間の距離は，該当するCFベクトルだけから計算できることが分かっている．さらに，CFベクトルは加法性を満たすため，ハードディスクから例を逐次読みながらCFベクトルとして表されたクラスタ情報を効率的に更新できるという特長を持つ．

アルゴリズムと性能

BIRCHが対象とするデータは通常，巨大すぎてメモリに収まらず，ディスクに保存されている．ディスクアクセスはメモリアクセスに比較して10^5～10^6倍遅い上，いくつかの制限がある．BIRCHはディスク上のデータを1スキャンしてCFベクトルの集合に変換してメモリに収め，メモリ上のCFベクトル集合をクラスタリングすることでこの問題に対処している．**図-2**にBIRCHアルゴリズムの概要を示す．BIRCHが，CF木というデータ構造を用いることや，よりよいクラスタリングを行うためにいくつかのオプションを用意していることなどが分かる．

BIRCHではデータスカッシング中，CFベクトルの数が増えすぎて，メモリに収まりきらない場合が起こり得る上，多数のCFベクトルを効率的に更新する必要がある☆5．高さ平衡木 (height balanced tree) は，子ノー

☆5 CFベクトルから元の例を復元することはできないので，1個のCFベクトルにまとめる例は多すぎない方がよい．データスカッシング自体も高速である必要があるため，CFベクトルの作成にクラスタリングなどの計算時間がかかる方法は採用できない．

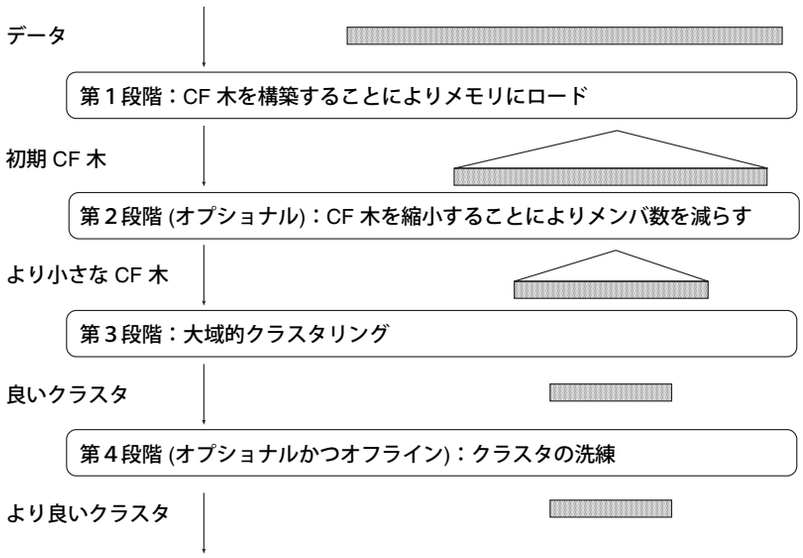

図-2　BIRCHアルゴリズムの概要

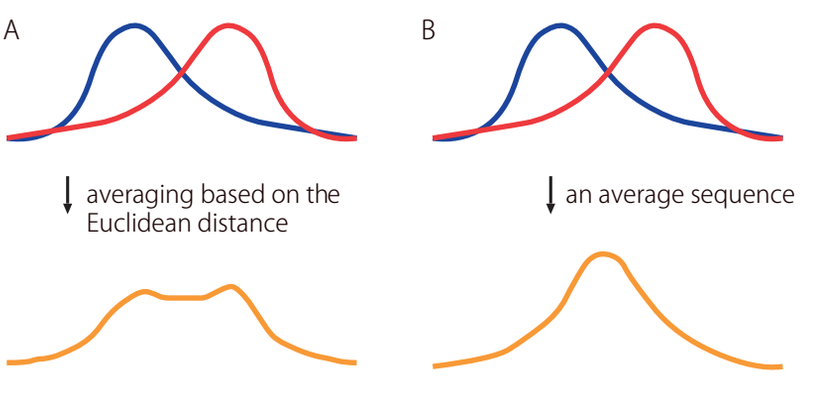

図-3　時系列データの「平均化」

ド数を調整することにより木の高さが常に一定値となるデータ構造であり，B+ 木などがよく知られている．BIRCHでは，CFベクトルを管理する高さ平衡木であるCF木を用いており，データスカッシングを行う基準となる距離に関する閾値Θを適切に増加することと併せて，上記の問題に対処している．そのほか，他の例と極端に離れた異常値をノイズと見なしてデータスカッシングの対象としない機能や，いったん構築したCF木を洗練する機能，および指定されれば最後にもう一度ディスク上のデータをスキャンしてクラスタリング結果を洗練する機能などを提供している．

BIRCH は高速であり，CF木の葉数が定数と見なせ図 -2の第4段階を用いないなら，時間計算量は$O(n)$である．実際，人工データを用いた実験結果においても，図-2の第4段階を用いないなら，実行時間の増加は線形に抑えられていた．Zhangらは各ピクセルを例と見なしBIRCHを実画像のフィルタとしても用いている．画像の表現を適切に設定した場合，BIRCHは樹木の画像から日に照らされた葉，枝，および木の影を抽出することに成功した．一般論として，BIRCH のような距離に基づくクラスタリングは，通常低次元のデータ集合に用いられる．これは次元数が増えると例空間は疎となり，学習が失敗する高次元の呪いが問題となるからである．BIRCH は低次元だが例数がきわめて多いデータ集合に有効であると考えられる．

BIRCHの拡張

BIRCHをWebアクセスログのクラスタリングに用いようとすると，クラスタリングに有用な属性を選択する機能や，一定間隔ごとにデータスカッシングを行う機能なども必要となる．奈良橋らは，これらの機能を満たすアルゴリズムを開発し，Webアクセスログからの侵入発見問題に成功裏に適用している[6]．

BIRCHは，例がいくつかの数値属性で記述されると仮定しているが，これらの数値属性間に構造がある場合には特別な手法が有利であると考えられる．中本らは，例が時系列データである場合のためのデータスカッシング法を考案し，比較的正確なクラスタリングができることを実験で示した[5]．彼らの手法では，図-3 (B) に示す時系列データの「平均化」手法を用いて，疑似的な時系列データを生成している．

■分類・回帰学習のためのデータスカッシング

さきほどの銀行顧客データには，口座を解約して他銀行に移ったか否かや，資産額など，重要な属性が存在する．顧客の中には，これらの重要な属性の値が空欄となっている者がいる．値が分かっている他の属性の値から，

重要な属性の値を正確に求められる関数があれば，重宝しそうである．分類学習と回帰学習は，与えられたデータからこのような関数を求める学習であり，それぞれ重要な属性が名目属性☆6の場合と数値属性の場合に相当する．分類学習と回帰学習において，重要な属性をそれぞれクラス，目的変数と呼び，上記の関数をそれぞれ分類モデル，回帰モデルと呼ぶ．

データスカッシング

“データスカッシング”という用語を初めて提唱したのはAT & T 研究所のDuMouchelらである[2]．彼らの主な動機は，データ要約とランダムサンプリングの良いところを合わせて元データから小さい疑似データを生成し，疑似データから分類・回帰学習を行うことで，元データを用いた場合に匹敵する正確な結果を高速に得ることである．大雑把な議論では，1% のランダムサンプルから推定された統計値は，元データから推定された統計値に比較して10標準偏差の誤差があるが，データスカッシングではこの誤差を1標準偏差に抑えることを目標としている．その根拠として，スカッシングされたデータに重みがついていれば，次節で説明する尤度に関するテイラー級数の最初の数項から正確な回帰モデルを得られるという理論的保証があげられる[2]．なお前章のBIRCHでは，属性は数値属性だけと仮定していたが，DuMouchelらの手法は名目属性を含むデータにも適用できる．

この手法は，グループ化，モーメント化，および疑似例化の3手順から構成される．グループ化においては，名目属性は値ごとに分け，数値属性は四分位数 (quantile) やデータ球などを利用することで，例集合をいくつかのグループに分割する．データ球は，前章で述べた次元の呪いに対処するため，例空間を領域の大きさではなく例数に基づき分割する手法である．モーメント化においては，各グループに対して，テイラー級数近似における望ましい次数を決め，モーメントを計算する．DuMouchelらは望ましい次数として，$\max(1, \lceil \alpha \log_2 p \rceil)$ を提案している．ただしαとpはそれぞれユーザが与えるパラメータ (0.5, 1, 2 など) とスカッシングされた例数を表す．疑似例化においては，各グループに対して，モーメントがほぼ同じとなるようにスカッシングされたデータを生成する．ここでは，2乗誤差が小さい重み付きの例集合を探索で求めている．

DuMouchelらが勤めているAT & Tでは，1999年当時においてさえ，数億個から数十億個の例から構成されるデータを日常的に処理していた．彼らは，データスカッシング手法の効果を調べるため，顧客が他の長距離通信会社に乗り換える確率$P(Y=1)$を予測する回帰学習問題を選んだ．データ集合は，元データからの学習が可能となるように比較的小さく，約75万例から構成されており，各例は目的変数のほかに2個の名目属性と5個の数値属性によって記述されていた．学習手法としては，ロジスティック回帰を用いた．この学習手法では名目属性をダミー変数に変換するため，属性は$X_1, X_2, \ldots, X_9$ の9個となった．回帰モデルはロジスティックモデルであり，

$$P(Y=1) = \frac{1}{1+\exp(-Z)}$$

$$\text{where } Z = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_9 X_9$$

で表される．

まず，疑似データから得られたロジスティックモデルが，元データから得られたロジスティックモデルと似ている度合を，ロジスティックモデルの係数$\beta_0, \beta_1, \ldots, \beta_9$に関する標準化された2乗誤差で測った．実験の結果，データスカッシング手法が1% ランダムサンプリングよりもはるかに正確な回帰モデルを生成できることが分かった．前者で推定した係数の誤差はほぼすべて1標準偏差以内であり，後者の大多数が5標準偏差よりも大きかった．

次に，疑似データから得られたロジスティックモデルの出力が，元データから得られたロジスティックモデルの出力と似ている度合を，2つの回帰モデルが出力する$P(Y=1)$ の差に関する分布で測った．実験の結果，グループ化において四分位数を用いるデータスカッシング手法は，元データに適用した場合に匹敵するほど正確であることが分かった．さらに$\alpha=0$として各グループを中心例で置き換える単純手法は，ランダムサンプルよりも不正確であり，データスカッシングにおいて各種統計値を考慮する正当性が示された．グループ化においてデータ球を用いる場合，$\alpha=2$が$\alpha=1$よりも不正確であり，より詳しく要約すればよいというわけではないことも実験で示された．

尤度に基づくデータスカッシング

統計学では，データDがパラメータθで指定される統計モデルに従って生成されたと仮定し，Dの生成確率で

☆6 値に順序がない属性

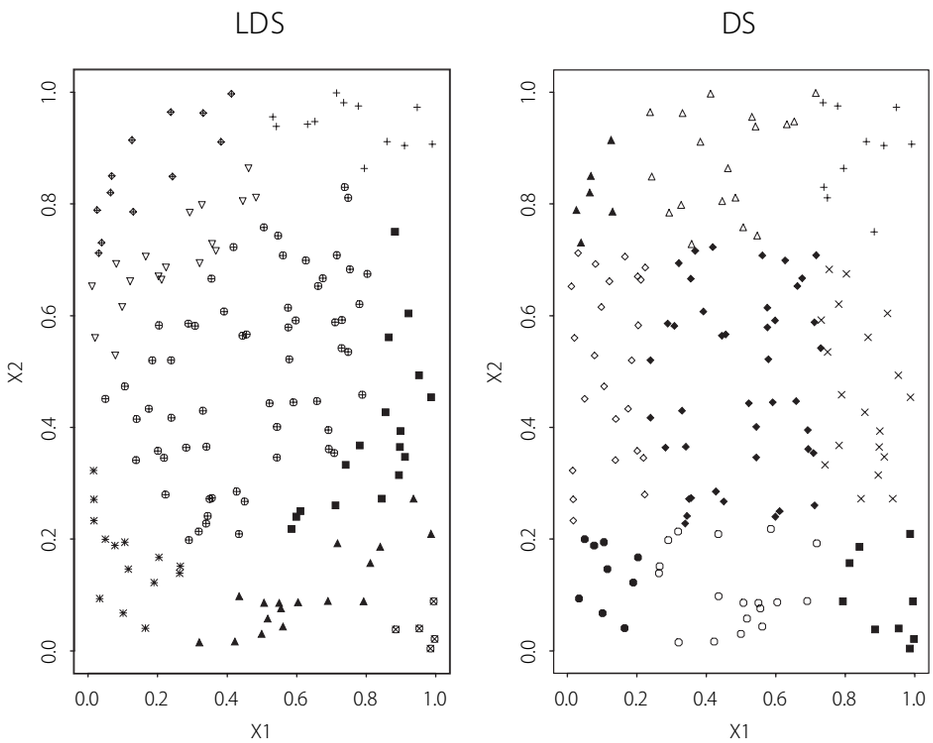

図-4　尤度に基づくデータスカッシングによるグループ化とDuMouchelらのデータスカッシングによるグループ化の例 (David Madigan 氏の許可を得て使用)

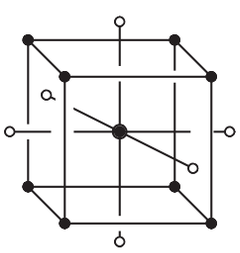

図-5　m=3の時の中心例, 星例, および階乗例 (David Madigan 氏の許可を得て使用)

ある尤度 (likelihood) $P(D|\theta)$ に基づいて議論を進める場合がある．たとえば最尤法は，$P(D|\theta)$ が最大となる$\hat{\theta}$で指定される統計モデルを最良と見なす評価基準である．この考え方からすれば，DuMouchel らの手法は，グループ化された例集合において，尤度がテイラー級数の最初の数項で近似できる場合には有効であると考えられる．もっとも，実験ではロジスティック回帰だけを用いており，他の学習手法での有効性は示されていない．

Madigan らは，尤度に基づくデータスカッシング (Likelihood-based Data Squashing: LDS) を提案した[4]．この手法は，DuMouchel らのデータスカッシングと概要は似ているが，グループ化と疑似例化の方法が異なる．その結果，**図-4**に示すように，両者のグループ化の結果が異なる場合がある．実験の結果，LDS がより正確であり，統計モデルが学習アルゴリズムと異なる場合でも性能が良いことが示されている．

LDS では，例の重みは尤度が似ている例数と見なせる．たとえば$D=\{e_1, e_2, e_3\}$ のとき，$P(\theta \mid D) \propto P(e_1 \mid \theta)\, P(e_2 \mid \theta)\, P(e_3 \mid \theta)\, P(\theta)$ である．ここで$P(e_1 \mid \theta) \approx P(e_2 \mid \theta)$ であれば，e_1とe_2に対応する疑似例e^*を生成し，その重みを2とする．すなわち，$P(\theta \mid D) \propto P(e^* \mid \theta)^2\, P(e_3 \mid \theta)\, P(\theta)$ と近似する．実際には，$P(e_i \mid \theta)$ をθに関するk個の値$\{\theta_1, \theta_2, \ldots, \theta_k\}$ について求め，尤度プロファイル $(P(e_i \mid \theta_1), P(e_i \mid \theta_2), \ldots, P(e_i \mid \theta_k))$ を求める．例e_iに関するθ_jは，属性数がmの場合，1個の中心例，$2m$個の属性軸方向の“星”例，および2^m個の“階乗”例に相当するので，$k=1+2m+2^m$である．**図-5**に，$m=3$の場合を示す．mが大きい場合については，階乗例の有効な選択方法が実験計画法で提案されている．

LDS は，選択フェーズ，プロファイルフェーズ，グルー

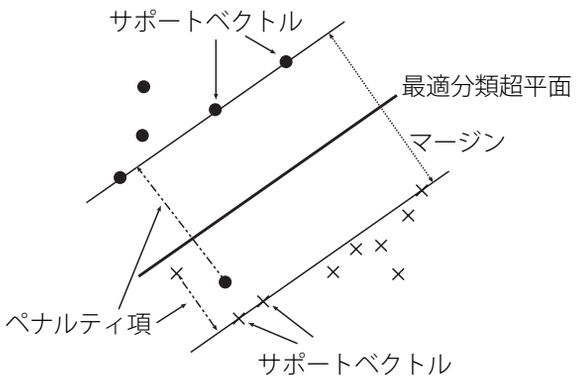

図-6　ペナルティ項を用いる場合のサポートベクトルマシンの具体例

プ化フェーズ，および生成フェーズから構成される．選択フェーズでは，$\hat{\theta}$ の推定値である$\check{\theta}$ を求め，プロファイルフェーズではデータを1スキャンして各例について尤度プロファイルを得る．グループ化フェーズではデータをさらに1スキャンして元データを尤度に基づく基準を用いてグループに分け，生成フェーズでは尤度に基づく基準を用いて各グループについて1個の疑似例を生成する．

Madiganらはまず，ロジスティック回帰を用い，LDSの性能を調べている．ここでのロジスティック回帰は，尤度を計算するための統計モデルともなっている．まず，例数1,000の人工データ集合を100種類用い，ロジスティックモデルの係数に関する標準化された2乗誤差を調べた．ただし，$\check{\theta}$ の正確性に関して3種類の手法を比較しており，それらはランダムサンプルからの推定値，最尤推定の近似値，および最尤推定値 ($\check{\theta}=\hat{\theta}$) である．その結果，1% のランダムサンプルから得られたロジスティックモデルに比較して，LDSはより正確となるパラメータの組合せが必ず存在し，$\check{\theta}$ が正確になるにつれてその優位性がゆるぎなくなることが分かった．$\check{\theta}$ を10%のランダムサンプルから得た場合には劣る場合もあるが，$\check{\theta}=\hat{\theta}$ の場合には常に優れており，10^5 倍の差がつく場合もある☆7．性能に関し，中心例から星例までの距離 d_S の影響はあまりないが，中心例から階乗例までの距離 d_F の影響はある．$\check{\theta}$ を1% のランダムサンプリングから得た場合には d_F が大きい方が性能が良く，$\check{\theta}=\hat{\theta}$ の場合には d_F が小さい方が性能が良かった．

DuMouchelらが用いたAT & Tデータを用いた実験では，LDSがDuMouchelらのデータスカッシングよりも，ロジスティックモデルの係数の正確性に関して，大幅に優れていることが分かった．ただしこのデータにはパラメータが10個あるため，階乗点を1,024個ではなく128個とする工夫をしている．顧客が他の長距離通信会社に乗り換える確率に関する回帰問題に関しても，LDSがDuMouchelらのデータスカッシングよりも優れていることが示された．

Madiganらは，ニューラルネットワークを用い，LDSの分類・回帰学習問題に関する予測性能を調べている．LDSは，ランダムサンプリング手法と比較してやはり大幅に優れており，元データからの予測精度に匹敵することが分かった．彼らは最後に，$\check{\theta}$ を $\hat{\theta}$ ほどは正確ではないが，反復的に求める方法を試している．実験の結果，3, 4回の反復回数で $\check{\theta}=\hat{\theta}$ に類似する性能を達成することが分かった．ただし彼らは，反復回数が進むにつれて d_F と d_S の値を減らしており，この点に関しては試行錯誤が必要かもしれない．

サポートベクトルマシンのためのデータスカッシング

Vapnikが提案したサポートベクトルマシン (support vector machine: SVM) は，ノイズはないが属性がきわめて多い小規模データに適する分類学習手法であり，画像やテキストなどの分類学習問題で予測精度がきわめて良い．もともとは，各例が数値属性で記述される，線形分離可能な2クラス分類学習手法において，最近傍例への距離が最大となる超平面分類モデルを求める手法である．線形分離不可能な2クラス分類学習問題に関しては，2クラスがほぼ線形分離可能である場合には各例 e_i について非負のペナルティ項 ξ_i を導入する方法が，その他の場合にはカーネル関数を用いる方法が提案されている．

前者の具体例を**図-6**に示す．サポートベクトルよりも他クラス側に近い例に関しては，距離に該当するペナルティ項が仮定される．具体的には，各例 e_i が属性値ベクトル x_i とクラス y_i (ただし $y_i=1$ あるいは $y_i=-1$)，求める分類モデルである最適分類超平面が $\mathrm{w}\cdot\mathrm{x}+b=0$ で表される場合，次が成立する．

$$x_i\cdot \mathrm{w}+b\geq +1-\xi_i \text{ for } y_i=+1$$
$$x_i\cdot \mathrm{w}+b\leq -1+\xi_i \text{ for } y_i=-1$$
$$\xi_i\geq 0\ \forall_i$$

この枠組みにおいて最小化すべき目的関数は $\|\mathrm{w}\|^2/2+$

☆7　100,000 例の中規模データを用いた場合にも同傾向の結果が得られた．

$C\sum_{i=1}^{n}\xi_i$である．ただしCはユーザによって与えられ，誤分類コストの程度を調節するためのパラメータを表す．超平面分類モデルの学習は，次の2次計画問題を解くことに帰着される．

$$\min_{\mathrm{w},b}\left(\frac{1}{2}\|\mathrm{w}\|^2+C\sum_{i=1}^{n}\xi_i\right) \quad (1)$$

$$\text{such that } y_i(x_i\cdot\mathrm{w}+b)-1+\xi_i>\forall_i \quad (2)$$

$$\xi_i\geq 0\ \forall_i \quad (3)$$

この2次計画問題は通常，大規模であり解くのに時間がかかるため，SVMに関する種々の高速化手法が提案されている．

Pavlovらは，前節の尤度に基づくデータスカッシングをSVM用に修正した[7]．彼らは尤度を計算するために，式(1)の引数がパラメータwとbに関する対数事後確率と見なせるというSVMのベイジアン的解釈を用いた．最初の項はwに関する事前確率であり，

$$p(\mathrm{w})\sim\exp(-\|\mathrm{w}\|^2)$$

と仮定し，次の項はデータの対数尤度であり，

$$\sum_{i=1}^{n}\mathrm{I}(1-y_i(\mathrm{x}_i\cdot\mathrm{w}+b))$$

に比例すると仮定する．ただし，I(z)は引数が成立すれば1，成立しなければ0となる関数のz倍である．データスカッシングの結果，疑似例に関して式(1)～(3)に類似する最適化問題を解けばよいことが分かった．相違点は，各疑似例ξ_iに関して例重みβ_iを仮定し，式(1)の項$C\sum_{i=1}^{n}\xi_i$を$C\sum_{i=1}^{n}\beta_i\xi_i$で置き換えることである．

機械学習やデータマイニングのベンチマークデータを用いた実験の結果，SVMのためのデータスカッシングのランダムサンプリング手法に対する優位性が示された．前者は後者に比較して，元データから得られた超平面分類モデルに近い超平面分類モデルを学習することができ，正答率も高い．疑似例データは小規模なので，交差検定を用いて学習の各種パラメータを最適化することもできる．高速化に関しては数倍程度と期待したほどの結果は出ていないが，これは元データにSVMを適用する場合と比較するために真に大規模なデータを用いなかったためである．Pavlovらの貢献は，尤度に基づくデータスカッシングの有効性を，応用上重要な分類学習手法で示したことだと考えられる．

ブースティングのための反復データスカッシング

ブースティング (boosting) は，ノイズがないデータに適する分類学習手法であり，分類モデルの可読性は悪いが正答率が高いという特長がある．ブースティングの基本的な考え方としては，訓練データの例分布を更新しながら分類学習手法を複数回適用し，得られた複数個の分類モデルを組み合わせて最終的な分類モデルとする．例分布は，各例に関する重みとして表され，クラス予測を間違える「難しい」例の重みを増して各回の分類学習で重要視する．AdaBoostは2クラス分類学習を対象とするブースティングの一手法であり，正答率が高いことが証明されている優れたアルゴリズムである．AdaBoost.M2は，AdaBoostを多クラス分類学習用に拡張した手法である．ただしブースティングは，分類学習手法を複数回適用するため，訓練データが大規模である場合には計算時間が長いという欠点がある．

長木らは，ブースティングで得られる例重みを活かして，ブースティングのためのデータスカッシング手法を提案した[1]．この手法では，より正確にデータを抽象化するために，データスカッシングを反復的に用いている．手順としては，クラスごとにCF木を生成して疑似例を得ることと，疑似例にブースティングを適用してCF木の葉を作成するための閾値Θを葉ごとに更新することを，指定回数行う．疑似例はCF木の各葉に1個生成し，閾値更新ではブースティングにおける最終重みと疑似例にまとめられた例数を考慮する．その他，CF木を生成する際に，ユークリッド距離ではなく，例分布を考慮する距離を用いている．

人工データとコンピュータネットワーク侵入データを用いた実験の結果，ブースティングのための反復データスカッシングの有効性が示された．この手法は，全データにブースティングを適用する場合に比較して，多くの場合正答率が数%しか低下せず，10～35倍高速だった．通常のデータスカッシングは，さらに5～6倍高速だが，正答率の低下が著しく，反復データスカッシングの優位性が示された．例分布を考慮する距離を用いることや，クラスごとにCF木を生成することの優位性も示されている．

関連議論

密度推定は，分類・回帰学習問題に比較して，前者が解ければ後2者も解けたことになるので，より難しい問

題である．Vapnikによれば，問題を解く際により難しい問題を解いてはいけない．よって，分類・回帰学習問題を解くために一種の密度推定を行うデータスカッシングの存在意義を問う声もある．そもそも，ロジスティックモデルの係数に関する正確性を調べることは，分類・回帰問題よりも難しく，密度推定に関連すると考えられる．

これに対する反論としては，データスカッシングによって既存の機械学習アルゴリズムがそのまま利用できる利点のほかに，データスカッシングで得られたモデルやデータが他の用途にも使用できる利点があげられる．たとえば，米国ではデータマイニングがプライバシーの侵害につながることが懸念されており，近年「プライバシーを保護するデータマイニング」に類するワークショップが多数開かれている．データスカッシングは，疑似データから元データを復元することが困難であると考えられるため，プライバシー保護に適すと期待できる．その他，人間は取得情報の約8割を視覚に頼っており，視覚と情報処理が密接な関係にあることから，データマイニングにおける情報可視化の重要性が指摘されている．大量のデータを可視化するには限度があるため，データを適切に変換するなどの工夫が必要である（たとえば文献8））．データスカッシングは，このような変換手法としても有望であると期待されている[2]．

さらに大規模なデータには？

機械学習アルゴリズムの効率にとって問題となるのは通常，例数ではなくて属性数である．この問題を軽減するために，機械学習においては属性選択や属性合成に関する研究が綿々と行われてきた．もっともデータマイニングでは，巨大なデータを扱うため，$O(n^2)$ となるアルゴリズムを用いることも躊躇される．このような背景から，「例合成」により例数を減らすデータスカッシングが誕生したと考えられる．なおテキストクラスタリングにおいては，例と属性の両方を似たもの同士にまとめるダブルクラスタリングが提案されている[3]．この手法がテキストデータ以外にも通用するのであれば，属性合成とデータスカッシングの統合手法に関する研究が盛んになると予想される．

現在，先端的なデータ工学研究者たちの間では，さらに大規模なデータを効率的かつ効果的に処理することが話題になっている☆8．この種のデータからのデータマイニングには，天文，ビジネス，ゲノム，および医療などの応用があり，データマイニング手法だけではなく，基盤となるデータベース技術やハードウェア技術を含む統合的なアプローチが必要になると考えられる．もちろん，データマイニング手法もより洗練された形に進化する必要があり，その中でデータスカッシング的な考え方が発展かつ普及していくと考えられる．

謝辞　本研究の一部は，文部科学省科学研究費特定領域研究「アクティブマイニング」，基盤研究(B) 16300042，および21世紀COEプログラム「情報通信技術に基づく未来社会基盤創生」の援助を受けている．

参考文献

1) Choki, Y. and Suzuki, E.: Iterative Data Squashing for Boosting Based on a Distribution-Sensitive Distance, Principles of Data Mining and Knowledge Discovery, Lecture Notes in Artificial Intelligence 2431 (PKDD), Springer-Verlag, pp.86-98 (2002).
2) DuMouchel, W., Volinsky, C., Johnson, T., Cortes, C. and Pregibon, D.: Squashing Flat Files Flatter, Proc. Fifth ACM Int'l Conf. on Knowledge Discovery and Data Mining (KDD), pp.6-15 (1999).
3) El-Yaniv, R. and Souroujon, O.: Iterative Double Clustering for Unsupervised and Semi-supervised Learning, Proc. Twelfth European Conf. on Machine Learning (ECML), pp.121-132 (2001).
4) Madigan, D., Raghavan, N., DuMouchel, W., Nason, M., Posse, C. and Ridgeway, G.: Likelihood-Based Data Squashing: A Modeling Approach to Instance Construction, Data Mining and Knowledge Discovery, 6(2), pp.173-190 (2002).
5) 中本和岐，山田　悠，鈴木英之進：動的時間伸縮法に適したデータ圧縮に基づく時系列データの高速クラスタリング，人工知能学会論文誌，18(3), pp.144-152 (2003).
6) Narahashi, M. and Suzuki, E.: Detecting Hostile Accesses through Incremental Subspace Clustering, Proc. 2003 IEEE/WIC International Conference on Web Intelligence (WI), pp.337-343 (2003).
7) Pavlov, D., Chudova, D. and Smyth, P.: Towards Scalable Support Vector Machines Using Squashing, Proc. Sixth ACM Int'l Conf. on Knowledge Discovery and Data Mining (KDD), pp.295-299 (2000).
8) Suzuki, E., Watanabe, T., Yokoi, H. and Takabayashi, K.: Detecting Interesting Exceptions from Medical Test Data with Visual Summarization, Proc. Third IEEE International Conference on Data Mining (ICDM), pp.315-322 (2003).
9) Zhang, T., Ramakrishnan, R. and Livny, M.: BIRCH: an Efficient Data Clustering Method for Very Large Databases, Proc. 1996 ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD), pp.103-114 (1996).

（平成16年12月2日受付）

☆8 SAINT2005付設Workshop on Computer Intelligence for Exabyte Scale Data Explosion.