ビッグデータ利活用のための新世代情報システム基盤技術

竹田正幸
（九州大学大学院システム情報科学研究院）

1　ビッグデータとは何か

「ビッグデータ」の定義には諸説あるが，必ずしも「大量なデータ」を指すのではなく，

　　従来技術では扱うことが困難なデータ群

を指すようである．データの扱いを困難にする要因として，以下の 3 つが挙げられる．

(1) 量が膨大であること (Volume).

(2) 発生・更新が頻繁であること (Velocity).

(3) 種類・形式が多様であること (Variety).

(1) の問題に対処する技術として，分散並列処理フレームワーク Hadoop が著名である．また，(2) に対しては，従来のストア型のデータ処理ではなく，データストリームをリアルタイムに処理する技術が必要であり，各社がそれぞれに製品を発表している．

(3) は，少し説明を要する．従来のデータベース技術は，関係データベースを中心に発展してきた．関係データは，定型データ (構造化データ) とも呼ばれ，図 1 に示すような表形式のデータを指す．表の最上段にみえる「傘の形状」「傘の表面」などを**属性**とよび，属性の並びを**関係スキーマ**とよぶ．

ところで，現在爆発的な増加を示す Web 文書，クリックストリーム，ソーシャルデータ，センサーデータ，個人の行動履歴，カメラ映像，音声などのデータは，いずれも関係デー

傘の形状	傘の表面	傘の色	···	胞子の色	発生形態	発生場所	分類
凸状	滑らか	茶		黒	散生	市街地	毒キノコ
凸状	滑らか	黄		茶	群生	草地	食用キノコ
鐘状	滑らか	白		茶	群生	低湿地	毒キノコ
凸状	鱗状	白		黒	散生	市街地	毒キノコ
凸状	滑らか	灰		茶	叢生	草地	食用キノコ
凸状	鱗状	黄		黒	群生	草地	食用キノコ

図 1: キノコに関する関係データ

タの形式をしておらず，非定型データ (非構造化データ) と称されている．これらの非定型データは，従来の関係データベース技術ではうまく扱えない．そこで，非定型データ利活用のために，新しいデータベース技術が必要になった．開発された非定型データベース技術には様々なものがあり，NoSQL と総称されている．

2　質問処理からデータ解析・採掘へ

関係データに対する「質問」は，一般に，関係代数を理論的基盤とする問合せ言語 SQL によって記述される．データベース分野では，この SQL 質問を効率的に処理するために様々な技術が研究されてきた．しかし，ビッグデータ利活用のためには，単なる SQL 質問処理ではなく，データを解析し何らかの有益な知識を得るデータ採掘の機能が必要である．このために，統計的手法とともに，機械学習の手法がよく用いられる．

$\mathcal{F}$ を D から I への関数 (の表現) の部分族とし，これを学習の対象とする．最も典型的な機械学習は，与えられた訓練例 $\langle a_1, b_1\rangle, \ldots, \langle a_s, b_s\rangle \in D \times I$ に矛盾しない未知の関数 $f_* \in \mathcal{F}$ (すなわち，$b_i = f_*(a_i)$) を求める過程である．たとえば，$D = \mathbb{R}^m$，$I = \{-1, 1\}$，$\mathcal{F} = \{\langle \mathbf{w}, b\rangle \mid \mathbf{w} \in \mathbb{R}^m, b \in \mathbb{R}\}$ とおくと，各超平面 $\langle \mathbf{w}, b\rangle \in \mathcal{F}$ は $\mathbb{R}^m$ から $\{-1, 1\}$ への 2 クラス識別関数を与える．これは m-次元空間における超平面による 2 クラス識別問題である．

3　機械学習のまな板に載せるまで

機械学習手法を非定型データに適用する際には，**特徴空間**とよばれる m-次元ベクトル空間を設定し，データをこの空間内のベクトル（**特徴ベクトル**）とみなしたアプローチをとることが多い．いったん特徴空間に写像してしまえば，様々な機械学習手法の蓄積が利用できる．この写像を設計する際には様々な方法が用いられるが，整理すると以下の 2 段階とみなすことができる．すなわち，まず，(1) 非定型データを文字列へ変換し，次いで，(2) 文字列を特徴空間へ写像する．(1) ではデータの形式・種類・性質および応用に強く依存した手法が用いられるのに対し，(2) では汎用的な手法が用いられる．

3.1　多様なデータを文字列へ

文書データは，単語の集合をアルファベットとする文字列である（日本語など，単語間の境界が自明でない場合，文を単語に分割するステップが必要となる）．DNA の塩基配列

やアミノ酸配列は，4 種類の塩基や 20 種類のアミノ酸を表す記号の集合をアルファベットとする文字列である．

一方，科学的観測データや各種のセンサーデータの多くは，実数値ベクトルの系列である．これらは，ベクトル空間を適当な個数 m の領域に分割することにより，サイズ m のアルファベット上の文字列へ変換される．

画像や音声などのメディア情報も，ベクトル系列を介して文字列へ変換される．静止画像をベクトル系列に変換する際には，色特徴量，テクスチャ特徴量，大域的形状特徴量，局所的な輝度勾配に着目した局所特徴量など，様々な特徴量が用いられる．例えば，色特徴量を用いる場合，各ピクセルのもつ色情報は，適当な色空間 (RGB 空間，HSV 空間など) における点とみなせるから，全ピクセルの色情報を左から右へ，上から下へ並べることにより，色空間上の点の系列を得る．

音楽データの場合，音響信号の各フレームから 12 次元の特徴量 (12 音名の周波数のパワーを複数のオクターブにわたって加算) を抽出し，12 次元クロマベクトルの系列を得る．

動画像は静止画像の時系列であるから，個々の静止画像を上記の方法で特徴ベクトルに変換し，特徴ベクトルの時系列を得る．

3.2 文字列から特徴ベクトルへ

関係データと異なり，文字列データは陽に属性をもっていない．このため，特徴空間への写像を考える際には，何か属性に相当するものを設定する必要がある．この切り口は，多くの場合，文字列パターン照合に基づいて与えられる．

また，特徴空間上で線形のデータ解析を行なう際には，特徴ベクトル間の内積が重要となる．一般に，次元が大きい場合，内積の計算にはコストがかかる．そこで，陽に特徴ベクトルの値を計算せず，文字列から直接内積の値を求める手法が好まれる．

以下では，アルファベット Σ 上の文字列に対する主な特徴写像について簡単にふれておく．

【n-グラム頻度統計に基づく特徴写像】 文字列 w に関する n-グラム頻度統計とは，長さ $n > 0$ の文字列の出現頻度 $\{\mathsf{occ}_u(w)\}_{u\in\Sigma^n}$ をいう．この n-グラム頻度統計に基づく特徴写像は，様々な応用でよく用いられる．たとえば，上述の静止画像に対する色特徴量で用いられるカラーヒストグラムは，$n = 1$ に相当する．次元数は $m = |\Sigma|^n$ となるが，$\mathsf{occ}_u(w) \neq 0$ となる $u \in \Sigma^n$ は高々$|w|$ 個である．任意の $w, w' \in \Sigma^*$ に対応する特徴ベクトル $\varphi(w), \varphi(w')$ 間の内積は，$|w| + |w'|$ に比例した時間で計算できる．

【ミスマッチを許す n-グラム頻度統計に基づく特徴写像】　DNA の塩基配列やアミノ酸配列では，高々k 個の文字の不一致を許容した生起頻度を用いることがある．上の $\mathsf{occ}_u(w)$ の値を，高々k 個の不一致を許した生起頻度で置き換えた写像となる．内積の値は，$n^{k+1}|\Sigma|^k(|w|+|w'|)$ に比例した時間で計算可能である．

【ギャップ付き n-グラム頻度統計に基づく特徴写像】　$u \in \Sigma^n$ が $w \in \Sigma^*$ の部分列であるとは，単調増加整数列 $\gamma = \langle i_1, \ldots, i_n \rangle$ $(1 \leq i_1 < \cdots < i_n \leq |w|)$ が存在して $w[i_1] \cdots w[i_n] = u$ となるときをいい，$\mathsf{width}(\gamma) = i_n - i_1 + 1$ を**出現の幅**とよぶ．このような単調増加列 γ 全体の集合を $I_{u,w}$ で表し，

$$\mathsf{occ}_u(w) = \textstyle\sum_{\gamma \in I_{u,w}} \lambda^{\mathsf{width}(\gamma)}$$

と定める．ここに，λ は $0 < \lambda < 1$ を満たすパラメータである．出現の幅 $\mathsf{width}(\gamma)$ が大きいほど，加算される $\lambda^{\mathsf{width}(\gamma)}$ の値は小さくなる．内積の値は，$n|w||w'|$ に比例した時間と $n\min\{|w|,|w'|\}$ に比例した領域を用いて計算できる．

【部分文字列頻度統計に基づく特徴写像】　n-グラム頻度統計において n にすべての正整数値を許したもの，すなわち，$\{\mathsf{occ}_u(w)\}_{u \in \Sigma^+}$ を考える．$u \in \Sigma^+$ は可算無限個あるが，そのうち $\mathsf{occ}_u(w) \neq 0$ となるのは高々$|w|^2$ 個である．w, w' に依存した Σ^* 上のある同値関係 $\equiv$ を導入すると，

$$u \equiv v \rightarrow (\mathsf{occ}_u(w) = \mathsf{occ}_v(w)) \wedge (\mathsf{occ}_u(w') = \mathsf{occ}_v(w'))$$

が成り立ち，同値類の個数は高々$2(|w|+|w'|)$ となる．同値関係 $\equiv$ に基づくデータ構造を用いることにより，内積の値は $|w|+|w'|$ に関する線形時間で計算できる．

4　時代を超えて通用する技術

有川節夫教授（現九州大学総長）らの研究グループは，SIGMA と名付けた汎用テキストデータベース管理システムを開発し，九州大学大型計算機センター (現情報基盤研究開発センター) において 1981 年より公開してきた．

SIGMA は，ユーザの与えた質問から一種の有限オートマトンを構築し，テキストデータをただ一度走査する間にすべての質問処理を完了する．事前にインデックスを創成せずデータを全部走査することから，開発当時，常識はずれで非効率な手法と見なされもしたが，これは，最初にあげた Velocity が要求する「実時間ストリーム処理」そのものである．

また，SIGMA は，入力を「陽には構造を持たない一本の文字列」として扱うが，データ中に含まれるタグにより，実行時にスキーマを認識しながら処理を行なうことができる．これは，Variety が要求する「非定型データ処理」そのものである．

さらに，SIGMA は，富士通 (株) の Interstage Shunsaku Data Manager (以下 Shunsaku) においてメインエンジンに採用されている．Shunsaku では，データを数百 MB 程度に分割し，各々に CPU を割り当てて分散処理を行ない，結果をマージすることで，大規模データに対応している．これは，Volume が要求する「分散並列処理」そのものである．

このように，SIGMA/Shunsaku は，ビッグデータという言葉が生まれるはるか昔から，ビッグデータ処理に必要な技術を提供していた．特筆すべきは，ビッグデータに関わる 3 つの課題 Volume/Velocity/Variety を，１つの技術によって同時に解決する点である．「オートマトンと形式言語理論」という，一見世の中の役に立ちそうにない基礎理論に根ざした要素技術が，実は時代のはるか先を行っていたというこの事実は，我々に，数理科学的な基礎理論研究の重要性を改めて認識させる．

ビッグデータを扱う際にはデータの圧縮が不可欠であるが，データ転送コストやストレージ容量を節減できる反面，利用時に展開のコストがかかる．著者らは，2002 年頃までに，データを圧縮したままの形で高速処理する研究を押し進め，非圧縮時よりもむしろ高速になるという「圧縮による高速化」技術を開発し，SIGMA 技術への組み込みを終えている．

また，テキストデータにタグを埋め込んで階層的な構造を与えた XML データの形式に着目し，これも SIGMA 技術へ組み込んでいる．これは，ある種の決定性文脈自由言語に関する理論に基づくものである．

富士通（株）は，ビッグデータ処理基盤として，(1) データストリームに対する実時間複合イベント処理を行なう Interstage Big Data Complex Event Processing Server，(2) 大量一括データの高速処理を行なう Interstage Data Effector，(3) 大量トランザクションの高速処理を行なう Symfoware Server の 3 つの製品を世に出している．各製品には，著者らと（株）富士通研究所による長年にわたる共同研究の成果として，SIGMA の流れを汲む技術が組み込まれている．

「ビッグデータ」はいわゆるバズワードであり，このブームがいつまで続くかわからない．だが，流行り廃りに関わらず,「様々な**現場**において，解決すべき課題を抱えた**ユーザ**が，**データ**を利活用する術をもたないでいる」という状況は，以前から存在したし，今後ますます顕在化するであろう．そして，深い基礎理論に基づく要素技術は，陳腐化することなく，時代を超えて通用するものと著者は信ずる．