Vol.2009-HPC-121 No.4
2009/8/4

線形方程式求解アルゴリズムの実行データに対するクラスタリング技術の一適用

小谷和正[†1]　須田礼仁[†1,†2]

自動チューニングの研究過程ではしばしば網羅的な性能評価が行われる。データの数が膨大であると人間の目だけでそこから情報を引き出すのは困難で、ある種の機械学習による支援が必要であると考えられる。

本稿では 345 個の疎行列に対して 8040 通りのパラメータ組み合わせで線形方程式の求解を行った実行データを用意し、それに対して EM アルゴリズムによるデータクラスタリングを行うことで行列を分類する実験を行った。素性としては、各種アルゴリズムを固定した場合にその他のパラメタを変化させて求解が収束に至った割合などを考えたが、結果としてはうまく分類できず、一般化した議論の難しさを再確認するに留まった。

An Application of Clustering Technique to Executional Data of Solving Linear Systems

Kazumasa Kotani [†1] and Reiji Suda [†1,†2]

Work on automatic tuning of softwares often needs exhaustive performance evaluation. As it's difficult to manually extract meaningful information from enormous executional data, so help of some kinds of machine learning techniques may be necessary.

We prepare the executional data of solving linear systems for 345 sparse matrices with 8040 parameter configurations. In this paper, some experiments of data clustering are done using EM algorithm to try classifying those matrices. As feature variables in the experiments, we take the rate of successful executions using each specific solving/preconditioning/scaling algorithm with other parameters widely changed. But its result shows that classification is far from successful, then we may reaffirm that common discussions about the effects of the algorithms is quite difficult.

1. はじめに

自動チューニング技術に関する様々な研究が行われている。そこでは研究段階あるいはエンドユーザによる利用段階においても、網羅的な性能評価が必要となる場合があり、得られる多量のデータから情報・知識を取り出すことが求められる。本稿では、我々の進めている研究[1] 以降得られた線形方程式求解の実行データを対象として、期待値最大化（EM）アルゴリズムを用いたデータクラスタリング[2] による知識発見を目指した実験を行う。

線型方程式の求解は、前処理も含めた求解アルゴリズムの性能が入力の行列データに強く依存する。これについては先行研究がいくつか[3),4)] が存在するが、そこで述べられている通り、ある程度の一般論はあるものの実用的には未だ体系化しきれないのが現状である。本研究においては未だ新しい知見は具体的に得られていないが、我々のアプローチを進めていくことで何らかの発見に寄与できればよいと考える。

以下ではまず EM アルゴリズムと本実験で具体的に用いた手法について説明する。次に我々の行った実験内容と結果を示し、最後に問題点などを含めてまとめる。

2. EM アルゴリズムによるクラスタリング

ここで述べるデータクラスタリングとは、与えられたデータ集合を教師無し学習により分類することを指す。

本実験ではクラスタリングによく使われている手法として、混合正規分布モデルと EM アルゴリズムによる最尤推定を採用する。以下では簡単に説明を行うが、詳細は文献[2] などを参照のこと。

2.1 混合正規分布モデル

観測データ $\mathbf{x}$ は、それぞれ予め定められたいくつかの素性 (feature) 値の組である。素性については後述するが、本実験では例えば各々の解法アルゴリズムを選んだ場合に収束しやすいかどうか、などといった情報をあてはめている。

N 個の観測データ $\mathcal{X} = \{x_1, ..., x_N\}$ を K 個のクラスタに分類する時、各データが K 個の確率分布の重み付け和で表される確率分布に従って独立に発生していると考え、

†1 東京大学 大学院情報理工学系研究科 コンピュータ科学専攻
Dept. of Computer Science, Graduate School of Information Science and Technology, University of Tokyo
†2 JST, CREST

$$p(x;\theta) = \sum_k \alpha_k f(x;\xi_k) \tag{1}$$

とモデル化する。分布族 f としては正規分布が用いられることが多く、本実験でもこれを採用する。また通常は正規分布の共分散については対角成分だけが用いられる。すなわちモデルパラメタ $\theta = \{\xi_1, ..., \xi_K, \alpha_1, ..., \alpha_K, \}$ は、ここでは K 個の正規分布の平均 μ_k と分散 σ_k^2、

$$\alpha \geq 0, \ \sum_k \alpha_k = 1 \tag{2}$$

を満たす重み α_k の組である。

最尤推定とは、データ $\mathcal{X}$ に対する（何らかの方法で推定した）パラメタ θ の「尤もらしさ」として、以下で定義する尤度

$$l(\theta;\mathcal{X}) = \prod_i p(x_i;\theta) \tag{3}$$

の対数

$$L(\theta;\mathcal{X}) = \sum_i \log \left\{ \sum_k \alpha_k \mathrm{f}(x_i;\xi_k) \right\} \tag{4}$$

を考え、これを最大化する推定法である。これを解くことは一般には難しく、以下で示すEM アルゴリズムによる（局所）最適化がよく用いられる。

2.2 EM アルゴリズムによる最適化

EM アルゴリズムは、隠れ変数をもつ確率モデルに対する最尤推定を反復法により行う手法である。

ここでは各サンプルデータ x_i が属するクラスタ番号 k_i を隠れ変数と考える。そうすると式 (13) に対応する完全データ $y_i = (x, k)$ の分布は

$$p(x,k;\theta) = \alpha_k f_k(x;\xi_k) \tag{5}$$

であるが、反復の t ステップ目においてモデルパラメタ $\theta^{(t)}$ が既知であるとすると、ベイズの定理よりデータ x_i がクラスタ k_i に属する確率が

$$q_{ik}^{(t)} = p(x_i, k_i; \theta^{(t)}) / p(x_i; \theta^{(t)}) \tag{6}$$

と求めることができる。

これを用いて、完全データ y の対数尤度の条件付期待値を以下のように θ についての関数 Q とおき

$$Q(\theta|\theta^{(t)}) = \sum_i \sum_k q_{ik}^{(t)} \log p(x_i, k; \theta) \tag{7}$$

と定め（Expectation ステップ）、Q を最大化するような θ を計算し（Maximization ステップ）、それを次反復の $\theta^{(t+1)}$ として更新していく。

分布 f が正規分布の場合、具体的には

$$\alpha_k^{(t+1)} = \frac{1}{n} \sum_{i=1}^{N} q_{ik}^{(t)}, \tag{8}$$

$$\mu_k^{(t+1)} = \frac{1}{n\alpha_k^{(t+1)}} \sum_{i=1}^{N} q_{ik}^{(t)} x_i, \tag{9}$$

$$(\sigma^2)_k^{(t+1)} = \frac{1}{n\alpha_k^{(t+1)}} \sum_{i=1}^{N} q_{ik}^{(t)} (x_i - \mu_k^{(t+1)})^2, \tag{10}$$

$$\tag{11}$$

と、容易に計算できる式が得られる。

なお本実験では、各クラスタのモデルパラメタ初期値を K 平均法によるクラスタリング結果で設定する手法を採用し、反復の終了条件は対数尤度の変化量が 10^{-6} 以下としている。

簡単な解説

1 つのデータ x_i に対する θ の尤度関数は、それが各クラスタ k に属する確率の和であることを考え、条件付期待値の定義から

$$\begin{aligned} p(x_i;\theta) &= \sum_k p(x_i, k; \theta) \\ &= \sum_k \{p(x_i, k; \theta)/p(k|x_i)\} p(k|x_i) \\ &= \mathrm{E}\left[p(x_i, k; \theta)/p(k|x_i)|x_i\right] \end{aligned}$$

と表せる。これを用いると、θ の対数尤度は

$$L(\theta;\mathcal{X}) = \sum_{x_i} \log \mathrm{E}\left[p(x_i, k; \theta)/p(k|x_i)\right]$$

となり、これは Jensen の不等式を適用し

$$L(\theta;\mathcal{X}) \geq \sum_{x_i} \mathrm{E}\left[\log p(x_i,k;\theta)/p(k|x_i)\right]$$
$$= \sum_{x_i} \mathrm{E}\left[\log p(x_i,k;\theta)\right] - \sum_{x_i} \mathrm{E}\left[\log p(k|x_i)\right]$$

と下から抑えることができる。第二項は θ によらないので、$L(\theta;\mathcal{X})$ を最大化するにはこの第一項を最大化すればよいことがわかる。

条件付期待値の定義を再度用いると

$$\mathrm{E}\left[\log p(x_i,k;\theta)\right] = \sum_k p(k|x_i;\theta)\log p(x_i,k;\theta)$$

となり、Q が導かれる。標準的な EM アルゴリズムでは、このうち $p(k|x_i;\theta)$ をそれぞれ t ステップ目で既知の $\theta^{(t)}$ により計算し、上式を最大化するように次の θ を定め更新するが、これはベイズ推定における事前確率と事後確率の関係に相当している。

3. クラスタ数の推定と素性選択

前節に示した基本的なアルゴリズムではクラスタ数 K が既知であったが、クラスタ数をいくつにするべきかは通常、事前には不明である。一般的には K をデータの数に近づけることで尤度も大きくすることができるため、K によるペナルティを与えて何らかの規準での最適値を推定することが多い。

3.1 手　　法

本稿では、K をある程度動かした上で赤池情報量規準（AIC）を計算して最適なクラスタ数を決定する方法をとった。本実験の場合は EM アルゴリズムで得られる局所最大化された対数尤度を $L^*(\mathcal{X})$、素性の数を N_f とすると、

$$\mathrm{AIC} = -2L^*(\mathcal{X}) + 2K(1+2N_f) \tag{12}$$

として推定を行う。

また本研究の目的としては、プログラムの性能について入力データ・実行パラメータ・性能の間にどのような関係があるかを発見したいというものがあるので、そこに関係のない余計な変数（素性）は除外したい。これは素性選択と呼ばれるものでいくつかの方法が存在するが、本実験では素性を一つずつ追加してゆくだけの単純な前進選択法により行う（**図 1**）。

素性の選択も AIC により行うこととするが、選択されていない素性変数を全く無視して式 (13) により尤度の計算をしてしまうと、異なるデータに対するクラスタリングについて尤度を比較していることになってしまう。そのため未選択の素性についてはクラスタに依存しない何らかの共通分布に従っているとしてモデルを比較するべきであると考えられる。

本稿では以下のように、クラスタリングに用いない各素性変数は平均・分散が全データについて変数ごとの平均・分散であるような正規分布にしたがっていると定めてモデル化を行うこととした。すなわち各素性変数の従う正規分布を c_j と書くと、式 (13) の確率分布モデルに項を追加して、

$$p(x;\theta) = \sum_k \alpha_k f(x;\xi_k)\cdot\prod_j c_j(x) \tag{13}$$

と定める。このとき式 (14) の対数尤度は

$$L(\theta;\mathcal{X}) = \sum_i \log\left\{\sum_k \alpha_k \mathrm{f}(x_i;\xi_k)\right\} + \sum_i\sum_j \log c_j(x_{ij}) \tag{14}$$

となり、これは選択済みの素性のみを用いて EM アルゴリズムを実行した結果に、予め計算可能な c_j による対数尤度をそれ以外の素性について加えれば良い。

3.2 人工的なデータによる確認

参考データとして、以下のように生成した 6 次元の点で実験を行う。素性のうち 2 次元は正規分布 $\mathcal{N}(m_i,I)$, ただし $m_1=(0,3), m_2=(1,9), m_3=(6,4), m_4=(7,10)$ に従い、他の 4 次元は i に関係ないノイズとしての正規分布 $\mathcal{N}(0,1)$ に従う点を、i につきそれぞれ 200 個ずつ作る。

生成した点に対して前節の手順でクラスタリングを行った結果を**図 2** に示す。素性数は 6 までとして、クラスタ数を 2〜最大 5 もしくは 30 の二通りに設定して推定を行い、それぞれ AIC が最小となった素性集合を選んでいる。また図の点が生成した点、円は各正規分布の平均 μ を中心として分散の平方根 σ の二倍を半径にとった領域を描いている。また各正規分布の平均と分散の平方根を**図 3** に示す。

最大クラスタ数を 30 に設定した場合は 3 つ目のノイズ変数も用いて 6 つのクラスタに分類している。どちらの結果が良いか（AIC 以外の方法で）客観的に評価することは現段階では難しいが、直感的には元々想定したクラスタ数に等しく領域間の距離が遠い前者の方が望ましく見える。

4. 実　　験

4.1 対象データ

クラスタリングを行う対象として用いる線形方程式の求解データは我々の昨年の発表[1] と

入力: 観測データ $\mathcal{X} = \{x_{ij}\}$ $(i = 1, \ldots, N\ ,\ j = 1, \ldots, N_f)$,
最大素性数 M_{max}
最小、最大クラスタ数 $K_{min},\ K_{max}$
出力: 選択された素性集合 F, モデルパラメータ θ, クラスタ数 K
初期状態: 素性集合 $F = \{\}$(空集合) とする.
while $|F| < M_{max}$:
foreach f' **in** $\{1, \ldots, N_f\} \setminus F$:
$F_{f'} := F \cup \{f'\}$
$\mathcal{X}' := \{x_{ij}\}$ $(i = 1, \ldots, N\ ,$ **in** $F_{f'})$
foreach K **in** $\{K_{min}, \ldots, K_{max}\}$:
入力データ $\mathcal{X}'$, クラスタ数 K として
2 節の通りにクラスタリングを行い,
式 (14) に従って AIC を計算する.
AIC が最小となるようなクラスタ K を採用する.
AIC が最小となるような素性 f' を選択
$F := F \cup \{f'\}$

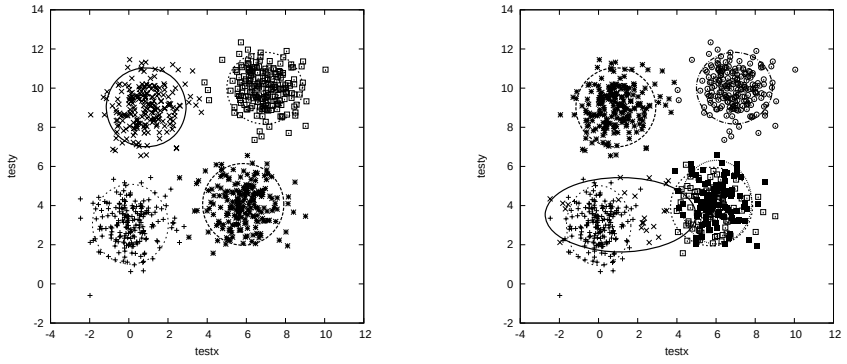
図 1 クラスタ数の推定と素性選択の二重ループ

同様、Florida 大の疎行列コレクション[5] の行列に対して数値計算ライブラリ Lis[6] による求解を行った実行データである。

実験条件は前回とほぼ同様だが、右辺ベクトル b は、行列データに同梱されている場合はそれを用い、なければ $b = A \times (1, \ldots, 1)^T$ を用いた。また収束判定基準は残差 $\leq 10^{-12}$ で、最大反復回数は行列サイズと同一に設定した。また対応アルゴリズムの増加に伴いパラメタの組み合わせは一つの行列につき 8040 通りであった。

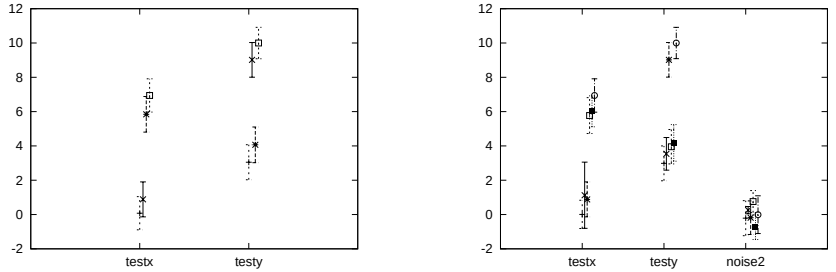
実行データの取得に用いたマシンは以下のような環境であった。

- CPU: Intel Core2Duo 6400 2.13GHz
- L2 Cache: 2048KB
- Main memory: 4GB
- OS: Debian Linux 4.1.1, Kernel 2.6-amd64
- Library: Lis-1.2.2
- Compiler: gcc-4.3.2 (最適化:-O2)



最大クラスタ数 5　　最大クラスタ数 30

図 2 人工的なデータに対するクラスタリング結果



最大クラスタ数 5　　最大クラスタ数 30

図 3 人工的なデータに対するクラスタリング結果 (μ と σ)

本実験では、アルゴリズムが収束に至った上で真の相対残差が 10^{-7} 以下のとき実行が成功であると定めた。以上の条件で、成功したパラメタ組が少なくとも 1 つ以上存在した行列 345 個の実行データに対して、次節のように実験を行った。

4.2 クラスタリング実験

本実験では、各求解・前処理・スケーリングのアルゴリズムについて、

(1) *_succ: それぞれを用いた実行パターンの中で、成功したものの割合 (0〜1)

(2) *_faster: 全 8040 パターンにおける最小実行時間の 1.3 倍を基準として、その基準以

素性	AIC	追加変数	K
0	10969.34		
1	10859.87	jacobi_p_succ	29
2	11631.74	gpbicg_s_succ	5
3	11932.89	gpbicr_s_succ	4
4	12223.29	crs_s_succ	4
5	12521.50	ssor_p_succ	9
6	12843.08	cgs_s_succ	4
7	13138.17	tfqmr_s_succ	4
8	13442.11	bicgstabl_s_succ	4

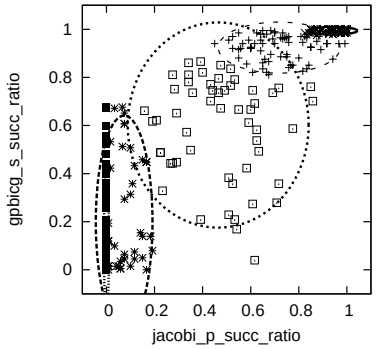


図 4 AIC の変化と素性数 2 の場合のクラスタリング結果 (成功率を用いた場合)

素性	AIC	追加変数	K
0	9774.27		
1	9798.27	iluc_p_faster	4
2	9800.06	fgmres_s_faster	4
3	9872.27	bicrstab_s_faster	7
4	9977.51	bicg_s_faster	13
5	10098.14	sainv_p_faster	17
6	10435.32	gpbicr_s_faster	25
7	11066.47	sor_s_faster	4
8	11791.22	cr_s_faster	4

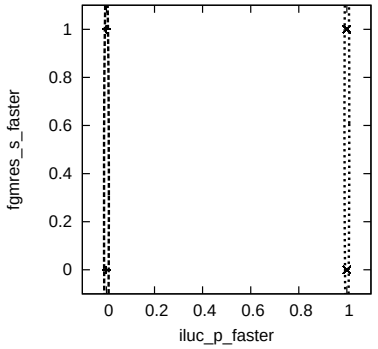


図 5 AIC の変化と素性数 2 の場合のクラスタリング結果 (速度を用いた場合)

内の実行時間を達成したパターンが少なくとも一つあるか否か（0 または 1）という 2 種類の素性集合を考えた。それぞれの素性候補を用いて、最大クラスタ数 36, 素性は 8 変数までとして図 1 のアルゴリズムを適用した結果、追加された素性と AIC の変化, また素性数 2 でのクラスタリングの結果をそれぞれ**図 4**, **図 5** に示す。それぞれ右側の図は素性のもつ値を点とし、円は μ を中心として $2\sigma + 0.01$ を径にとった領域を描いている。

素性数 0 とはつまり全ての変数が領域分割せずにそれぞれ単純な正規分布に従っていると考えた場合である。素性を用いてクラスタリングを行った場合はそれよりも尤度が小さくなることが望ましいが、実際にはほぼ素性を追加するごとに大きくなってしまっている。図 4 で唯一、jacobi 法前処理の成功率のみでクラスタリングを行った場合に AIC が小さくなるが、各クラスタに対応する正規分布の平均と分散は図 **図 6** のようになっており、これだけでは少なくとも人間の目に分かりやすい結果とは言えない。

なお、図 4 の場合は、また各クラスタの平均と分散（**図 7**）によっても確認できるが、各クラスタはおよそ対角線上に分布しているように見える。これについて、jacobi 前処理法と gpbicg 法それぞれの収束性に対する効果に相関があると見るべきか、またはそもそも特定のアルゴリズムには関係なく全体的な収束率が影響した分類になっていると見るべきか、現時点では判断できない。全パラメータ中での単純な成功率を素性として含めた場合も結果に変化がなかった（図は省略）こともあり、今のような逐次に素性を追加する方法では難しいと思われる。

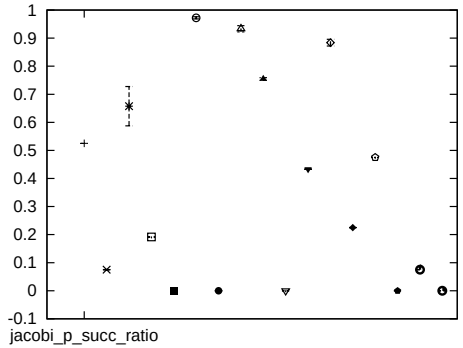


図 6 jacobi 前処理での成功率による分類結果、各正規分布の $\mu \pm \sigma$

また、図 1 に示したアルゴリズムは効率が悪く、素性数やクラスタ数に対して組み合わせ探索が必要になる。本実験では素性およそ 30 個、クラスタ数 36 の場合だが、素性を一つ追加するためのループ一回に 5 分程度を要した。この問題に関しては、素性選択も確率モデルに含めた上で EM アルゴリズムによりクラスタリングと素性選択を同時に行う手法[7] が既に提案されており、そちらを用いるべきだと思われる。また素性の有効性が全て 0 から 1 の連続値で推定できるという利点もある。

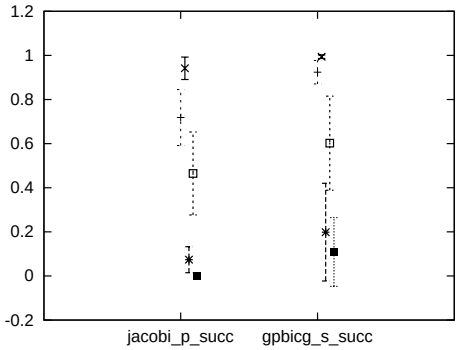


図 7 jacobi 前処理と gpbicg 法での成功率による分類結果、各正規分布の $\mu \pm \sigma$

4.3 行列データとの対応づけ

プログラムの実行結果に関して何らかの有意なクラスタリング結果が得られたとして、それを用いてチューニングを行うためには、それぞれの入力データ（行列）がもつどのような性質がクラスタ分類に関係するかを確認する必要がある。そのための方法の一つとして、本研究では以下のような方法と採ることを考えている。

入力データに関する何らかの性質をいくつか取り出した素性ベクトルを考え、何らかのモデルを用いて入力データの素性と実行結果のクラスタについての対応を（教師つき）学習する。ここでは単純ベイズ識別器を使い、例として行列が対称行列か否か（厳密には対称性99.9%以上、とした）という素性を用いた場合の学習結果を示す。確率スコアについて、ここでは、$p(S = \text{対称} \,|C = \text{クラスタ}\ k) = (k$ に分類された行列のうち対称行列の数$)/(k$ に分類された行列数$)$ を図示している。

図 9 は前節の結果のうち jacobi 法前処理と gpbicg 法の収束成功率によって行列を分類した場合で、また jacobi 法のみの場合は**図 8** に示す。どちらの場合も対称行列かどうかはいずれのクラスタに属するかに極端な差は出ていないものの、素性を追加選択することで何らかの知見を得ることが目標となる。

5. まとめと今後の課題

本稿では、Florida 大の疎行列コレクションに存在する 345 個の行列を用いて、線形方程式を求解ライブラリ Lis により解いた実行データを用意し、その実行パラメタに依存した性能の違いによって行列データを分類する実験を行った。

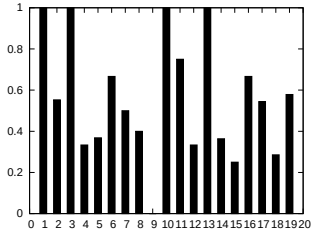


図 8 対称行列であることによる各クラスタへの帰属確率スコア (素性数 1)

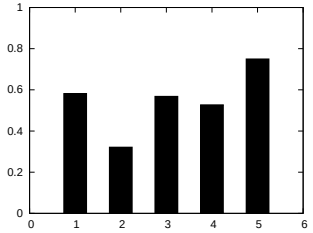


図 9 対称行列であることによる各クラスタへの帰属確率スコア (素性数 2)

実験では各種アルゴリズムにより解が正常に求まった割合、もしくは高速な実行を達成できるかどうかの二種類の素性ベクトルを考え、それぞれに対して混合正規分布と EM アルゴリズムによるクラスタリング、AIC によるクラスタ数決定、前進選択法による素性選択を適用した。しかしながら、ほとんどの素性に関して単純な正規分布よりも評価が悪くなってしまう結果となり、うまく分類ができたとは言えなかった。これは各種アルゴリズムの性能が、他のアルゴリズムやパラメタの選択にも大きく依存しており、一般的な議論がしづらいということが確認できるものでもあり、加えて性能を評価する素性の抽出にも深い配慮が必要であることを示しており、本研究のアプローチを進めるには重要な点である。

また、今後の展望として実行データの分類と入力データの性質を結びつける方法について示した。クラスタリング結果があまり有意でないため、ここでは手順を示すにとどまっている。将来的にはある特定の実行パターンが成功するかどうかなどを素性とすることが考えられる。また、ここでも素性選択は必要となる。

謝辞 本研究は、文部科学省科学研究費補助金、特定領域研究「情報爆発」の補助の下で実施しました。また、実験の多くは同提供のプラットフォーム InTrigger をお借りして行いました。

参 考 文 献

1) 小谷和正，須田礼仁：統計的パターン認識手法によるソフトウェア自動チューニングのための実験計画 (2008 年並列/分散/協調処理に関する『佐賀』サマー・ワークショップ (SWoPP 佐賀 2008))，情報処理学会研究報告 (HPC)， No.74, pp.37–42 (2008).
2) 赤穂昭太郎：EM アルゴリズム : クラスタリングへの適用と最近の発展，日本ファジィ学会誌， Vol.12, No.5, pp.2–10 (2000).
3) 伊藤祥司：体系的評価から見た線形方程式求解に対する前処理の実効性能 (SWoPP 佐賀 2008)，情報処理学会研究報告 (HPC)， No.74, pp.115–119 (2008).
4) 伊藤祥司：線形方程式求解アルゴリズムに対する体系的な性能比較について，情報処理学会研究報告 (HPC)， No.87, pp.281–286 (2006).
5) Davis, T.: The University of Florida Sparse Matrix Collection, http://www.cise.ufl.edu/research/sparse/matrices.
6) Project, S.: Lis, http://www.ssisc.org/lis/.
7) Law, M., Figueiredo, M. and Jain, A.: Simultaneous feature selection and clustering using mixture models, *IEEE Transactions onPattern Analysis and Machine Intelligence*, Vol.26, No.9, pp.1154–1166 (2004).