情報処理学会研究報告
IPSJ SIG Technical Report

Vol.2011-MPS-82 No.13
2011/3/7

部分空間クラスタリングを用いた自己教示学習

東松信幸[†1]　上原邦昭[†2]

Raina らの自己教示学習は，未知データに対して予測を行う際に，繰り返し計算を含む最適化を行う必要があり，計算コストが大きい．未知データに対する予測に必要な計算コストが大きい事は，その手法の実用化を考えた際に問題となる．本研究では．これを解決する手法として，自己教示学習に部分空間クラスタリングを使用する手法を提案する．

Self-taught learning using subspace clustering

Nobuyuki Tomatsu[†1] and Kuniaki Uehara[†2]

Raina's self-taught learning method needs iterative calculation and is computationally expensive during the prediction of unknown data. High computational cost is a problem in practical use. In this paper,we propose a self-taught learning method using subspace clustering as a solution to this problem.

1. はじめに

機械学習の研究分野の一つに，学習対象に似たドメインのデータを活用して，予測精度を向上させる転移学習がある．転移学習における学習対象のドメインを目標ドメイン，学習対象に似たドメイン (転移させる情報を持っているドメイン) を元ドメインと呼ぶ．転移学習の中でも，元ドメインのラベルなしデータと目標ドメインのラベルありデータを用いる手法は特に自己教示学習と呼ばれている．

†1 神戸大学大学院工学研究科
Kobe University Graduate School of Engeneering
†2 神戸大学大学院システム情報学研究科
Kobe Universit Graduate School of System infomatics

自己教示学習の代表的な手法として，Raina らの手法[3] がある．Raina らの手法は，スパースコーディングアルゴリズムという手法を用いてデータ表現を学習し，転移学習を行う手法である．しかし，学習の時だけでなく，未知データに対する予測の段階でも最適化問題を解く必要があり，予測に大きな計算コストを必要としてしまう．本稿では，部分空間クラスタリングを用いて，未知データに対する予測の計算コストを低減し，より実用に耐える自己教示学習の手法を提案する．

2. 既存手法

Raina らによって提案された手法では，スパースコーディングアルゴリズムが用いられている．スパースコーディングアルゴリズムは，スパースなデータ表現を学習する手法である．[1] データ表現がスパースであるとは，要素（各次元の値）のほとんどが 0 であることを言う．これは，脳の一次視覚野の神経細胞が局所的に活動することに由来しており，画像処理などで利用されている．具体的には，画像からスパースな性質を制約にして，新たなデータ表現を計算するパラメータを学習すると，そのパラメータにエッジなどの局所的な特徴が現れるという性質を用いるのである．

スパースコーティングアルゴリズムは，データを複数のベクトル（ベースベクトルと呼ぶ）の線形和で近似する．以下に，スパースコーディングアルゴリズムを，データ表現の学習時と変換時に分けて述べる．

まず，データ表現の学習時のアルゴリズムについて述べる．最適化の目的関数を式 (1) に示す．

$$\sum_{i=1}^{m} ||x^{(i)} - \sum_{j=1}^{n} a_j^{(i)} b_j||^2 + \beta \sum_{i=1}^{m} \sum_{j=1}^{n} |a_j^{(i)}| \tag{1}$$

$x^{(1)}, \cdots, x^{(m)}$ をデータベクトル，$b_1, \cdots, b_n$ をベースベクトル，$a^{(1)}, \cdots, a^{(m)}$ を係数とする．それぞれ列ベクトルである．最初の項が近似の誤差であり，2 つ目の項が係数をスパースにする制約である．学習時には，与えられたデータベクトルに対して，式 (1) をベースベクトルと係数の両方について最小化し，ベースベクトルの最適化を行う．最適化は，ベースベクトルと係数について交互に最小化し，式 (1) の値を収束させる手法をとる．

次に，データ表現の変換時のアルゴリズムについて述べる．変換する対象のデータベクトル，ベースベクトルが与えられた時に，式 (1) を係数について最小化して，係数をスパースにする．この係数が新たなデータ表現となる．この計算を高速に行う手法として feature-sign

search algorithm[2] が提案されている．このアルゴリズムは，内部で繰り返し計算を必要とし，その回数が増えるほど計算コストが大きくなってしまうという問題がある．この問題を解決するには，繰り返し計算などの大きい計算量を必要とせず，特徴空間の領域から直接特徴量に変換できる方法が適している．次章では，この解決策としてクラスタリングを用いた手法を提案する．

3. 提案手法

3.1 概　　要

提案手法では，部分空間クラスタリングを用いている．まず，元ドメインのデータで部分空間クラスタリングを行い，その結果を目標ドメインに適用し，一般的な分類器で学習を行う．

部分空間クラスタリングとは，データの持つすべての次元ではなく，その部分空間でクラスタリングを行う手法である．ここでは，部分空間とは，データの持つ次元の部分集合を指す．提案手法で行う部分空間クラスタリングは，部分空間の決定，クラスタリングに分かれている．部分空間の決定に，ENCLUS_INT という手法に変更を加えた手法を使用し，クラスタリングには，クラスタ数を自動的に判定し，クラスタと部分空間の領域を対応させる方法を使用している．また，目標ドメイン，元ドメインと目標ドメインの関係について，それぞれ仮定を設けている．目標ドメインには low density separation を拡張した仮定を，元ドメインと目標ドメインの関係にはそれぞれの事例の分布の関係を仮定している．

3.2 仮　　定

目標ドメインの仮定として，low density separation と呼ばれる仮定を複数の空間に拡張している．low density separation とは,「決定境界は低密度の領域に存在する」という仮定で，半教師付き学習で使われている仮定である.[6] 決定境界とは，異なるクラスの存在する領域同士の境界線の事である．

提案手法では，low density separation を次のように拡張する．

仮定 1　「決定境界は，特徴空間の部分空間で低密度な領域に存在する.」

low density separation は，一つの空間での密度に関する仮定であるが，提案手法は，複数の空間での低密度領域を同時に利用している．具体的には，複数の部分空間でクラスタリングを行った結果を，データ表現として同時に利用して，一つの空間のみではなく複数の部分空間の低密度領域を組み合わせた決定境界を見つける事を可能にしている．

次に，元ドメインと目標ドメインの関係の仮定について述べる．low density separation をそのまま活用するには，半教師付き学習のように，クラスタリングを行う対象と分類を行う対象の分布が同じでなければならない．しかし，自己教示学習では，元ドメインと目標ドメインが異なる分布を持つ事を想定している．そこで，元ドメインと目標ドメインの分布の関係について，以下の仮定を設ける．

仮定 2　「元ドメインでの低密度の領域は，
目標ドメインの対応する領域においても低密度である.」

このような仮定を設け，元ドメインでの低密度領域の情報を，目標ドメインでの低密度領域の情報として利用する．最終的に，仮定 (1)，(2) が共に成り立つ場合，目標ドメインにおける低密度領域の情報を活用して，決定境界をより正確に学習し，予測精度を向上させる事ができると考えられる．

3.3 部分空間の決定

まず，部分空間の決定に用いる ENCLUS_INT について説明し，次にその問題点と，提案手法での変更点について述べる．ENCLUS_INT は，エントロピーを用いて，部分空間クラスタリングに有用な部分空間を求める手法である．クラスタリングに有用な部分空間であるかを判断するため，エントロピーと interest_gain という指標を用いている．エントロピーは確率分布の不確かさを表す指標である．確率関数の分布が偏っていて，小さい領域に事例が集まる場合にはエントロピーが低くなるので，クラスタが存在する可能性が高いと判断する．interest_gain は，次元間の相関を評価するのに使用する指標である．より相関の高い部分空間ほどクラスタリングに有用であると判断する．interest_gain は，次元間の相関を表す指標である interest を用いて計算する指標で，部分空間を構成する次元集合から 1 次元取り除いた場合の interest と，取り除かない場合の interest との差の最大値（取り除く次元についての最大値）を表す．

部分空間を構成する次元を $s = \{d_1, \cdots, d_p\}$ とする．p は部分空間の次元数である．以下の条件を満たす部分空間を，クラスタリングに有用な部分空間と判断する．

$$H(s) < \omega \tag{2}$$

$$interest\_gain(s) > \epsilon' \tag{3}$$

$H(s)$ は部分空間 s のエントロピー，$interest\_gain(s)$ は interest_gain を表す．ω, ϵ' は定数である．

これらの指標を計算する準備段階として，特徴空間を複数の領域に分割し，それぞれの領域に含まれる事例の数を数え上げる．まず，すべての次元にそれぞれ等間隔に $k-1$ 個の区切りを設ける．各次元の区切りによって，特徴空間は k^p 個の領域に区切られる事になる．

こうして区切られた各領域をセルと呼ぶ．

部分空間 s のエントロピー，interest_gain は，以下のように計算される．

$$H(s) = \sum_{i=1}^{k^p} P_i^s \log P_i^s$$

$$P_i^s = \frac{insnum(c_i^s)}{N}$$

$$interest(s) = \sum_{i=1}^{p} H(\{d_i\}) - H(\{d_1, \cdots, d_p\})$$

$$interest\_gain(s) = interest(\{d_1, \cdots, d_p\}) - \max_i \{interest(\{d_1, \cdots, d_p\} - \{d_i\})\}$$

$$interest\_gain(s) = 0 \qquad (p = 1)$$

$c_1^s, \cdots, c_{p^k}^s$ は部分空間 s における各セルを，$insnum(c)$ はセル c に存在する事例の数，N は全事例数を表す．

次に，ENCLUS_INT の探索方法について述べる．ENCLUS_INT は，低次元の部分空間から高次元の部分空間へ，次元数を増やしながら指標を計算していく．具体的には，アプリオリアルゴリズムのように，1 次元から成る部分空間を組み合わせて 2 次元の部分空間を作り，それを組み合わせて 3 次元の部分空間を作る，といったように，徐々に多次元の部分空間を作りながら，クラスタリングに有用な部分空間を探索する．その際，探索する範囲をエントロピーによって制限している．具体的には，式 (2) を満たさない部分空間は，それ以上，他の部分空間との組み合わせを行わず，探索を打ち切るようにする．この時，エントロピーは次元を加えるごとに減少するので，式 (2),(3) を満たす全ての部分空間を見つける事が可能である．

次に，ENCLUS_INT の問題点について述べる．問題点は 2 つある．1 つ目は，部分空間のエントロピーによる評価である．エントロピーによる部分空間の評価は，同じ次元数の部分空間どおしならば，公平に評価できるが，次元数の異なる部分空間どおしでは，公平に評価することができない．

図 1 のような場合に，それが顕著に現れる． 図 1 における部分空間 (a) と部分空間 (b) は同じデータ集合，特徴空間を表していて，(a) は 1 次元の部分空間のセルに区切られており，(b) は (a) より 1 次元多い部分空間のセルに区切られている[⋆1]．人間が (a) と (b) を比較したとき，明らかに B の方がクラスタリングに適していると判断できる．しかし，エントロピーにより評価を行うと，(a) の方がクラスタリングに適していると判断されてしまう．ENCLUS_INT は，エントロピーによって部分空間の探索範囲を決めるため，ω の値によっては (b) が探索されなくなる可能性がある．つまり，よりクラスタリングに有用と考えられる部分空間を低く評価し，無視してしまう可能性がある．

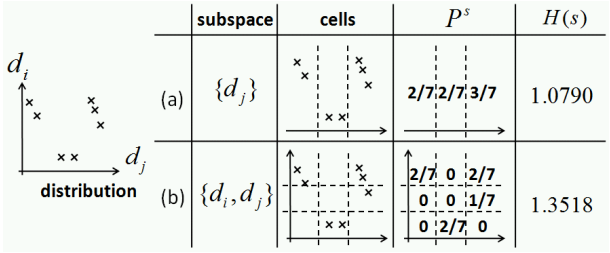

図 1 次元数の異なる部分空間におけるエントロピー

2 つ目の問題点は，互いにほとんど相関のない多次元空間を含む空間の評価である．ENCLUS_INT で相関の評価に使用している interest_gain は，部分空間の次元の中に，他の次元と相関のほとんどない次元がまじっている場合，低い値をとるように作られているため，そのような空間をクラスタリングに有用な空間と判断するのを防ぐ事ができる．しかし，図 2 のように，部分空間の次元が複数のグループに分かれており，グループ内部では次元間の相関が高く，グループ間での相関はほとんど無いような場合，interest_gain は低い値とならず，閾値 ϵ' の値次第では，その部分空間をクラスタリングに有用であると判断してしまう可能性がある．

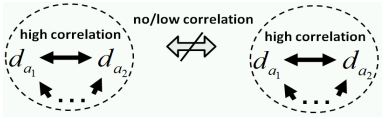

図 2 内部では相関が高く，互いには相関の無い多次元の部分空間

⋆1 1 次元の部分空間がクラスタリングに有用と判断されることはないが，わかりやすさのため 1 次元と 2 次元の例を示した．

上に示した問題点のうち，1 つ目の問題を解決するため，部分空間の評価にエントロピーを使用せず，以下のような条件を設ける．

$$interest\_gain() > \epsilon'' \tag{4}$$

ϵ'' は $\epsilon'' \leq \epsilon'$ を満たす定数である．探索範囲の制限は，エントロピーによる方法と同じように行う．すなわち，式（4）が満たされない部分空間であれば，それ以上，他の部分空間との組合せを行わず，探索を打ち切る．

interest_gain には，エントロピーのような部分空間を構成する次元を増やした場合に単調増加する性質がないので，式 (3) を満たす全ての部分空間を見つけることはできないが，式 (4) を設ける事によって，ENCLUS_INT の 2 つ目の問題点を緩和する事ができる．すなわち，探索過程において，次元数を一つずつ増やす度に式 (4) の判定を行い，相関のない次元が 1 次元追加されたタイミングで探索を打ち切ることができる．探索の際，式 (3) を満たす部分空間のうち，$interest\_gain(s) < interest\_gain(s'), s \in s'$ を満たす部分空間 s, s' では，s より多くの情報を持ち，より評価の高い s' の方が有用であると考え，s をクラスタリングに使用しない．以上のような変更を加えたアルゴリズムを使用し，部分空間を決定している．

3.4 クラスタリング

部分空間におけるクラスタ数は，事前にはわからないため，クラスタ数を自動的に判定する新たなクラスタリングアルゴリズムを用いる．クラスタリングには，部分空間の決定の段階で特徴空間に設けたグリッド状の分割を利用するため，1 つのクラスタは，複数のセルで構成される領域となる．

提案手法で使用するクラスタリングアルゴリズムは，クラスタの探索，不要なクラスタの除去の 2 段階に分かれている．まず，クラスタの探索について述べる． 図 3 にクラスタリングの探索の実行例を示す[⋆1]．クラスタを探索する際は，事例数の多いセルから順に処理していく．(図 3(a)) 隣接するセルにラベルがない場合は，新しいクラスタをクラスタのリストに追加し，そこに当該セルを追加する．そして，当該セルに新しく作ったクラスタのラベルを付加する．(図 3(b))

隣接するセルにクラスタのラベルがあり，その種類が 1 つの場合，当該セルをそのクラスタに追加し，当該セルにそのクラスタのラベルを付加する．(図 3(c))

隣接するセルのラベルの種類が複数の場合，新しいクラスタをリストに追加し，当該セルと，隣接するセルと同ラベルを持つ全てのセルに新しいクラスタのラベルを付与し，それらのセルを新しいクラスタに追加する．(図 3(d)) 以上の作業を事例の存在する全てのセルに実行する．(図 3(e))

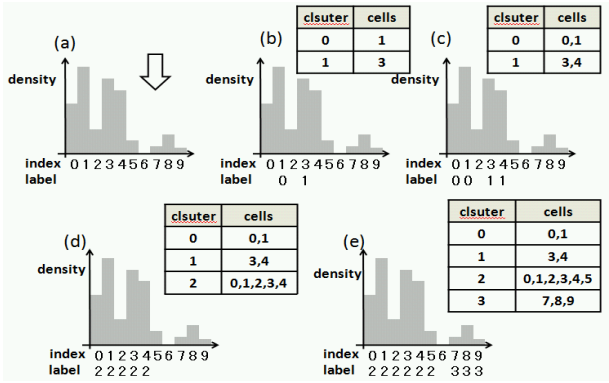

図 3 提案手法おけるクラスタリングのアルゴリズム

次に，不要なクラスタの除去について述べる．提案手法のようなクラスタの探索を行うと，偶発的に生まれた小さな密度の山をクラスタと判断したり，外れ値によって生まれる，事例数の極端に少ないセルを，クラスタと判断してしまう．このような小さなクラスタを除去するため，クラスタの大きさを表す指標を定義し，クラスタの間引きを行っている．クラスタの大きさを表す指標を式 (5) に示す．

$$size(C) = \frac{\sum_{c \in cell(C)} (insnum(c) - insnum(valley(C)))}{N} \tag{5}$$

N は全事例数，$cell(C)$ はクラスタ C に含まれるセルの集合，$valley(C)$ はクラスタ C と他のクラスタとの間の谷の位置にあるセルである，図 4(a) を C とすると，$valley(C)$ は図 4(c) の位置のセルにあたる．$size(C)$ は，$valley(C)$ の事例数に関係なく，クラスタの大きさを表す．図 4 を例にすると，クラスタ (a)，(b) は，事例数の合計が異なるが，$valley(C)$ の事例数を越えている分の合計は同じなので，$size(C)$ は同じ値になる．

3.5 目標ドメインへの適用

元データで行ったクラスタリングの結果に基づいて，目標ドメインのデータ表現を変換す

⋆1 実際は多次元の空間で行うが，理解の容易さのために 1 次元の例を示している．

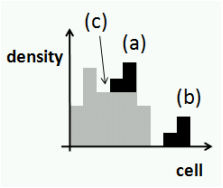

図 4 高密度の領域にあるクラスタと低密度の領域にあるクラスタ

る．まず，目標ドメインの事例に対してその事例の存在するセルを求める．次に，全てのクラスタについて，それぞれ 1 次元のデータ表現に変換する．この時，事例がクラスタのセルに含まれていた場合は元ドメインでクラスタリングを行った時のセルの事例数を出力し，そうでない場合は 0 を出力する．こうして出力したデータ表現を，データが持つ元々のデータ表現に追加して，一般的な学習器で学習を行う．

新しく追加したデータ表現によって，学習器は事例がどのクラスタに属しているかを知ることができる．クラスタは，その周囲に比べて高密度なので，クラスタの間は比較的低密度の領域となる．つまり，事例がどのクラスタに属しているかを知る事により，その事例が，クラスタの間にある低密度領域の，どちら側にあるかの区別をつける事ができるようになり，低密度領域に存在する決定境界を発見することが可能になる．また，事例の属するセルの事例数をデータ表現とし，事例の存在する領域の密度を知ることができるようになる．これによって，厳密にクラスタの境界を通る決定境界だけでなく，クラスタの境界から少しずれた領域にある決定境界も発見することが可能になる．

4. 既存手法との比較

まず，手法の持つ仮定について既存手法と提案手法を比較する．既存手法で用いるスパースコーディングは，データが少数のベースベクトルの線形和で表されると仮定している．また，目標ドメインでの精度向上の要因は，スパースコーディングを用いて学習したデータ表現が高度であることを挙げているが，データ表現が高度であるという事の意味については述べられていない．一方，提案手法では，目標ドメインに関して，仮定 (1) を置き，元ドメインと目標ドメインの関係に関しては，仮定 (2) を置いている．この 2 つ仮定が成立すると，元ドメインの部分空間における低密度の領域に関する情報を目標ドメインで使えば，密度の低い領域での境界を発見しやすくなり，目標ドメインでの精度向上につながる．

また，自己教示学習には，元ドメインと目標ドメインの分布が同一であるという仮定が必要ではないかという指摘がある．[5] 既存手法では，スパースコーディングの際，元ドメインと目標ドメインで同じ確率モデルを使用しているため，両者の分布の違いに明確に対応しているとは考え難い．提案手法では，仮定 (1) と仮定 (2) を設けているため，元ドメインの分布と目標ドメインの分布が異なる状況でも，自己教示学習による精度向上を期待できる．

次に，手法の持つ仮定について，既存手法と提案手法を比較する．既存手法は，データ表現の変換に繰り返し計算が必要であり，繰り返しが最も早く終了する場合で $O(s^2 + sn)$ の計算量が必要になる．なお s は元のデータ表現の次元数，n は新しいデータ表現の次元数である．提案手法は $O(sn)$ の計算量で変換する事ができる．また，データ表現の変換は，事例に対応するセルを求め，各クラスタに対して，そのセルに応じた事例数（クラスタに含まれなければ 0）を読み込むだけなので，繰り返し計算の必要がない．また，変換したデータ表現を学習に使用した場合，学習手法によっては予測に特定の次元のみが必要になり，他の次元は必要なくなることもあるが，既存手法では，常に $O(s^2 + sn)$ 以上の計算量が必要になる．一方，提案手法では，各次元の変換が独立しているため，n' 次元のデータ表現への変換には $O(sn')$ の計算量で変換する事ができる．

5. 実　　験

提案手法の転移学習としての効果を検証するため，人工データと実データで実験を行った．教師あり学習，提案手法での学習器は SMO を用いる．また，エラー率は，訓練事例，テスト事例のランダムサンプリングを 30 回繰り返し，平均エラー率により求める．

人工データとして，クラスタが存在する 2 つの 2 次元部分空間と，クラスタの存在しない (一様分布) 次元 10 次元からなる部分空間，計 14 次元からなるデータを用意する．(図 5) そして，図中の 2 つのクラスタ (図 5(a) と図 5(b)) の両方に属する事例を正例，そうでない事例を負例とする 2 値のクラスを与える．

実験結果を表 1 に示す．検定により有意差が認められたものを太字で示している．提案手法を適用した結果，教師あり学習と比較して，エラー率が低下している．意図した部分空間とクラスタを発見し，2 つの部分空間で 4 つずつ，合計 8 個のクラスタを発見した．訓練事例数を変化させての実験も行った．理想的な条件では，事例数を増やした場合でも，エラー率の低下の効果が変らない事が確認できた．

実データの実験には，手書き数字画像，手書きアルファベット文字画像，フォントアルファベット文字画像を使用する．データ表現は，画像の各ピクセルの黒さを使用する．

手書きアルファベット文字とフォントアルファベット文字の実験では，エラー率の低下に

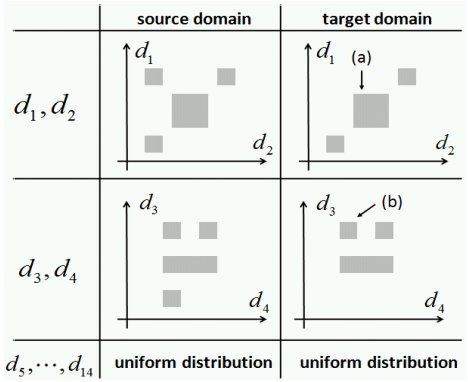

図 5 人工データ

クラス毎の訓練事例数	教師あり学習	自己教示学習
1	0.4399	**0.4050**
10	0.1442	**0.1048**
100	0.0463	**0.0016**

表 1 人工データの実験結果（エラー率）

成功した．エラー率低下の原因とて，目標ドメインと同じアルファベットのデータでクラスタリングを行うので，アルファベットを見分けるのに適したクラスタを学習できた事が考えられる．一方，手書き数字と手書きアルファベット文字の実験では，エラー率の低下はほとんどみられなかった．これは，元ドメインを数字にしたため，数字を区別するのに適したクラスタを学習し，目標ドメインであるアルファベットの区別をつけるのにはあまり役立たなかったものと考えられる．[⋆1]この実験のように，アルファベット以外を元ドメイン，アルファベットを目標ドメインとする問題設定の場合，エラー率をより低下させるには，アルファベットと共通する特徴 (線や文字の概形など) を表すようなクラスタを学習する必要があると考えられる．そのためには．特定の文字集合に特化させないために，元ドメインのデータに多様な文字データを使用するか，元ドメインにアルファベットに似た特徴を持つデータを含める等が効果的であると考えられる．

⋆1 元ドメインと目標ドメインを両方手書き数字にすると，アルファベットを目標ドメインにした時以上のエラー率低下が見られた

元ドメイン	目標ドメイン	教師あり学習	提案手法
手書き数字画像	手書きアルファベット画像	0.8519	**0.8510**
手書きアルファベット画像	フォントアルファベット画像	0.6304	**0.6165**

表 2 実データの実験結果（エラー率）

6. おわりに

本研究では，未知データに対する計算量を低減した自己教示学習を手法を提案した．本研究では，部分空間クラスタリングを自己教示学習に適用して，理論的な計算量を既存手法に比べて小さくする事ができた．また，実験結果より，手法が仮定する理想的なデータと，実データにおいて自己教示学習の効果を発揮する事を確認した．今後の課題としては，さらに多様なデータでの実験とそれぞれのドメインにおける仮定の検証，パラメータの自動最適化する手法の考案などが挙げられる．提案手法では，部分空間クラスタリングのパラメータが学習時間に大きく影響するという問題がある．部分空間クラスタリングでは，パラメータによって探索範囲が大きく変わるため，計算時間も比例して大きく上下する．こうした理由から，学習を繰り返し，多くのパラメータを試しながら調節するのは時間的に困難である．このため，より実用的な手法にするためには，データから適したパラメータを推定する手法が必要になる．

参 考 文 献

1) Bruno, A.O. and David, J.F., Emergence of simple-cell receptive field properties by learning a sparse code for natural images, Nature, Vol. 381, No. 381, pp. 607-609 (1996).
2) Honglak, L., Alexis, B., Rajat, R. and Andrew, Y.N.: Efficient sparse coding algorithm, Advances in NIPS, Vol. 19, pp. 801-808 (2007).
3) Rajat, R., Alexis, B., Honglak, L., Benjamin, P. and Andrew, Y.N.: Self-taught learning: Transfer learning from unlabeled data, Proc. of 24th ICML, pp. 759-766 (2007).
4) Wenyuan, D., Qiang Y., Gui-Rong X. and Yong Y.: Boosting for transfer learning, Proc. of 24th ICML pp. 193-200 (2007).
5) 神嶌敏弘: 転移学習のサーベイ, 人工知能学会研究会, 第 9 回データマイニングと統計数理研究会 (2009).
6) Oliver C., Bernhard S. and Alexander Z., Semi-Supervised Learning, MIT Press (2006).