

Online Passive-Aggressive Algorithm を用いた クラスタリング

山本 悠二^{†1} 増山 繁^{†1}

オンライン学習は、逐次取り出したデータに対するパラメータ更新を行うだけでよい。そのため、実行時間やメモリ使用量の観点で効率がよく、特に教師ありのクラス分類問題を解くために広く使われ始めている。本稿では、逐次的に取り出したデータ点に対してのマージンを更新することにより、クラスタリングを行う手法を提案する。この手法は最大マージンクラスタリングの発見的解法に当たる。実験では標準的なデータセットを用いて提案手法の有効性について評価した。

Clustering via Online Passive-Aggressive Algorithm

YUJI YAMAMOTO^{†1} and SHIGERU MASUYAMA^{†1}

Online learning has been widely used in supervised classification problems, because its update rule of a given data point is simple and efficient in both computing time and memory space. In this paper, we propose a clustering method to retain a margin for a given data point in an online fashion. This method corresponds to a heuristic solution for maximum margin clustering. We evaluated its effectiveness in standard datasets.

1. はじめに

クラスタリングとは、ある一定の基準にしたがって、ラベルなしのデータ点を、かたまり(クラスタ)にまとめ上げる技法である。この方法は、データに内在する構造や概念を人間が発見できるようにすることや、まとめ上げたデータを別の機械学習手法と組み合わせるの

に用いられている。代表的な手法として、 k -means 法やスペクトラルクラスタリング、混合分布に基づくものが提案されている。以降では、2つのクラスタに分割するクラスタリング手法に限定して論じる。この場合にも先ほど述べた手法がよく用いられているが、別の手法として最大マージンクラスタリング¹⁾(Maximum Margin Clustering; 以降 MMC と略記)がクラスタリングとしての性能の点で特に優れていることから、研究がさかんに行われている^{*1}。

MMC は、与えられたデータ集合に対してマージンが最大となる分離超平面を求めることでクラスタリングを行う手法である。この手法は Support Vector Machine の定式化と類似するが、以下の2点が大きく異なる。

まず1点目に、MMC はクラスタリング手法であるため、ラベルが与えられていない。そのため、使用する最適化問題の種類は、凸二次計画問題の SVM とは異なり、非凸二次計画問題と一般に計算が困難な問題に属する。したがって、MMC の最適化問題は厳密解ではなく近似解を求める手法について研究されている。

2点目に、MMC では、データ点が各クラスタに分離されるようにバランス制約を置いている点である。このような制約を置いている理由は、「すべてのデータ点に対してマージンが最大となる分離超平面」を求める場合の自明な解は、どれか一つのクラスタにすべてのデータ点が属することである。このような解になることを避けるために、MMC ではデータ点が分離超平面を境にして空間的に均一に分布するようにバランス制約を設けている(詳細な説明は2節で述べる)。

本稿では、MMC に対する発見的解法として、逐次的に取り出したデータ点に対してのマージンを更新することによりクラスタリングを行う手法を提案する。具体的には、取り出したデータ点において、Online Passive-Aggressive Algorithm を用いてデータ点に対するパラメータの更新を行う。ただし、このパラメータ更新する場合は、データ点を +1 もしくは -1 のどちらのラベルに分類させるかが問題となる。そこで、どちらのラベルを割り当てた場合に、よりバランス制約を満たすかを評価して、ラベルを定める。このようなオンライン学習の枠組みによる方法は、他の MMC の解法で用いられているオフライン学習の枠組みと比較して、複数のデータ点に関する制約を持つ最適化問題を考える必要がない分、実行時間やメモリ使用量の観点で効率が良い。

^{†1} 豊橋技術科学大学 知識情報工学系
Department of Knowledge-based Information Engineering, Toyohashi University of Technology

^{*1} なお、マージン最大化基準による、3クラスタ以上のクラスタリング手法は、多クラス最大マージンクラスタリング²⁾(Multi-class Maximum Margin Clustering) と呼ばれる。

本稿の構成は以下の通りである．まず，2節で MMC の定式化について説明を行う．次に，3節で教師ありのオンライン学習である Online Passive-Aggressive Algorithm について説明を行う．そして，4節で Online Passive-Aggressive Algorithm によるクラスタリングを提案する．5節の実験では，MMC の先行研究で広く用いられている UCI Repository のデータセットを用いて提案手法の有効性について論じ，6節でまとめを行う．

2. 最大マージンクラスタリング

この節では，最大マージン原理に基づいた教師あり学習である SVM の定式化について述べた後，その拡張である MMC について説明する．

まず，教師あり学習の問題設定として n 個の「データ点集合とそのラベル」の対の集合 $T = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ (ただし， $\mathbf{x}_i \in \mathbf{R}^d$ ， $y_i \in \{-1, +1\}$) が入力として与えられる．このとき SVM は，以下の最適化問題を解くことにより，識別関数 $y = \mathbf{w}^T \phi(\mathbf{x})$ を得る^{*1}．

$$\min_{\mathbf{w}, \xi_i} \quad \frac{1}{2} \|\mathbf{w}\|^2 + \frac{C}{n} \sum_{i=1}^n \xi_i$$

subject to

$$\begin{aligned} y_i \mathbf{w}^T \phi(\mathbf{x}_i) &\geq 1 - \xi_i \\ \xi_i &\geq 0 \end{aligned}$$

ただし，定数 $C (> 0)$ はコストパラメータを表し，値が大きくなるほどハードマージン，値が小さくなるほどソフトマージンになる．また，関数 $\phi(\cdot)$ は特徴ベクトルを高次元に写像する関数を表す．

一方，MMC は，教師なし学習であるため， n 個のデータ点集合 $U = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ ($\mathbf{x}_i \in \mathbf{R}^d$) のみが入力として与えられる．そして，以下の最適化問題^{*2}を解くことで，識別関数 $y = \mathbf{w}^T \phi(\mathbf{x})$ を得る．

$$\min_{\mathbf{w}, \xi_i} \quad \frac{1}{2} \|\mathbf{w}\|^2 + \frac{C}{n} \sum_{i=1}^n \xi_i$$

subject to

$$\begin{aligned} |\mathbf{w}^T \phi(\mathbf{x}_i)| &\geq 1 - \xi_i \\ \xi_i &\geq 0 \\ \frac{1}{n} \left| \sum_{j=1}^n \mathbf{w}^T \mathbf{x}_j \right| &\leq l \end{aligned}$$

制約条件の1番目は，データ点 \mathbf{x}_i が，原則，1以上のマージンを取るようになるものである．また，制約条件の3番目はバランス制約 (balance constraint) と呼ばれるものである．なお，制約式中の l は十分小さい正の定数である．この条件を入れずに，単純に「すべてのデータ点に対してマージンが最大となる分離超平面を取る」という問題を解くと，データ点のすべてのラベルが -1 のみ，もしくは $+1$ のみになるような自明な解が得られる．このような解はクラスタリングとして意味がない．そのため，MMC の最適化問題では，「分離超平面から正例側のデータ点へのマージンの総和と，分離超平面から負例側のデータ点へのマージンの総和が大体同じになるように分離超平面を定める」という制約を加えている．

3. Online Passive-Aggressive Algorithm

オンライン学習とは，データが1つずつ逐次的に与えられるという設定の下で，与えられたデータに基づいて機械学習のパラメータを更新するものである．このような学習方法は，逐次取り出したデータに対するパラメータ更新を行うだけでよいから，実行時間やメモリ使用量の観点で効率がよく，特に教師ありのクラス分類問題を解くために広く使われている．

Online Passive-Aggressive Algorithm⁴⁾ は，先に挙げたオンライン学習による教師あり学習の一手法であり，2クラス分類問題に限らず，多クラス分類問題や回帰問題の学習へ拡張が可能な手法となっている．また，実験的には，SVM のようなオフラインの機械学習に比べて，ほぼ同等の性能が得られることが多いから，自然言語処理などの大規模データを扱う場合によく用いられる⁵⁾．ここでは，2クラスの分類問題についての Online Passive-Aggressive Algorithm について説明する．

まず，2節の冒頭で述べた教師あり学習の問題設定と同様に， n 個の「データ点集合とそのラベル」の対の集合 $T = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ が入力として与えられる．そして，オンライン学習によって最終的に更新された重みベクトル \mathbf{w} を用いて，識別関数 $y = \mathbf{w}^T \phi(\mathbf{x})$ を得る．更新の手順は次の通りである．まず，初期重みベクトル $\mathbf{w}_1 = \mathbf{0}$ を設定する．そして，各

*1 以降，説明の簡便さから定数項を省いた識別関数を求めることを考える．

*2 論文 3) で示された MMC の定式化である．

データ点 \mathbf{x}_i ($i = 1, \dots, n$) を順に取り出し, \mathbf{x}_i のラベルが -1 であるか $+1$ であるかを予測した後, 正解ラベル y_i を比較して重みベクトルを更新する. 重みベクトルを更新するための最適化問題は以下のように表される.

$$\mathbf{w}_{i+1} = \arg \min_{\mathbf{w}, \xi} \frac{1}{2} \|\mathbf{w} - \mathbf{w}_i\|^2 + C\xi$$

subject to

$$y_i \mathbf{w}^T \phi(\mathbf{x}_i) \geq 1 - \xi$$

$$\xi \geq 0$$

なお, 定数 C (> 0) は SVM の定式化と同様にコストパラメータを表し, 値が大きくなるほどハードマージン, 値が小さくなるほどソフトマージンになる. この最適化問題は解析的に求まり, 以下のような更新則になる.

$$\mathbf{w}_{i+1} = \mathbf{w}_i + \tau \phi(\mathbf{x}_i)$$

$$\text{ただし, } \tau = \min\{C, \max\{0, \frac{l_i}{\phi(\mathbf{x}_i)^T \phi(\mathbf{x}_i)}\}\}, l_i = \max\{0, 1 - y_i \mathbf{w}_i^T \phi(\mathbf{x}_i)\}$$

更新則に注目すると, データ点 \mathbf{x}_i が正しく予測でき, かつ, 十分なマージン ($|\mathbf{w}_i^T \phi(\mathbf{x}_i)| \geq 1$) を持つ場合は, $\tau = 0$ となり, \mathbf{x}_i が与えられる前の重みベクトルが継続して用いられる. しかし, 間違えて予測した, もしくは, 正しく予測が行えたが十分なマージンが得られなかった場合は, $\tau > 0$ となり, 重みベクトルは \mathbf{x}_i に対して十分なマージンを持って正しく予測されるように重みベクトルが更新される. このように, 学習の際に, 重みベクトルに対して Passive な更新と Aggressive な更新があることから, Passive-Aggressive Algorithm と呼ばれる.

4. Online Passive-Aggressive Algorithm によるクラスタリング

4.1 基本的なアイデア

MMC の発見的解法として, Online Passive-Aggressive Algorithm を用いて, ラベルなしデータ集合 $U = \{\mathbf{x}_i\}_{i=1}^n$ に対してクラスタリングを行うことを考える. 問題設定は, 3 節と同じくデータが 1 つずつ逐次的に与えられるという設定である. ここで, 与えられたデータ点を \mathbf{x}_i , \mathbf{x}_i が与えられた時点での重みベクトルを \mathbf{w} とする. 2 節で述べたように, MMC では, データ点に対するマージンが, 原則, 1 以上 (つまり, $|\mathbf{w}^T \phi(\mathbf{x}_i)| \geq 1$) を取るように, 重みベクトル \mathbf{w} を最適化する. そこで, 重みベクトルの Aggressive な更新の対

象を, データ点 \mathbf{x}_i のマージン $|\mathbf{w}^T \phi(\mathbf{x}_i)|$ が 1 より小さいものにする. そして, Aggressive な更新を行う場合, 更新した重みベクトルを用いた識別関数が データ点 \mathbf{x}_i について, ラベル -1 を出力させるようにするか, $+1$ を出力させるようにするかを決定する必要がある. そこで, -1 のラベルを取ると仮定してベクトルの更新を行った場合と, $+1$ のラベルを取ると仮定してベクトルを更新した場合に, どちらがよりバランス制約を満たすかについて調べ, よりバランス制約を満たすほうのラベルを定める. このようにベクトルの更新を行うことで, バランス制約をできるだけ満たしながら逐次的にデータ点のマージンを広げてクラスタリングを行うことが期待できる.

4.2 提案手法

4.1 節で述べたアイデアに基づき, Online Passive-Aggressive Algorithm によるクラスタリングの擬似コードを図 1 に示す.

なお, コード中にある関数 $\text{shuffle}(U)$ はデータ集合の要素をシャッフルしたものを返す関数である. これは, ベクトルの更新の際にできるだけ局所解に陥ることを避けるためにランダム性を加えるのに用いる.

以下に擬似コードの概要を示す. まず, 8 行目でデータセットを I 巡らせて, 以降の重みベクトルの更新の処理を行う. 11 行目でシャッフルしたデータからデータ点を取り出す. そして, 12 行目で, そのデータ点が 1 以上のマージンが確保できているか否かを調べる. もし, データ点 \mathbf{x} に対するマージンが 1 未満である場合は $l > 0$ となるため, 13 行目の条件分岐により 14 行目から 23 行目の重みベクトルの更新に進む. この処理では, 重みベクトルの更新によって, データ点 \mathbf{x} のラベルが $+1$ を持つようにするか -1 を持つようにするかを決定する必要がある. そこで, どちらのラベルを付与した場合に, よりバランス制約を満たすかについて調べる. ここで, 15 行目において, \mathbf{x} のラベルに $+1$ を付与して更新された重みベクトルが $\mathbf{w} + \tau \phi(\mathbf{x})$ になることに注目する. すると, この行では, 変数 psum に, \mathbf{x} のラベルに $+1$ を付与して重みベクトルを更新した場合に, その重みベクトルを用いた識別関数によるデータ全体のスコアの総和を表す. 同様に, 16 行目では, -1 のラベルを付与して重みベクトルを更新した場合に, その重みベクトルを用いた識別関数によるデータ全体のスコアの総和を変数 nsun に代入している. バランス制約は, 識別関数によるデータ全体のスコアの総和の絶対値が小さいほど, より適切であるとみなす. よって, 17 行目の条件分岐によって, よりバランス制約を満たすラベルを付与することができる. そして, ベクトルの更新を 23 行目で行う.

このように, Online Passive-Aggressive Algorithm を用いたクラスタリングは, 一般に

```

1: // 入力:  $U = \{\mathbf{x}_i\}_{i=1}^n$ : ラベルなしデータ集合
2: //       $I$ : 訓練事例集合単位での反復回数
3: //       $C$ : コストパラメータ
4: // 出力:  $\mathbf{w}$ : 重みベクトル
5: function train(  $U, I, C$  )
6:  $\mathbf{w} = \mathbf{0}$ ;
7:  $n = |U|$ ; // ラベルなしデータ集合の個数を格納
8: for (iter = 1; iter <= I; iter++) {
9:    $SU = \text{shuffle}(U)$ ; // データ集合をシャッフル
10:  for (j = 1; j <= n; j++) {
11:     $\mathbf{x} = SU^{(j)}$ ; // データを取り出す
12:     $l = \max\{0, 1 - |\mathbf{w}^T \phi(\mathbf{x})|\}$ ;
13:    if ( $l > 0$ ) { // マージン違反がある場合 (aggressive な更新)
14:       $\tau = \min\{C, \frac{l}{\phi(\mathbf{x})^T \phi(\mathbf{x})}\}$ ;
15:       $\text{psum} = \sum_{i=1}^n \{\mathbf{w} + \tau \phi(\mathbf{x})\}^T \phi(\mathbf{x}_i)$ ;
16:       $\text{nsum} = \sum_{i=1}^n \{\mathbf{w} - \tau \phi(\mathbf{x})\}^T \phi(\mathbf{x}_i)$ ;
17:      if ( $|\text{psum}| < |\text{nsum}|$ ) {
18:         $y = +1$ ;
19:      }
20:    } else {
21:       $y = -1$ ;
22:    }
23:     $\mathbf{w} = \mathbf{w} + y \tau \phi(\mathbf{x})$ ;
24:  }
25: }
26: }
27: return  $\mathbf{w}$ ;

```

図 1 擬似コード: Online Passive-Aggressive Algorithm によるクラスタリング
Fig. 1 Pseudo code: Clustering via Online Passive-Aggressive Algorithm

MMC の最適化に使われている二次計画問題や半正定値計画問題に対するソルバーを介す必要がないため、実装が容易に行える。特に使用するカーネルを線形に限定する場合、図 1 の擬似コードは $O(In)$ (I : 訓練事例集合単位での反復回数, n : データ数) で動作するため^{*1}, 他の MMC の最適化手法に比べて高速に動作することが見込める^{*2}。

4.3 解候補集合の作成と準最適解の選択

4.2 節で示した擬似コードで求まる重みベクトル \mathbf{w} は、場合によっては局所最適解に陥るなどの理由でクラスタリング性能として不十分であるものがある。そこで、 J 個の重みベクトル候補 $\{\mathbf{w}_1^{cand}, \dots, \mathbf{w}_J^{cand}\}$ を求めて、その中からクラスタリング性能が最良のものを選択する。クラスタリング性能の評価は 2 節で示した MMC の最小化問題の目的関数の値を最も小さくする重みベクトルとする。つまり、以下の式を満たす重みベクトルを選択する。

$$\mathbf{w} = \arg \min_{\mathbf{w}' \in \{\mathbf{w}_1^{cand}, \dots, \mathbf{w}_J^{cand}\}} \frac{1}{2} \|\mathbf{w}'\|^2 + \frac{C}{n} \sum_{i=1}^n \max\{0, 1 - |\mathbf{w}'^T \phi(\mathbf{x}_i)|\}$$

5. 実験

5.1 実験設定

UCI Repository の ionosphere, digits, letter, satellite にて提案手法の評価を行った。データセットの使用方法は論文 6) と同じで、以下の通りである。digits のデータについては弁別が難しいとされる数値の対 3 vs 8, 1 vs 7, 8 vs 9 を使用した。また、letter は最初の 2 クラスである A vs B を使用した。satellite についても同様に最初の 2 クラス C1 vs C2 を使用した。

提案手法で定める必要のあるパラメータとしては、コストパラメータ C (4.2 節を参照)、訓練事例集合の反復回数 I (4.2 節を参照)、解候補の個数 J (4.3 節を参照) がある。本実験では、 $I = 10$, $J = 10$ に固定した。また、最適な C はデータセットに大きく依存すると考えたため、 $C = \{0.1, 1, 10, 100\}$ のうちで最良の結果になったものを採用した。

提案手法ではカーネルの選択ができるため、線形カーネルと Gaussian カーネル $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}') = \exp(-\|\mathbf{x} - \mathbf{x}'\|^2 / \sigma^2)$ の両方を試して最良の結果になったものを採用している。ただし Gaussian カーネルに使用する場合は、さらにパラメータ $\sigma = \{2D, 3D, 4D, 5D\}$ (ただし、 $D = (\sum_{k=1}^d [\max\{X^k\} - \min\{X^k\}]^2)^{1/2}$, $\{X^k\}$ はデータセット中の k 番目の

*1 ただし、擬似コードの 15 行目, 16 行目で現れる $\sum_{i=1}^n \mathbf{x}_i$ を事前に計算しておくといったコードの書き換えが必要である。

*2 同じく時間計算量がデータ数に比例する MMC の最適化手法として論文 3) があるが、切除平面法を用いて複数回の二次計画問題を解く部分のオーバーヘッドがあるため、オンライン学習手法のほうが効率的であると考えられる。

属性値を表す)のうちで最良の結果になったものを採用した。

5.2 結 果

5.1 節に基づいてパラメータとカーネルを選択して結果を求めた。それぞれのデータセットについてのパラメータとカーネルの選択については表 1 に示している。

提案手法、および、他の手法の性能を測る指標として、論文 6) 等と同様にエラー率を用いる。ここでエラー率とは、クラスタリングによって二分されたクラスターについて、元々あったラベルと合わせた場合にどれくらいの数の不一致が生じるかを割合で表したものである。

表 2 には、提案手法と、MMC のオフラインの最適化により局所最適解を求める手法 (IterSVR, CPMMC, EVOMMC) のエラー率を併せて示している。なお、提案手法のエラー率は、10 回の試行のエラー率の平均である。また、IterSVR の結果は論文 7) から、CPMMC は論文 3) から、EVOMMC は論文 8) からの引用である。

オンライン最適化による提案手法と、オフラインの最適化による手法 (IterSVR, CPMMC, EVOMMC) を比較する。若干、提案手法のほうがエラー率が高いが、その差はほとんどデータセットで 2% 以内に収まる。このことから、オンラインの最適化による MMC の発見的解法は、オフラインの最適化に比べて大差がない場合が多いといえる。

ただし、データセット Ionosphere について見ると、提案手法と EVOMMC ではエラー率が 10% 以上も差が開いている。また、IterSVR や CPMMC と EVOMMC の比較でも、同じく 10% 以上の差がある。IterSVR や CPMMC は、非凸最適化を convex-concave procedure⁹⁾ により凸最適化を解く計算を反復適用することで局所最適解を求める一方、EVOMMC では進化的アルゴリズムを用いて割り当てるラベルを積極的に変化させて最適化問題を解いている。EVOMMC のエラー率が低く抑えられた理由は、メタ戦略による最

適化が MMC において有効に働いたためであると考えられる。このようなメタ戦略に基づく最適化をオンライン学習の枠組みでも組み入れることができるかについては興味深い課題である。

6. ま と め

本稿では、MMC の発見的解法として、逐次的に取り出したデータ点に対してのマージンを更新することにより、クラスタリングを行う手法を提案した。実験から、提案手法とオフラインによる MMC の局所最適解を求める手法では、性能に関して大差がない場合が多いことがわかった。

今後の課題として、提案手法の多クラス化、および、メタ戦略への適用が挙げられる。

謝辞 本研究は文部科学省グローバル COE プログラム「インテリジェント センシングのフロンティア」の援助により行われた。

参 考 文 献

- 1) L. Xu, J. Neufeld, B. Larson, and D. Schuurmans.: Maximum Margin Clustering, *Proc. of Neural Information Processing Systems (NIPS)*, 2004.
- 2) B. Zhao, F. Wang, C. Zhang.: Efficient MultiClass Maximum Margin Clustering, *Proc. of International Conference on Machine Learning (ICML)*, 2008.
- 3) B. Zhao, F. Wang, C. Zhang.: Efficient Maximum Margin Clustering via Cutting

表 1 それぞれのデータセットで提案手法が採用したパラメータ
Table 1 Parameters adopted in the proposed method on each dataset.

データセット	データ数	C	使用したカーネル
Digits 3 vs 8	357	100	線形
Digits 1 vs 7	361	10	Gaussian (2D)
Digits 2 vs 7	356	1	線形
Digits 8 vs 9	354	10	線形
Ionosphere	351	1	Gaussian (2D)
Letter	1555	10	Gaussian (2D)
Satellite	2236	0.1	線形

表 2 各手法のエラー率 (%)
Table 2 Clustering errors on each dataset.

データセット	提案手法	IterSVR	CPMMC	EVOMMC
Digits 3 vs 8	3.70 ± 1.62	3.4 ± 0	3.08	2.52 ± 0
Digits 1 vs 7	0.03 ± 0.24	0 ± 0	0.0	0.0 ± 0
Digits 2 vs 7	0.08 ± 0.20	0 ± 0	0.0	0.0 ± 0
Digits 8 vs 9	4.63 ± 3.56	3.7 ± 0	2.26	3.22 ± 0.40
Ionosphere	29.32 ± 1.38	28.2 ± 0	27.64	17.94 ± 6.84
Letter	5.64 ± 0.21	7.2 ± 0	5.53	3.67 ± 0.33
Satellite	1.28 ± 0.06	3.6 ± 0	1.52	1.14 ± 0.65

- Plane Algorithm, *Proc. of SIAM International Conference on Data Mining*, 2008.
- 4) K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, Y. Singer.: Online Passive-Aggressive Algorithms, *J. of Machine Learning Research*, Vol. 7, pp. 551-585, 2006.
 - 5) D. Okanojima, J. Tsujii.: A Discriminative Language Model with Pseudo-Negative Samples, *Proc. of Annual Meeting of the Association of Computational Linguistics (ACL)*, 2007.
 - 6) K. Zhang, I. W. Tsang, J. T. Kwok.: Maximum Margin Clustering Made Practical, *Proc. of International Conference on Machine Learning (ICML)*, 2007.
 - 7) K. Zhang, I. W. Tsang, J. T. Kwok.: Maximum Margin Clustering Made Practical, *Trans. IEEE on Neural Networks*, Vol. 20, No. 4, pp. 583-596, 2009.
 - 8) F. Gieseke, T. Pahikkala, O. Kramer.: Fast Evolutionary Maximum Margin Clustering, *Proc. of International Conference on Machine Learning (ICML)*, 2009.
 - 9) A. L. Yuille.: CCCP Algorithms to Minimize the Bethe and Kikuchi Free Energies: Convergent Alternatives to Belief Propagation, *J. of Neural Computation*, Vol. 14, Issue 7, pp. 1691-1722, 2002.
-