2006 年情報論的学習理論ワークショップ
2006 Workshop on Information-Based Induction Sciences (IBIS2006)
Osaka, Japan, October 31 - November 2, 2006.

制約を修正に用いた半教師有りクラスタリング

Semi-supervised Clustering through Modification using Constraints

新納浩幸* 佐々木稔† 村上浩司‡
Hiroyuki Shinnou Minoru Sasaki Koji Murakami

Abstract: In this paper, we propose a new semi-supervised clustering method. This method first conducts a clustering without constraints, then modifies the clustering result by using constraints. This method can choose any clustering algorithm in the first step, which is most suitable for the given data. Furthermore, the method is robust for the given constraint. In the experiment, we use the Net news articles (395 documents, 5 categories) and Wine data (178 instances, 3 categories) in the UCI repository. The experiment shows that our method is more effective than conventional methods, COP-kmeans and CCL.
Kewword: clustering, semi-supervised, modification, constraint

1 はじめに

本論文では半教師有りクラスタリングの新しい手法を提案する。まず与えられた制約を用いずにクラスタリングを行い、次に制約を使ってそのクラスタリング結果を修正してゆく手法である。

半教師有りクラスタリングとはクラスタリングの対象のデータのいくつかに、クラスターに関する情報を与えた上でクラスタリングを行うタスクである。より現実的なクラスタリングのタスクとして近年注目されている [4]。与える情報がクラスターのラベルであれば Seed 型のクラスタリングや分類問題と同じタスクとなる。通常はペアのデータを複数個選び、それらペアのデータにペアのデータが「同じクラスターに属する」(must-link) や「異なるクラスターに属する」(cannot-link) という制約を与える。

従来の半教師有りクラスタリングの手法は制約ベースの手法と距離ベースの手法に大別できる [2]。制約ベースの手法とは通常のクラスタリングの目的関数に制約項を含めた新たな目的関数を定義し、与えられた制約を満足するようにクラスタリングを行う手法である [8][1]。代表的な研究として Wagstaff らの提案した COP-kmeans (Constrined K-means) という手法がある [8]。そこではデータが制約を満たすように k-means でクラスタリングを行う。また距離ベースの手法とは、データ間の距離を、制約を考慮した形で再計算し、その距離を使って通常のクラスタリングを行う手法である [6][9]。代表的な研究として Klein らの提案した CCL (Constrined Complete-link) という手法がある [6]。そこでは must-link の制約を持つデータ間の距離を 0、cannot-link の制約を持つデータ間の距離を ∞ とし、更に must-link に関連したデータの距離を適切に修正する。最終的にデータ間の距離行列を作成して、その行列を使ってボトムアップのクラスタリング手法である complete-link [5] でクラスタリングを行う。

これら従来の手法にはいくつか問題がある。まず制約ベースの手法では利用するクラスタリングの手法が限定されてしまう。クラスタリングの手法は多々あり、どれを利用したらよいかは問題に依存する。そのためある問題で有効な半教師有りクラスタリングの手法であっても、別種の問題に適用してよい結果が得られるとは限らない。距離ベースの手法は、クラスタリング手法を基本的には問わないが、CCL の場合には cannot-link の制約を有効に働かせるために complete link を用いる必要がある。また制約ベースの手法や距離ベースの手法どちらであっても、制約にセンシティブであるという問題がある。つまり制約として選ばれるペアのデータによってクラスタリング結果が大きく異なってきてしまう。場合によっては制約を全く使わずにクラスタリングした方が

*茨城大学工学部情報工学科, 316-8511 茨城県日立市中成沢町 4-12-1, shinnnou@mx.ibaraki.ac.jp,
Department of Computer and Information Sciences, Ibaraki University, 4-12-1 Nakanarusawa, Hitachi, Ibaraki 316-8511
†同上, sasaki@cis.ibaraki.ac.jp,

‡東京工業大学イノベーションシステム研究センター, 226-8303 横浜市緑区長津田町 4259 番地 S1, murakami@iri.titech.ac.jp,
Center for Innovation Systems Research, Tokyo Institute of Technology, 4259-S1 Nagatsuda, Midori-ku, Yokohama 226-8303

よい結果になる場合もある。本質的には、これは従来の手法が制約に対するデータの近傍に対しても、制約を適用させようとするからである。

これら問題を解決するために、本論文では制約をクラスタリングの修正に利用することを提案する。まずその問題に適したクラスタリング手法を利用してクラスタリングを行う。次に与えられた制約を満たすように、そのクラスタリング結果を修正してゆく。本手法ではベースとなるクラスタリングの手法を問わない。また本手法は、最初に得られるクラスタリング結果に矛盾する制約を、できるだけ少ないデータで満たそうとする。このために選ばれたペアのデータによらずにある程度のパフォーマンスを示す。

実験ではネットニュース記事（395 文書、5 カテゴリ）と UCI repository [1] の Wine データ（178 データ、3 カテゴリ）を利用した。COP-kmeans と CCL との比較実験を行い、それらの手法よりも良い結果を得ることができた。

2 制約によるクラスタリングの修正

本論文で提案する手法は、まず制約を利用せずにクラスタリングを行い、その次に制約を利用して先のクラスタリング結果を修正してゆく。その修正の処理の際には、各データにクラスター番号が与えられていることを注意しておく。当然、このクラスター番号は誤っていることもある。

与えられた制約が第 1 段階のクラスタリング結果と矛盾しないならば、その制約は利用されない。その制約はなかったものと同じ扱いになる。逆に与えられた制約が第 1 段階のクラスタリング結果に矛盾した場合に、クラスタリング結果を修正する必要がある。この修正は must-link の制約による修正と cannot-link の制約による修正のどちらかである。どちらの場合でも、ある 2 つのデータ間の制約であり、本手法では、一方のみのデータのクラスター番号を変更することで制約を満たすようにする。修正する方のデータをどのように判定するか、また、どのクラスター番号に変更するかが問題である。

2.1 must-link からの修正

データ e_1 と e_2 間の must-link の制約が第 1 段階のクラスタリング結果に矛盾する場合、第 1 段階のクラスタリング結果では、データ e_1 がクラスター C_1 に属し、データ e_2 がクラスター C_2 に属し、しかも $C_1 \neq C_2$ という状況になっている。データ e_1 のクラスターを C_2 に変更するか、データ e_2 のクラスターを C_1 に変更することで制約を満たすようにする。

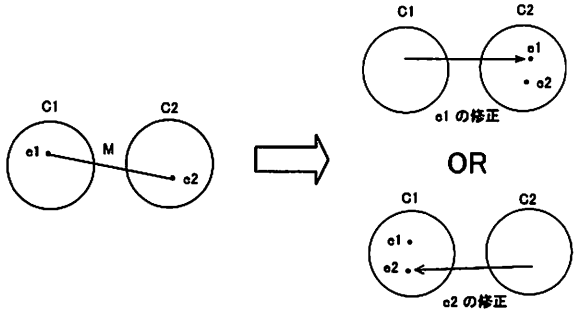

図 1: must-link からの修正

データ e_1 とクラスター C_2 との距離 d_1 と、データ e_2 とクラスター C_1 との距離 d_2 を比較し、距離の短い方のクラスターを修正する。ここでデータとクラスター間の距離は、データとクラスターの重心との距離で求める。第 1 段階のクラスタリング結果を利用することで、クラスターの重心を求めることはできる。

2.2 cannot-link からの修正

データ e_1 と e_2 間の cannot-link の制約が第 1 段階のクラスタリング結果に矛盾する場合、第 1 段階のクラスタリング結果では、データ e_1 とデータ e_2 がともにクラスター C_1 に属しているという状況になっている。データ e_1 のクラスターを C_1 以外の C_x に変更するか、データ e_2 のクラスターを C_1 以外 C_y に変更することで制約を満たすようにする。

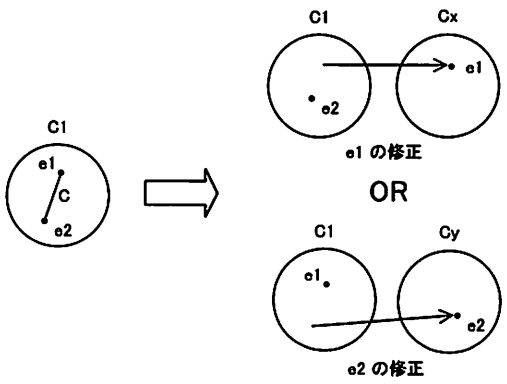

図 2: cannot-link からの修正

まずデータ e_1 と C_1 以外の各クラスター間の距離を測り、最も距離の短い C_x を求める。C_1 以外の各クラスターとの距離の和を sum_1、C_x との距離を d_x とおく。同様に、データ e_2 と C_1 以外の各クラスター間の距離を測り、最も距離の短い C_y を求める。C_1 以外の各ク

[1] http://www.ics.uci.edu/ mlearn/MLRepository.html

ラスターとの距離の和を sum_2、C_y との距離を d_y とおく。そして d_x/sum_1 と d_y/sum_2 の値の小さい方のデータのクラスターを修正する。修正先のクラスターは e_1 を修正するときは C_x、e_2 を修正するときは C_y となる。

2.3 制約の伝搬

第1段階のクラスタリング結果に矛盾する制約を集めて、先の手順で制約を満たすようにデータのクラスターを修正すると、新たに満たすべき制約が生じてくる場合がある。

例えば、データ e_1 と e_2 間には must-link があり、これは第1段階のクラスタリング結果に矛盾しないとする。一方、データ e_2 と e_3 間には cannot-link があり、これは第1段階のクラスタリング結果に矛盾するとする。そして上記の手順で、もしもデータ e_2 のクラスターが変更された場合、データ e_1 と e_2 間の must-link は満たされなくなってしまう。

新たに満たすべき制約が生じた場合、本手法では、先の修正処理を繰り返し実行する。具体的には、第1段階のクラスタリング結果に矛盾する制約を取り出して、それら制約に対して上記の修正処理を行う。その結果新たに作られたクラスタリング結果に対して、再び与えられた制約をチェックし、矛盾する制約があれば、上記の修正処理を行う。これを満たすべき制約がなくなるまで繰り返す。

厳密にはこの繰り返し処理も収束しない場合もあるので、繰り返しの回数に制限を設けている。ただし、本論文で行った実験では繰り返し処理を行う場合はほとんどなかった。

3 実験

3.1 従来手法との比較

本手法の有効性を示すために、代表的従来手法である COP-kmeans と CCL との比較実験を行った。クラスタリングに用いたデータは、ネットニュース記事（395文書、5カテゴリ）と UCI repository の Wine データ（178データ、3カテゴリ）である。どちらの場合も制約の数を 5 から 50 まで 5 刻みで実験を行った。クラスタリングの評価としては正解数を利用した。また与えられる制約による影響を少なくするために、上記の実験を制約を変更して 10 セット行った。以降に示す結果はその 10 セットの実験の平均値である。

また COP-kmeans では k-means を用いるので、最初に選ぶクラスターの代表点によって結果が異なる場合がある。通常はランダムに初期値の代表点を変えて評価関数の値が最良となるものをとるが、ここでは各データに対して、その真のクラスターの重心までの距離を求め、各クラスターに対して、上記距離の平均値に最も近いデータをクラスターの代表点とした。

3.1.1 Wine データ

Wine データは 178 データ、3 カテゴリである。素性は 13 種類で連続値、離散値など様々であるが、ここではそれらの素性の値を正規分布に従うとして、その値を標準化したものに変換した。結果的にデータを 13 次元の実ベクトルとして表現した。本手法の第1段階目で利用したクラスタリングアルゴリズムは ward 法 [5] である。距離はユークリッド距離を用いた。COP-kmeans や CCL で用いた距離もユークリッド距離である。結果を表1に示す。値は正解数（最高 178）を示す。

最大値で比較すると CCL が最も高い正解数を出しているが、最低値との差が大きく、平均的には最も低い。つまり CCL の場合、与えられる制約によって正解数が大きく変化することがわかる。COP-kmeans や本手法では与えられる制約による正解数の変化は小さい。また初期の正解数から制約を与えることで増加する正解数に大きな差はない。このため初期の正解数が最終的な正解数に効いてくる。このタスクの場合、k-means よりも ward 法の方が適切なクラスタリング結果を出すので、結果的に本手法が最もよい値を出している。

本手法は、制約を使っても若干しか正解数が向上していない。これは与えられた制約が第1段階のクラスタリング結果に矛盾しない場合は、その制約は実際には利用されないからである。本手法の場合、第1段階のクラスタリング結果の正解率が高いために、与えられた制約の多くが利用されない。表1に本手法で実際に利用された制約の平均数を示す。制約を 50 個与えたときには、平均して 8.2 個の制約が利用され、それらを用いて 1.3 個正解数が上昇したことがわかる。

3.1.2 ネットニュース記事

ネットニュース記事は 2003年11月25日から12月5日までの10日間でニュースサイト `http://news.goo.ne.jp/` に掲載されたニュース記事である。5カテゴリ (政治、経済、国際、社会、スポーツ) から集めた総数 394 文書のデータである。文書は索引語ベクトルで表現した。ベクトルの i 次元は i 番目の索引語 $term_i$ に対応する。i 次元の値は TF-IDF を利用して重みをつけた。具体的には $c_i \log(n/f_i)$ により重みを計算した。ここで c_i はその文書中の $term_i$ の頻度、n は全体の文書の数、そして f_i は $term_i$ を含む文書の数を表す。これによって得られたベクトルを大きさ 1 に標準化することで、文書をベク

	0	5	10	15	20	25	30	35	40	45	50
COP-kmeans (最大)	153	154	156	156	157	157	158	159	160	159	160
CCL (最大)	137	164	170	170	167	169	172	172	173	173	174
本手法 (最大)	160	161	162	162	162	163	163	163	164	164	164
COP-kmeans (最小)	153	153	153	153	153	153	153	153	153	154	154
CCL (最小)	137	108	108	105	139	118	121	121	117	117	122
本手法 (最小)	160	159	158	158	158	158	157	156	156	157	157
COP-kmeans (平均)	153	153.9	154.3	154.7	155.0	155.0	155.2	155.7	156.4	156.3	156.5
CCL (平均)	137	132.5	147.4	153.3	157.6	151.2	153.2	148.6	138.7	136.5	149.7
本手法 (平均)	160	160.0	159.9	160.1	160.4	160.5	160.6	160.6	160.7	161.1	161.3
COP-kmeans (分散)	0	0.09	0.81	0.61	1.0	1.2	1.76	3.41	4.64	2.61	4.05
CCL (分散)	0	345.45	378.44	333.61	63.64	317.56	300.76	432.64	422.21	361.05	413.61
本手法 (分散)	0	0.2	1.09	1.29	2.04	2.65	3.84	4.44	4.41	3.29	3.21
実際の制約数 (M)	0	0.2	0.8	1.0	1.8	2.0	2.4	3.2	3.2	3.6	3.6
実際の制約数 (C)	0	0.2	0.6	1.2	2.2	2.4	2.6	2.8	4.0	4.4	4.6

表 1: Wine データ実験結果

トルで表現した。次元数は 3364 であった。

本手法の第 1 段階目で利用したクラスタリングアルゴリズムは文書クラスタリングに有効とされるグラフベースの手法である [3]。そこでは文書をノードとし、ノード間のエッジに類似度を与えたグラフを考え、min-max cut のアルゴリズムによりグラフを分割してゆくことでクラスタリングを行う。データ間の類似度はコサインを用いた。本手法の 2 段階目以降に利用する類似度もコサインである。COP-kmeans や CCL で用いた類似度もコサインである[2]。結果を表 2 に示す。値は正解数（最高 394）である。

この実験では CCL 、COP-kmeans ともに、与えられる制約によって正解数が大きく変化している。一方、本手法では与えられる制約による正解数の変化が小さいことがわかる。また初期の正解数は他の手法よりも圧倒的に高い。制約数 50 の平均の正解率でみると COP-kmeans が 54.3%、CCL が 50.3%に対して、本手法は 63.4%と高い値を示しており、本手法の有効性が確認できる。

ネットニュース記事の実験も Wine データの実験と同様、実際に利用した制約数は与えられた制約数よりもかなり小さい。表 2 に本手法で実際に利用された制約の平均数を示す。制約を 50 個与えたときには、平均して 20.0 個の制約が利用され、それらを用いて 3.7 個正解数が向上したことになる。

3.2 修正データの選択の精度

本手法では第 1 段階目のクラスタリング結果と矛盾する制約が生じた場合に、その制約に対する 2 つのデータのどちから一方を選択して、選択した方のクラスターを修正する。

ここではその選択の精度について調べた。利用したデータは先のネットニュース記事である。すべての実験を通じて、選択が行われた種類数と選択の正解数を調べた[3]。注意として must-link の場合は、選択が行われたら、そのデータが属するクラスターは同時に決定されるが、cannot-link の場合は、選択が行われた後に、そのデータが属するクラスターを更に選択する必要がある。ここで行う調査は最初の選択、つまりデータの選択のみである。クラスターの選択は考慮していない。結果を表 3 に示す。表中に「その他」とあるのは、選択対象の 2 つのデータのどちらを選択しても正しい選択にならない場合である。

	選択数	正しい	間違い	その他
must-link	61	36	16	9
cannot-link	39	30	9	0
合計	100	66	25	9

表 3: 修正対象のデータの選択

選択数から「その他」を引いたものの中で、正解数の割合は 72.5% であり、修正対象のデータの選択がほぼ正しく行われていることがわかる。

次に選択されたデータが属するクラスターを決める必要があるが、must-link の場合、正しいデータ選択が行われた場合は、正しいクラスターが選択される。cannot-link の場合、正しいデータ選択が行われた上で、正しいクラスターが選択されたかどうかを調べた。正しいデー

[2]ここでのデータは大きさが 1 に正規化されているので、コサインの類似度を s とすれば、$2^{*}(1\text{-}s)$ により距離に直せる。

[3]ランダムに 2 つのデータを取り出したときに、その 2 つのデータからは、cannot-link の制約が作られる可能性が高いが、ここでは must-link の選択数の方が多くなっている。本手法の場合、第 1 段階で作られるクラスタリング結果に矛盾する制約だけが選択の対象となるので、cannot-link の制約は満たされる可能が高く、結果として must-link の選択数の方が多くなっている。

	0	5	10	15	20	25	30	35	40	45	50
COP-kmeans (最大)	191	196	202	202	217	220	220	220	221	222	224
CCL (最大)	148	207	220	216	237	205	228	226	229	212	222
本手法 (最大)	246	248	249	249	251	252	252	253	254	254	253
COP-kmeans (最小)	191	189	191	190	193	192	192	194	193	193	198
CCL (最小)	148	133	161	144	156	143	147	144	168	152	184
本手法 (最小)	246	245	245	246	246	246	245	245	245	245	246
COP-kmeans (平均)	191	192.2	194.8	195.2	199.4	202.6	206.9	207.7	207.4	207.1	213.8
CCL (平均)	148	172.8	187.8	187.4	185.6	179.1	189.7	192.5	190.6	184.0	198.2
本手法 (平均)	246	246.4	246.9	247.5	248.1	248.6	248.5	249.1	249.1	249.5	249.7
COP-kmeans (分散)	0	4.36	8.86	14.76	83.64	94.84	100.09	89.01	88.04	112.29	75.36
CCL (分散)	0	425.96	328.84	567.24	795.84	356.09	539.00	463.44	485.61	366.96	255.01
本手法 (分散)	0	0.64	1.29	1.45	2.89	3.84	4.25	5.09	7.29	9.84	8.84
実際の制約数 (M)	0	1.0	2.0	3.8	5.0	6.4	6.6	8.6	9.2	11.2	12.2
実際の制約数 (C)	0	0.4	1.0	1.8	2.6	3.2	4.0	4.8	5.2	6.2	7.8

表 2: ネットニュース記事実験結果

タ選択は 30 回あったが、そのうち 22 回 (73.3%) で正しいクラスターが選択されていた。

本手法で利用した修正の手法も有効に機能していることがわかる。

4 考察

4.1 制約数と正解数の向上数

実験で示したように、本手法では与えられた制約数に比べて、正解数の向上が少ない。これは実際に利用した制約数が少ないということの他に、その制約から正しいデータのクラスターを推理するのが難しいという理由がある。

データ間の距離に基づいて、正しいクラスターに修正できるのであれば、それは第 1 段階のクラスタリング結果に矛盾することはない。つまり第 1 段階のクラスタリング結果に矛盾する制約のデータは特異になっている。特異なデータのクラスターを推理するのは困難である。

本手法で提案した修正アルゴリズムでもデータ間の距離しか使っていないので、その精度に大きな期待はできない。表 3 を見ると、実験では修正データが正しいクラスターに修正できる率は 66% であった。この数値の場合、100 個制約があっても 34 個は間違いであり、正解数と相殺されて、結果的に 32 個しか正解数は向上しない。この部分の正解率をどのようにあげていくかが今後の重要なポイントとなる。データ間の距離というナイーブな素性だけでは効果は少なく、第 1 段階のクラスタリング結果に使った素性とは別種の素性の利用が鍵だと考えている。

また、この点において本手法が従来手法よりも劣ることはないことも注記しておく。特異な制約から正しいクラスターを導けない問題は、同じ素性を使っている以上、どのような手法であっても被ってしまう。例えば COP-kmeans であっても特異な制約からは正しいクラスターを導けないので、制約数を増やしていっても、どこかで正解数が頭打ちになってゆく。

この点を示すために、追加の実験を行った。図 3 はネットニュースの記事を使って COP-kmenas と本手法の制約数を 300 まで増やした結果である。横軸は制約の数、縦軸は正解数を示す。これも 10 セットで実験を行い、正解数の平均値を出している。制約数を多くしてゆくと、正解数が頭打ちになってゆく様子がわかる。さらに頭打ちになる付近でも本手法は COP-kmenas よりもよい値を出している。

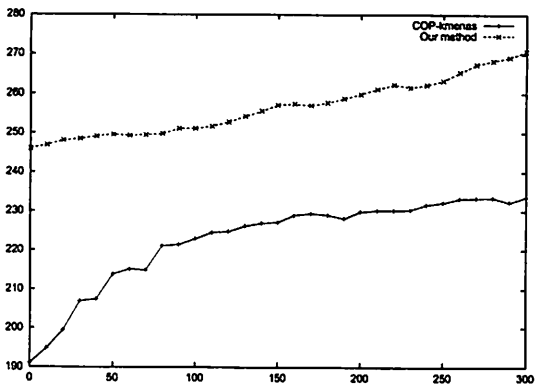

図 3: 制約数を 300 まで増やした実験

4.2 近傍データの修正

本手法では制約 1 つに対して高々1 つのデータのクラスターしか修正を行わない。また第 1 段階で行うクラスタリングの結果に矛盾しない制約であれば、その制約は全く利用されない。

制約の情報を更に効率的に利用するためには、制約を

持つデータの近傍のデータも同時に修正の候補として考慮していく方法が考えられる。概略述べれば、制約によってデータ e のクラスターが h だと判断された場合に、データ e の近傍となるデータ e' もクラスターが h だと判断しようという考え方である。この方法は従来研究では暗に利用されている。COP-kmeans では e のクラスターが h だと判断されると、クラスター h の重心が e に近づくので、その影響で繰り返し処理の中で、データ e' もクラスター h に属する形になってゆく。CCL では制約によってデータ e とデータ x の距離が変更されると、データ e' とデータ x の距離も修正される可能性が高い。

しかしこの方法は本手法ではうまく働かない。実際に実装させてみると、現状よりも大きく改善される場合もあったが、多くの場合、正解数は減少した。データ e とデータ e' の距離が近ければ、第1段階で行うクラスタリングにおいて同じクラスターに属しているはずである。つまり制約のもつデータの近傍を考慮することは、第1段階で行うクラスタリングで既に行っている形になっている。先の考察でも述べたように、第1段階で行うクラスタリングに矛盾する制約のデータは特異であり、一種の外れ値（outlier）と考えられる。この場合、外れ値 e を手がかりにして他のデータのクラスターを推理するのは危険であり、データ e の修正だけにとどめるのが現実的な対処である。

4.3 Active Learning

本手法では第1段階で行うクラスタリングの結果に矛盾しない制約であれば、その制約は全く利用されない。利用されない制約を人間が設定するのは無駄であり、最初にクラスタリングを行い、その結果から設定すべき制約をユーザにフィードバックする方が現実的である。これは Active Learning である。本来、must-link や cannnot-link の制約を大量に与えるのは人間の負荷が高い。大量のデータに制約を与えるくらいなら、少量のデータにラベルを与え、半教師有りの分類問題として扱った方が効率はよいはずである。つまり半教師有りクラスタリングのタスクは Active Learning の形で行うのが正しい方向だと考えている。

その際には、must-link を誘うようなデータの選択が重要であると思われる。半教師有りクラスタリングのタスクでは must-link の制約は cannot-link の制約に比べると情報が多いからである。QBC [7] のような戦略でクラスターの曖昧なデータ x と、第1段階で行うクラスタリングの結果でデータ x の属するクラスターの重心に最も近いデータ y とのペアに対して制約を要求してゆくような方向を考えている。半教師有りクラスタリングのタスクを Active Learning の形で行うことを今後の課題とする。

5 おわりに

本論文では半教師有りクラスタリングの手法を提案した。まず与えられた制約を用いずにクラスタリングを行い、次に制約を使ってそのクラスタリング結果を修正してゆく手法である。特徴としては、クラスタリングのタスクに応じたクラスタリング手法を利用することできる点と、与えられる制約に関して頑健である点である。

実験ではネットニュース記事と UCI repository の Wine データを利用し、従来手法の COP-kmeans と CCL との比較実験を行った。その結果、それらの手法よりも良い結果を得ることができ、本手法の有効性を示せた。また修正の手法も 70% 強で正しく機能していることも確認できた。

修正のアルゴリズムを改良することと、本手法に適した Active Learning の形で、半教師有りクラスタリングのタスクに対処することが今後の課題である。

参考文献

[1] S. Basu, A. Banerjee, and R. J. Mooney. Semi-supervised Clustering by Seeding. In *ICML-2002*, pp. 19–26, 2002.

[2] M. Bilenko, S. Basu, and R. J. Mooney. Integrating Constraints and Metric Learning in Semi-Supervised Clustering. In *ICML-2004*, pp. 81–88, 2004.

[3] C. Ding, X. He, H. Zha, M. Gu, and H. Simon. Spectral Min-max Cut for Graph Partitioning and Data Clustering. In *Lawrence Berkeley National Lab. Tech. report 47848*, 2001.

[4] N. Grira, M. Crucianu, and N. Boujemaa. Unsupervised and Semi-supervised Clustering: a Brief Survey. In *citeseer.ist.psu.edu/727015.html*, 2004.

[5] A. K. Jain, M. N. Murty, and P. J. Flynn. Data Clustering: A Review. *ACM Computing Surveys*, Vol. 31, No. 3, 1999.

[6] D. Klein, S. D. Kamvar, and C. D. Manning. From Instance-level Constraints to Space-level Constraints: Making the Most of Prior Knowledge in Data Clustering. In *ICML-2002*, pp. 307–314, 2002.

[7] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *5th annual workshop on Computational Learning Theory (COLT-92)*, pp. 287–294, 1992.

[8] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained K-means Clustering with Background Knowledge. In *ICML-2001*, pp. 577–584, 2001.

[9] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems 15*, pp. 505–512, 2003.