ロジスティク回帰モデルを用いたラベル付文書クラスタリング

岡野原 大輔 † 辻井 潤一 †‡§
† 東京大学情報理工学系研究科コンピュータ科学専攻
‡School of Computer Science, University of Manchester
§NaCTeM (National Center for Text Mining)
{ hillbig, tsujii }@is.s.u-tokyo.ac.jp

概 要

本稿では，ロジスティック回帰モデルによるクラスタリング手法を提案する．本手法では，データの各クラスタへの所属確率を多クラスロジスティック回帰モデルで定義し，データ中のどの特徴がクラスタ決定に重要かを訓練時に同時に決定できるようになっている．更に学習時に重みに対し L_1 正則化を適用し，更に任意の部分文字列が特徴として利用できる学習を適用することで，少数の特徴（文字列）のみがクラスタの決定に寄与し，それらをラベルとして利用しやすい特徴がある．本手法を文書集合のクラスタリングに適用し，従来手法と比較し，本手法の有効性を述べる．

1 はじめに

クラスタリングとは，与えられたデータ集合を，データ間の類似性や，決められた距離尺度に基づいて，似たもの同士にまとめあげる問題である．

これまでに, クラスタリング手法には様々な手法が提案されており，代表的なものでは K-means，混合分布に基づく方法，スペクトラルクラスタリング [**?**] などが提案されている．

特に近年では，最大マージンクラスタリング (以下 MMC; Maximum Margin Clustering) [**?**] が高精度である点と，教師有学習であるサポートベクトルマシン（以下 SVM）と定式化がほぼ同じであり，従来の SVM で培われた技術が応用可能である点から注目を受け，多く研究されている [**?**, **?**, **?**].

MMC は，もともと教師有学習に利用されていた多クラスサポートベクターマシン [**?**] をクラスタリングに応用したものである．多クラスサポートベクターマシンでは，各特徴の重みを訓練データを用いて学習し，それを利用して新たなデータを分類していたが，MMC では重みだけでなく，各データのラベルも同時に決定するような最適化問題を解くことでクラスタリングを行う．つまり,「データ集合がきれいに分かれるような重みベクトルと各データのラベル」を同時に求める．MMC をそのまま適用した場合の自明な解は，すべてのデータに対し同じラベルを与えた場合になってしまうが，こ各クラスに所属するデータ数の制約 [**?**]，またはそれの近似であるマージンの合計値の制約 [**?**, **?**] を最適化問題に加えることで，意味あるクラスタリングを求めることを可能としている．

本稿では，このクラスタリング問題に対し，次の三つの手法を提案する．

一つ目はロジスティック回帰モデルを用いてクラスタリングを行う手法の提案である．従来の MMC では，マージンの最大化を行うという最適化を行っているため，各データのクラスへの所属確率を与えることができなかった．本稿では，MMC と同様の高精度を達成しつつ，各データの各クラスタへの所属度を与えられるように，ロジスティック回帰モデルを利用したクラスタリングを提案する．

二つ目はクラスターに対しラベル付を行えるよう，利用する特徴のスパース化を L_1 正則化によって実現する．MMC も含め，従来クラスタリング手法の問題点として，各クラスタが何を意味しているのかがわかりにくい点があった．本稿では，最適化の際に重みベクトルに対し L_1 正則化を適用することにより，クラスタリングに利用する特徴の数を制約する．これは従来手法にみられるような，クラスタリングしてから代表する特徴を抽出したり，またはその逆に，代表する特徴を抽出してからクラスタリングを行うようなアドホックな組み合わせではなく，統一的な枠組みでクラスタへのラベリングを行うことが可能である．

三つ目は任意の部分文字列を利用した文書クラスタリングである．これは，任意の部分文字列を利用した文書分類 [**?**, **?**] の本問題への自然な適用である．これにより，文書中に出現する任意の部分文字列を利用しつつも計算量は文書長に比例する程度で抑えられ，クラスタリングを決定するのに効いている部分文字列を網羅的に探すことができる．

これらについて，順に説明を行う．

2 ロジスティック回帰によるクラスタリング

本稿では，入力データ x_i $(i = 1, \ldots, n)$ を k 個のクラスターに分ける問題を考える[1]．入力データ x は特徴関数を利用し，m 次元の特徴ベクトル $(x) \in R^m$ で表されるとする．また，各データのクラスを $y_i \in \{1, \ldots, k\}$ $(i = 1, \ldots, n)$ とする．従来の教師付学習とは違って，y_i は前もって与えられておらず，これを求めるのがタスクとなる．この時，入力 x が y である確率 $p(y|x; \boldsymbol{w})$ を多クラスロジスティック回帰では次のように定義する，

$$p(y|x; \boldsymbol{w}) = \frac{1}{Z(x)} \exp \ \boldsymbol{w}_y^T \ (x) \ . \tag{1}$$

ただし $\boldsymbol{w}_y \in R^m$ は，y に関する重みベクトルであり，$Z(x) = \sum_y \boldsymbol{w}_y^T \ (x)$ は正規化項である．また，$\boldsymbol{w}_y$ $(y = 1, \ldots, k)$ を並べたものを $\boldsymbol{w} \in R^{m \ k}$ と表わすことにする[2]．

この時，入力の生成確率が最大となるようなクラスタ割り当てと重みのペアを求め，その時のクラスタ $\boldsymbol{y}$ をクラスタリング結果とする，

$$(\boldsymbol{y} \ , \boldsymbol{w}) = \arg\max_{\boldsymbol{y}, \boldsymbol{w}} \sum_i \log p(y|x_i; \boldsymbol{w}). \tag{2}$$

この定式化では，重みを自由に決めて良い中で，更にクラスも自由に決定してよい問題となっている．

ここで，重みの最適化に L_1 正則化 $\|\boldsymbol{w}\|_1 = \sum_i |w_i|$ を適用する．L_1 正則化を与えて最適化を行った場合，殆どの重みが 0 となるような結果が得られる．L_1 正則化は，重みパラメータに対して，ラプラス分布を仮定した場合の事後確率最大化をした場合と考えることもできる．よって全体の最適化は次の通りとなる．

$$(\boldsymbol{y} \ , \boldsymbol{w}) = \arg\max_{\boldsymbol{y}, \boldsymbol{w}} \sum_i \log p(y_i|x_i; \boldsymbol{w}) \quad C\|\boldsymbol{w}\|_1 \tag{3}$$

但し，C は $\boldsymbol{w}$ をどの程度スパースにするかのトレードオフパラメータであり，C が大きい場合には，多くの $\boldsymbol{w}$ 中の値が 0 となり，逆には $\boldsymbol{w}$ の多くの値が 0 ではなくなる．

この最適化問題では，$\boldsymbol{y}$ の決定も含まれ，最適化は困難であるため，**[?]** と同じように，変数 $\boldsymbol{y}$ を消去した次の問題を考える．

$$\boldsymbol{w} \ = \max_{\boldsymbol{w}} \sum_{i,y} M(i, y; \boldsymbol{w}) \log p(y|x_i; \boldsymbol{w}) \quad C\|\boldsymbol{w}\|_1 \tag{4}$$

但し，$M(i, y; \boldsymbol{w})$ は次のように定義される関数である[3]，

$$M(i, y; \boldsymbol{w}) = \pi_{y' \neq y} I(\boldsymbol{w}_y^T \ (x_i) > \boldsymbol{w}_{y'}^T \ (x_i)). \tag{5}$$

[1]最適化な k の決定については，今回は扱わない．

[2]各クラスのバイアス項は，特徴ベクトルを拡張することにより対応するとする

[3]もし，値が同じ場合は y の添え字が小さい場合に，1 となる変数

直感的には，$M(i, y; \boldsymbol{w})$ は y は x_i に対し最も高い確率を与えるラベルが y の時 1，それ以外の時は 0 を返すような関数である．**[?]** で用いられているのと同様な証明により，式 **(??)** と式 **(??)** の最適値は一致し，この時の変数 $\boldsymbol{w}$ は等しい（証明略）．

この時，結果として得られた重みベクトルから，それぞれのデータのクラスターは次のように求められる．

$$y_i = \arg\max_y \boldsymbol{w}_y^T \ (x) \tag{6}$$

この最適化の自明な解は全てのデータに対し同じラベルを与えた場合である．また，一つの外れ値からなるクラスと，残り全てのデータからなるクラスといったような望ましくないクラスタリングも防ぎたい．そこで，制約として，一つのクラスが非常に大きく，または小さくなってしまわないような制約を与えることを考える．例えば，**[?, ?]** では，全てのクラス間, p, q について，次のような制約を最適化問題に加える．

$$l \quad \sum_{i=1}^n \boldsymbol{w}_p^T \ (x_i) \quad \sum_{i=1}^n \boldsymbol{w}_q^T \ (x_i) \quad l \tag{7}$$

本稿では，これを更に簡略した次のペナルティを最適化に加えることを提案する．

$$L(\boldsymbol{w}) = \sum_y \left(\sum_i \boldsymbol{w}_y^T \ (x_i) \right)^2 = \ \sum_y \ \boldsymbol{w}_y^T \ (all) \ ^2 \tag{8}$$

但し，$(all) = \sum_i \ (x_i)$ である．この関数は，全てのクラスが大体均等な大きさになった場合に最も小さくなり，一つのクラスが大きい，もしくは小さくなった場合に大きくなるような関数である．

これらをまとめると，次のような最適化を行う

$$\boldsymbol{w} \ = \max_{\boldsymbol{w}} \sum_{i,y} M(i, y; \boldsymbol{w}) \log p(y|x_i; \boldsymbol{w}) \quad C\|\boldsymbol{w}\|_1 \quad C_2 L(\boldsymbol{w}) \tag{9}$$

但し，C_2 は，クラスタの大きさを調整するパラメータである．

式 **(??)** の最適化には，**[?]** で用いられていた手法と同様に，ラベルの割り当てと，重みの最適化を交互に行うことにより求められる．これは，**(??)** 中の $M(i, y; \boldsymbol{w})$ の値を最適化の 1 ステップ中は固定することで実現される．

3 全ての部分文字列を用いた文書表現

次に，どのように文書を特徴ベクトルで表現するかについて述べる．従来の文書分類やクラスタリングでは，文書の表現に，文書中に出現する単語や，N 個の文字や

単語の連なりである文字 N-gram，単語 N-gram のそれぞれがベクトルの各次元に対応している，いわゆる Bag of Words(BOW) 表現を利用していた．BOW 表現は単非常に単純化された表現になっているが，文書のクラス決定には少数の単語の出現が効いている場合が多いため，BOW 表現でも高精度で分類できる場合が多い．

しかし，単語 BOW では次のような問題があることが知られている．一つめは，日本度など分かち書きされていない場合には，形態素解析を行うが，その場合には解析エラーが発生することである．二つ目はデータがそもそも未知ドメインのデータなどで，単語分割が容易にできない場合があるということである．三つ目は最も重要な問題だが，分類/クラスタリングに効くような文字列単位と形態素単位が違う場合があることである．例えば映画タイトルなどは，単語単位に切り分けてしまった場合，情報が失われてしまう．

そこで，全ての部分文字列を利用して BOW 表現することを考える．文書中に出現する全ての部分文字列を直接考えた場合，文書長の 2 乗個の部分文字列がありうるため，そのまま部分文字列をすべて列挙し最適化を行うことは計算量的に困難である．部分文字列の出現回数や長さなどで，利用する部分文字列を制限する場合も，網羅性や最悪時の挙動を制御できないなど問題が残る．

これに対し，適切なデータ構造とアルゴリズムを利用することにより，近似なしで，全文書長に比例した時間で全ての部分文字列を利用した BOW 表現でも線形時間で学習できることが知られている [**?**, **?**]．本稿では，この手法を今回のクラスタリングに応用し，任意の部分文字列を利用した文書クラスタリングを考える．

はじめに，いくつか用語を定義する．入力文書 T と，任意の二つの部分文字列 q_1，q_2 が与えられた時，これらの出現位置が全て同じ場合を $q_1 =_P q_2$ と定義する (正確な定義は [**?**, **?**] を参照)．例えば，$T = abracadabra$ の場合，$ab =_P abr$，$bra =_P abra$ だが，$a \neq_P ab$ である．この時，T 中に出現する全ての部分文字列は $=_P$ の関係により，いくつかの集合に分割される．この各集合に属する部分文字列の中で，一番長い文字列[4]を極大部分文字列と呼ぶ．これらのうち出現回数が二回以上の極大部分文字列の集合を M_T で表すとする．例えば，入力文字列が “abracadabra” の場合，$M_T = \{a, abra\}$ である．極大部分文字列の数は，文書長より少なく，実際には文書長の数分の一であることが知られている．この問題を文書集合にも適用できるようにするために，文書データ (x_i) に対し，それらを入力には出現しない特殊文字 $\$_i$ を間に挟んでつなげた配列 $x_1\$_1x_2\$_2...x_n\$_n$ を作成し，T と置く．

この時，T の二つの部分文字列 q_1, q_2 が，$q_1 =_P q_2$ を満たすならば，q_1 と q_2 の各文書での出現位置は全く同じであり，同様に出現回数も同じである．よって，BOW 表現の場合，$=_P$ の関係を持つ部分文字列の集合は全て同じ統計量を持つ（たとえば文書中の出現回数な文書頻度などは同じである）．よって，極大部分文字列のみを部分文字列の素性として最適化すればよいことになる．

これらの極大な部分文字列は，拡張接尾辞配列を用いることで，効率的に列挙することができる．まず，極大部分文字列は必ず，接尾辞木における内部節点または，葉に対応する部分文字列である．特に，出現回数が 2 回以上の極大部分文字列は必ず，内部節点に対応する．しかし全ての内部節点が極大部分文字列であるとは限らない．極大部分文字列の必要十分条件は，その部分文字列が内部節点に対応し，かつ，その節点に対応する BWT 配列が二種類以上の異なる文字を持つ場合である．接尾辞木中の内部節点の個数は最大でも s　1 個であり [**?**]，これより，極大部分文字列を代表とする集合の数は多くても s　1 個である．

この極大部分文字列に関して，**??**の各重みに関する勾配を求めることは [**?**, **?**] 同様に，補助データ構造を利用することで定数時間で可能である．

さらに，あらかじめ全ての部分文字列を含めて最適化を行わず Grafting を利用することで，実際に利用される部分文字列に比例する時間で最適化を行うことが可能である．

Algorithm 1 アルゴリズム全体の流れ

入力: 訓練データ $\{x_i\}$ $(i = 1, \ldots, n)$，トレードオフパラメータ C
$H = \{\}, \boldsymbol{w} = \mathbf{0}$
loop
　極大部分文字列中から勾配が最大のもの k を探索する (O(n) time)
　(v_k を勾配の値とする)
　if $|\boldsymbol{v}_k| < C$ **then**
　　break
　end if
　$H = H \cup k$
　H に含まれる素性についてのみ，式 **(??)** を最適化
end loop
各 i につき，$y_i = (\boldsymbol{w}\ )^T\ phi(x_i)$ をクラスラベルとして出力

4　実験

提案手法の性能を調べるため，データセットとして **20 newsgroup**(20-news-18828)[5]，**WebKB** を利用しクラスタリングを行った．これらのデータセットを [**?**] らと比較が可能なようにデータを作成した．20 newsgroup は，大

[4] 各集合の極大部分文字列は必ず一意に決まる

[5] http://people.csail.mit.edu/jrennie/20Newsgropus

表 1: 各手法のクラスリング精度 (%) の比較

データ	LLC+ALLSTR	LLC+BOW	K-MEANS	NC	MMC
20-NEWS	71.12	69.15	35.27	41.89	70.63
WK-CL	71.05	65.38	55.71	61.43	71.95
WK-TX	64.50	60.15	45.05	35.38	69.29
WK-WT	74.05	73.12	53.52	32.85	77.96
WK-WC	71.20	71.05	49.53	33.31	73.88

トピックの中から rec を選んだ．rec 中には，トピックとして，autos, motorcycles, basebal, そして hockey が含まれている．また，WebKB は 4 つの大学のウェブページのクローリング結果からなり，それぞれ *student, faculty* など 7 つのトピックに分かれている．

クラスタリングの精度を測るために，[**?**, **?**] で用いられているのと同様の方法を用いた．これは，(1) ラベル付きデータからラベルを取り除く (2) クラスタリングアルゴリズムを動かす．この時，クラス数は元々のラベル種類数と一致させる (3) それぞれのクラスタに対し，その中に含まれるデータにもともと与えられていたラベルを調べ，各クラスタにクラスタ中の最もラベルが大きいものを割り当てる (4) 各クラスタで，正しく分類できているものの個数を調べる．

これは必ずしも，この精度が低い場合に不正解とは言えないが，クラスタリング精度の一つの指標を示しているといえる．

本手法と K-Means，スペクトラルクラスタリングの Normalized Cut(NC) [**?**], そして最大マージンクラスタリング (MMC) [**?**] を比較した．本手法以外の数値はすべて [**?**] からの引用である[6]. 提案手法は，単語をそのまま用いた場合 (LLC+BOW) と，単語と任意の部分文字列 (LCC+ALLSTR) を利用した場合を比較した．結果から提案手法は，大部分で MMC と同程度の精度であった．同じ BOW 表現を用いた場合には，本手法が MMC よりもわずかに低い精度であった．これは MMC が分類精度の最適化を行っているのに対し，LLC は確率の最大化を行っているためであり，確率情報を利用する場合には提案手法の有利性がみられると考えられる．また，任意の部分文字列を利用した場合にはいくつかのデータセットで提案手法が従来の MMC を上回る精度を達成している．

5 まとめ

本稿では，L_1 正則化付ロジスティック回帰モデルによるクラスタリングを提案し，文書クラスタリングに適用した．本手法は高精度を達成しながら，所属確率も与えることができるほか，各クラスタの決定に寄与する特徴をスパースにするようなにすることもでき，各クラスタのラベリングも結果として得られる．

さらに，文書から，任意の部分文字列を文書の特徴として利用しながら文書長に比例する計算量でクラスタリングできることを示した．これにより，文書中に出現する特徴的なフレーズなどをもれなく利用してクラスタリングすることが可能となる

今後の課題としては，クラスタリングの制約としてユーザーが与える指標を今回与えたような指標以外にもっとフレキシブルにできるかということである．また，今回の最適化においては，初期値によってはクラスタリング精度が悪くなる場合もみられた．今後は，初期値に依存しないような，よりロバストな最適化手法を考えていく予定である．

[6] いくつかに関しては再実装を行ったが大体同じか僅かに低い精度であった

参考文献

[1] C. H. Q. Ding, X. He, H. Zha, M. Gu, and H. D. Simon. A min-max cut algorithm for graph partitioning and data clustering. In *ICDM*, pages 107–114, 2001.

[2] L. Xu, J. Neufeld, B. Larson, and D. Shuuramans. Maximum margin clustering. In *NIPS 17*, 2004.

[3] K. Zhang, I. W. Tsang, and J. T. Kowk. Maximum margin clustering made practical. In *ICML 24*, 2007.

[4] B. Zhao, F. Wang, and C. Zhang. Efficient maximum margin clustering via cutting plane algorithm. In *SDM*, pages 751–762, 2008.

[5] B. Zhao, F. Wang, and C. Zhang. Efficient multiclass maximum margin clustering. In *ICML*, 2008.

[6] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *JMLR*, 2:265–292, 2001.

[7] A. J. Smola, S. Vishwanathan, and T. Hofmann. Kernel methods for missing variables. In *AISTATS 10*, 2005.

[8] 岡野原 大輔 and 辻井 潤一. 全ての部分文字列を考慮した文書分類. In **自然言語処理研究会** *(SIGNL-187)*, 2008.

[9] D. Okanohara and J. Tsujii. Text categorization with all substring features. In *SDM*, 2009.

[10] D. Gusfield. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, 1997.

[11] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 2000.