インスタンス選択による文書データからの効率的な分類モデル構築手法

小幡智裕[1]　佐々木稔[2]
[1] 茨城大学大学院理工学研究科情報工学専攻
[2] 茨城大学工学部情報工学科

1. はじめに

機械学習の手法に「教師あり学習」がある。これは、入力に対して何を出力するべきか、学習データと呼ばれる入出力ペアの事例が複数与えられ、その学習データをもとに、新しいデータに対する出力を予測する手法である。「教師あり学習」を用いた識別手法として k-NN(k-nearest neighbor algorithm)や、SVM(support vector machine)[1]が有名である。これらの手法は高い識別能力があるが、学習データが増えると計算が膨大になるという問題点がある。

このような問題を解決するため、学習データの中から分類に必要なデータだけを残すことをインスタンス選択という。インスタンス選択の手法には HMN(Hit Miss Network)[2]や SNG(Sparsifying Neural Gas)[3]といったものがある。これらの手法は、精度を維持しつつ学習データのインスタンス数を減らすことができるため、学習や識別に必要な計算量の減少が期待できる。しかし、HMN は距離計算を 1 回しか行わないために選択後の分類モデルは精度が悪く、SNG は距離計算を 2 回行うためにインスタンス選択で時間がかかるという問題がある。

そこで本研究では、従来手法における問題を解決するために、HMN と SNG を組み合わせたインスタンス選択のシステムを提案し、実験及び評価を行う。このシステムにより、時間効率の向上と、教師あり学習における学習データの質の向上を目指す。本論文では、いくつかの異なる文書データを用意し、文書分類による評価実験を行う。提案手法に基づいて学習データから分類に有効なデータを選択し、SVM を用いて分類モデルを構築する。この分類モデルを使い、テストデータの識別を行う。従来手法である HMN、SNG のインスタンス選択手法と比較して、優れた識別精度を維持しつつ学習データのインスタンスの数を減らすこと、分類モデルの構築にかかる時間を減らすことを目的としている。

本システムの評価尺度には正解率を用いる。正解率は、テストデータの総数に対する、分類モデルにより正しく識別されたテストデータの割合である。この評価尺度により、従来手法と提案した手法を比較してどれだけ識別精度が上がっているかを確認する。また、インスタンス選択にかかる時間、学習にかかる時間、識別にかかる時間についても確認する。

2. SVM (Support Vector Machine)

SVM は線形識別器の教師あり学習アルゴリズムの 1 つであり、現在知られている多くの手法の中で最も識別能力が優れているものの 1 つとされている。識別器とは、ある学習データを 2 つのクラスのいずれかに識別することを目的としている。図 1 に SVM の概念図を示す。

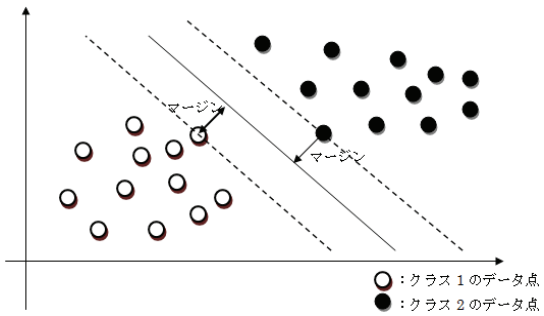

図 1：SVM の概念図

学習データに含まれる各データ点の集合を $\{X_1, X_2, ..., X_n\}$ とする。また、それに対応するクラスを $c = (c_1, c_2, ...c_n)^T, c = \pm 1$ とする。このような学習データが与えられた場合、SVM はクラス 1 に対応する c=-1 を持つ点の集合と、クラス 2 に対応する c=1 を持つ点の集合を上図のように分離する超平面を構成する。このとき、SVM は線形分離不可能な問題にも適用できるように、学習データを高次元空間に射影する。その特徴空間上で、あるクラスのデータ点と最も近い位置にある他クラスのデータ点との間に、これらの点からのユークリッド距離が最も大きくなるような位置に分離超平面を設定する。つまり、クラスの異なるデータと距離（マージン）が最大になる分離超平面を求める。その後、分類したいデータ点

は分離超平面をもとに予測されたクラスを出力する。一般的にデータの特徴量の次元を増やすと識別精度が悪くなるといわれているが、SVM はマージン最大化の概念によりデータの特徴量の次元が大きくなっても識別精度が悪くならないという利点がある。逆に、学習データが増えると計算量が膨大になるということ、また多値分類問題への適応が難しいという欠点がある。

3. 従来手法

3.1. HMN(Hit Miss Network)

HMN は、ベクトル情報とクラス情報を保持した学習データにおいて、各データ点と最近接にあるデータ点との関係を図式化して表現したものである。HMN を用いて各データ点の情報を読み取ることで、学習データ内のどのデータ点がクラス間の境界の点となっているかを算出することができる。HMN の例を図 2 に示す。

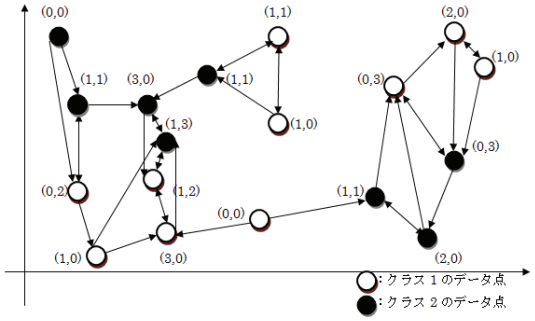

図 2：HMN の例

各データ点において、同一クラスの他の点との最近接の関係(hit)、異なるクラスの点との最近接の関係(miss)となる点に対し、それぞれ有向辺を張る。それぞれの点において、他の点に hit、miss の関係を持つのはそれぞれ他のひとつの点のみであるが、他の点から hit, miss の関係を受けるのは 1 点のみからであるとは限らない。

図 2 中の各データ点における数字は、(hit 次数,miss 次数)を表わしている。ここで、hit 次数とは他の点からの hit 辺の数、miss 次数とは他の点からの miss 辺の数のことである。また、hit 次数と miss 次数を合わせたものを総次数という。

一般的に、総次数が 0 のものは 1-NN においてクラス間の境界から遠い点であるとされる。そして、miss 次数が大きければ大きいほど 1-NN におけるクラス間の境界に近い点であるとされている[2]。

続いて、HMN をもとに HMscore を算出する。HMscore は次のように定義される値である。

$$HMscore(p_a^h, p_a^m) = p_a^h \log \frac{p_a^h}{1/2\, p_a^h + 1/2\, p_a^m} - p_a^m \log \frac{p_a^m}{1/2\, p_a^h + 1/2\, p_a^m}$$

a は学習データ中のインスタンスであり、p は hit 次数、miss 次数を a における HMN の総次数を割ったものである。HMscore の値が大きいインスタンスを学習データから取り除くと、SVM のような、マージン最大化アルゴリズムにおいて、他の多くの点のマージンが減少してしまうとされている[2]。HMN における各データ点の次数と、HMscore には以下の関係がある。

- hit 次数の値が 0 のインスタンスは HMscore の値は 0 もしくは負となる。
- miss 次数の値が 0 かつ hit 次数が 0 より大きいインスタンスは HMscore の値が正となる。
- miss 次数より hit 次数が大きいインスタンスは HMscore の値が正となり、hit 次数より miss 次数が大きいインスタンスは HMscore の値が負となる。

以上の性質から、以下の手順で HMN を用いたインスタンスの選択を行う。

1. 学習データの各データ点について、HMN を算出する
2. 各データ点の HMscore を算出する
3. HMscore の値が負のもの、または 0 であるデータ点は、学習データから取り除いても他のデータ点とのマージン減少の影響が小さいと判断し、学習データから取り除く
4. HMscore の値が正のデータ点の集合を S として出力する

HMN を用いたインスタンス選択の特徴として、この手法は最初に全ての学習データ同士の距離計算を行っているが、その際にクラス情報を使用して効率的な選択を行う。その反面、多くのデータを取り除いてしまい、精度が落ちてしまうという問題がある。

3.2. SNG(Sparsifying Neural Gas)

ニューラルガスは、ベクトル量子化において用いられるネットワーク構造である。ここで、ベクトル量子化とは、多数の入力ベクトルの集合を比較的少ないユニットで近似することをいう。このニューラルガスに対して、インスタンス選択を行う手法として、SNG が提案されている[3]。この SNG を行うための手順を以下に示す。

1. 学習データの中からインスタンスをランダムに 2 つ選ぶ。この 2 つは、ネットワークの初期ニューロンとなる
2. 学習データの全てのインスタンスについて、以下の手順を逐次的に行う
3. インスタンスから最もユークリッド距離の近いニューロンをネットワークから探し、インスタンスがそのニューロンの領域(距離の 2 乗)より遠い位置にあれば、インスタンスを新たにニューロンとしてネットワークに追加する。
   領域より近い位置にあれば、最も近いニューロンをインスタンスに近づけ、インスタンスはネットワークには追加しない

SNG を用いたインスタンス選択は、上記の手順で作成された SNG と学習データとの距離関係を比較することで行われる。以下に手順を示す。

1. SNG を作成する
2. 学習データの全てのインスタンスについて、以下の手順を逐次的に行う
3. インスタンスと最も近いニューロンと 2 番目に近いニューロンを探し、それぞれのニューロンのクラスが異なれば、そのインスタンスを学習データとして残す。それぞれのニューロンのクラスが同じであればそのインスタンスは学習データから取り除く
4. 最終的に残ったインスタンスとニューロンの集合を学習データとする

SNG を用いたインスタンス選択の特徴として、SNG 作成そのものは逐次的に行われるためあまり時間がかからないが、インスタンス選択の際に全ての学習データとニューロンの距離計算を行っているために多くの時間が必要となる。

4. 提案手法

従来のインスタンス選択手法において、インスタンス選択に時間がかかってしまうという問題点を解消するために、本研究では HMN に基づく SNG を用いたインスタンス選択を提案する。これは、従来手法である HMN と SNG の有効な部分を組み合わせてインスタンスを選択するシステムである。SNG を作成し、近似された学習データに対して HMN を作成し、インスタンスの選択を行うというものである。提案手法の手順を以下に示す。

1. SNG を作成する。作成の際に、全てのインスタンスについて、どのニューロンと最も距離が近いかを記録しておき、全てのインスタンスはいずれかのニューロンに属するとする
2. 作成された SNG から HMN を構成する。そこで miss 次数が 0 以上であるニューロンを残す
3. 残ったニューロンに属するインスタンスの集合を新たな学習データとする

SNG を作成し、データ全体を近似し数を減らした後で HMN を作成することで、インスタンスどうしの距離の計算量が減るため、インスタンス選択にかかる時間が減少すると考えられる。

5. 実験

5.1. 実験方法

実験で使用したデータの概要を以下に表 1 として示す。

表 1：使用したデータ

データ名	文書数	語彙数	クラス数
re1	1557	3758	2
news	1067	61188	2
text1	1946	7511	2

re1 は Reuters-21578 から一部を抽出して得られたデータセットである。1657 文書あり、1557 文書を学習データとし、100 文書をテストデータとした。news は、20newsgroups のうち、カテゴリを 2 つ選び、1 つのデータセットとしたものである。text1 は、従来手法である HMN の論文において実験で使われたデータセットである[2]。各手法を用いてインスタンス選択を行い、インスタンス選択にかかる時間と、テストデータの識別を行い比較する。学習の方法としては SVM を用いる。時間効率の評価には、インスタンス選択にかかる時間、学習にかかる時間、識別にかかる時間を用いる。識別精度の評価には、テストデータの識別を行った結果、テストデータの総数に対しどれだけの数のテストデータが正しく識別されていたかの割合を表した正解率を用いる。

実験では、以下に示す 3 種類のインスタンス選択手法を利用して得られた学習データを使用して識別を行う。

I. HMN によりインスタンス選択を行った学習データ
II. SNG によりインスタンス選択を行った学習データ
III. 提案手法により選択を行った学習データ

5.2. 実験結果

表 2 に、各手法についての実験結果として、インスタンス選択、学習、および識別の各実行にかかった時間を示す。単位は秒で、小数点以下第 2 位まで表示している。

表 2：実行時間

インスタンス選択時間			
データ名	I	II	III
re1	450.98	541.56	**222.93**
news	2844.45	2287.5	**799.94**
text1	433.81	312.1	**52.19**
学習時間			
re1	**0.34**	3.91	30.30
news	**23.48**	187.23	440.75
text1	16.94	**7.895**	54.15
識別時間			
re1	**0.99**	1.45	5.09
news	**69.56**	175.24	320.92
text1	4.48	**2.85**	8.73
合計時間			
re1	452.32	546.94	**258.31**
news	2937.49	2649.99	**1561.62**
text1	455.22	322.88	**115.06**

表 3 に、IからIIIの各手法とランダムにインスタンスを選択した場合におけるテストデータの識別精度の結果を示す。単位はパーセントであり、小数点以下第 1 位まで表示している。

表 3：識別精度

データ名	I	II	III	random
re1	98.0	98.0	**99.0**	97.0
news	55.0	55.0	**55.2**	55.0
text1	63.0	93.0	**95.0**	89.0

6. 考察

表 2 より、インスタンス選択にかかる時間を比較すると、従来手法に比べ大きく削減され提案手法が最も短くなった。学習時間と識別時間に関しては、提案手法が最も時間がかかっている。これは、インスタンス選択を行った結果、どれだけデータが削減されたかが関係していると考えられる。HMN では、非常に多くのデータが取り除かれているためこのような結果になったと考える。しかしながら、合計時間をみると提案手法が最も時間が少ない。SNG を使いデータ数をあらかじめ減らした後で HMN を適用することにより、時間のかかる距離計算の量が減ったため、実行時間が短くなったと考える。

識別手法については、いずれのデータセットにおいても提案手法が最も良い結果となった。どれだけのデータが取り除かれたかにもよるが、提案手法は SNG と HMN を組み合わせたことで、よりクラス境界に近いインスタンスを残し、クラス境界から遠いインスタンスを除外することができたと考えられる。

7. まとめ

本研究では、文書データに対して実行可能な HMN に基づく SNG を用いたインスタンス選択を提案し、インスタンス選択に必要な時間の効率化と分類精度の維持を目的とした手法の開発を行った。その結果、従来手法に比べて学習時間と識別時間については時間がかかってしまったが、総合的にみると識別精度を維持しつつ高速化を達成することができた。

最後に今後の課題について述べる。インスタンス選択にかかる時間を大きく削減することはできたが、学習と識別に時間がかかってしまっているため、より多く学習データのインスタンス数を減らすという改善を行うことが課題である。

参考文献

[1] Corinna Cortes and Vladimir Vapnik, “Support-Vector Networks”, *Machine Learning*, Vol.20, No.3, pp.273-297, 1995.

[2] Elena Marchiori, “Class Conditional Nearest Neighbor for Large Mergin Instance Selection”, *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol.32, no. 2, pp. 364-370, Feb.2010.

[3] Mario Zechner and Michael Granitzer, “A Competitive Learning Approach to Instance Selection for Support Vector Machines”, *Proceedings of the 3rd International Conference on Knowledge Science, Engineering and Management,* pp.146-157, 2009.