遺伝的プログラミングによるテキスト分類アルゴリズムの組み合わせ

Combination of Text Classification Algorithms by Genetic Programming

新美　礼彦
Ayahiko Niimi

公立はこだて未来大学 システム情報科学部 情報アーキテクチャ学科
Department of Media Architecture, Future University-Hakodate

Abstract: The analysis is done by expert who analyzes data while combining various algorithms with the trial and error on text mining. In this paper, two or more text classification algorithms are combined by using the genetic programming, and it proposes the system that classifies the text. The tuning of the parameter of the algorithm at the same time constructing the best use of the feature of each algorithm by learning the combination by the genetic programming becomes possible. We discuss combined text mining with genetic programming for mail classification task.

1　はじめに

今までにいくつかのキーワード抽出法が提案されているが、各キーワード抽出法は文献に応じて精度に違いがあり、パラメータチューニングなども大変である。この問題に対して、文献をカテゴリごとに分類し、遺伝的プログラミングを用いてカテゴリごとにキーワード抽出法を自動選択し、キーワードの抽出を行うシステムを提案した。 [1, 2, 3]

本論文では、同じような考えに基づき、テキスト分類に対しても遺伝的プログラミングを用いてテキスト分類手法の組み合わせによるテキスト分類システムの構築について提案する。

本論文では、テキスト分類問題として、スパムメールの分類問題を取り上げる。基本的にメールの内容はテキスト形式で記述されているので、スパムメールとそれ以外のメールに分類するという作業は、テキスト分類作業であるといえる。そのため、メール分類作業にテキスト分類で用いられる様々なアルゴリズムを適用することができる。とくに、スパムメールとそれ以外のメール (正当メール) をそれぞれ正例と負例と捕らえると 2 クラスへの分類問題と考えられる。以前、テキスト分類アルゴリズムとして、テキスト分類で良く用いられているベイズ理論と SVM(Support Vector Machine) を取り上げ、それらによるフィルタを用いて、スパムメールとそれ以外のメールを分類するシステムを構築し、その性能の評価を行った。その結果から、ベイズ理論によるフィルタ (ベイジアンフィルタ) と SVM によるフィルタには対象メールによって、性能に差が出ることがわかった。 [4] そこで、遺伝的プログラミングによりベイジアンフィルタと SVM フィルタの使い分けを自動学習するシステムを検討する。

2　遺伝的プログラミング

遺伝的プログラミング (Genetic Programming:GP) は、生物進化論の考えに基づいた学習法であり、そのアルゴリズムの流れは遺伝的アルゴリズム (Genetic Algorithm:GA) と同様である。 [5] その特徴は染色体表現が GA と異なり、関数ノードと終端ノードを用い構造表現ができるように拡張してあることである。GP では、関数ノードと終端ノードを用いて LISP の S 式形式で個体を表現する。

GP では、個体評価に適応度関数を用いる。適応度関数には、個体の精度、大きさ、計算時間など複数の指標を総合して組み込むことが可能である。

3　メール分類

スパムメールに対する代表的なメールフィルタとして、以下のものがある。

1. 基本的なテキストフィルタ
2. ホワイトリストによるフィルタ

3. ブラックリストによるフィルタ

1 は、今までに受け取ったことがあるメールを元に、簡単な文字列によるルール設定を作成し、そのルールに基づきメールを分類する方法である。たとえば、「Subject ヘッダに” 未承諾広告※” を含んでいたらスパムメールである」などのルールを作成し、メールを分類する。一般的にこのルールを手作業で登録する必要があり、すでに受け取ったことのあるスパムメールからしかルールを作成できない、ルールを作成するのに時間がかかるなどの問題点がある。

2 は、受信を許可するメールアドレスを記述しておき、それ以外のアドレスからのメールを受信しない方法である。受信者が受信許可するメールアドレスを登録する以外に、送信者がアドレスを登録するシステムもある。登録されていないメールアドレスからのメールは、受信者リストへの登録を呼びかけるメールを送信者に送り、応答のあったメールアドレスを自動的に受信者リストに登録する方法である。受信者リストをつくるのにコストがかかるという問題のほかに、正当なメッセージをフィルタリングしてしまいスパムメールと誤検知してしまう可能性が高いという問題がある。

3 は、受信を許可しないサーバまたは、メールアドレス) を記述しておき、それ以外のメールのみ受信する方法である。2 とは逆に、許可しないメールアドレスのリストを作成する方法である。一般的に許可するメールアドレスは個人ごとに異なる可能性が高いが、スパムメールのアドレス、もしくはスパムメールを配信しているサーバは共通していることが多いため、リストを共有することができる。この方法では、正当なメールを見逃してしまう可能性は低くなるが、スパムメールを見逃してしまいフィルタが効率よく動作しなくなる可能性が高い。

これらのフィルタはスパムメール、正当メールの特徴を手作業で抽出する方法である。これに対して、メールの特徴を自動的に抽出する方法が考えられる。メール情報はテキスト形式で記述されているので、メール分類はテキスト分類の一つと捕らえることができる。スパムメールとそれ以外のメール (正当メール) をそれぞれ負例と正例と捕らえると 2 クラスに分ける分類問題と考えられる。そのため、テキストの自動分類アルゴリズムをメール分類に利用することができる。

テキストの自動分類アルゴリズムは、すでにいくつか提案されている。[6, 8, 10] これらの成果をスパムフィルタの構築にも利用することは充分考えられる。

4 ベイジアン・スパムフィルタ

ベイジアン・スパムフィルタは、ベイズ理論を元にしたスパムフィルタである。[11] ベイズ理論では、ある事象の原因となるすべての事象の確率とその原因の元である事象が起こる条件付き確率をもとに、ある事象が起きたときにある原因が起きた確率を求めることができる。メールで使われている文字列 (トークン) の出現確率からスパムメールであるかどうかの確率をベイズ理論により求め、フィルタリングするフィルタである。トークンとして、単語 (またはその語幹)、n 文字の連続する文字列などが用いられる。

あるトークン (w) が含まれているとき、そのメールがスパムメールである確率 (スパム確率:$p(w)$) を、以下の式で定義する。

$$p(w) = \frac{b/n_{bad}}{\alpha g/n_{good} + b/n_{bad}} \tag{1}$$

ここで

$p(w)$: あるトークン (w) が含まれているときのスパムメールである確率 (スパム確率)

n_{bad}: スパムメール数

$b(w)$: スパムメール中で、あるトークン (w) が出現した回数

n_{good}: 正当メールでないメール数

$g(w)$: 正当メール中で、あるトークン (w) が出現した回数

α: 重み

とした。この定義では、正当メール数に重みをつけることによって、スパムメールの誤検知率を減らすようにしている。

また、複数のトークンを同時に含む場合にスパムメールである確率 (複合確率) は、以下のように定義した。

$$P(w_1, w_2, \ldots w_n) = \frac{p(w_1) \times p(w_2) \times \cdots \times p(w_n)}{p(w_1) \times \cdots \times p(w_n) + (1 - p(w_1)) \cdots (1 - p(w_n))} \tag{2}$$

ここで、

$P(w_1, w_2, \ldots w_n)$: あるトークン ($w_1, w_2 \ldots w_n$) が同時に含まれているときのスパムメールである確率 (複合確率)

$p(w_1)$: あるトークン (w_1) が含まれているときのスパム確率

とした。

メールをスパムメールかどうか判定する手順は以下の通りである。手順は事前処理 (フィルタの学習) と判定処理 (フィルタリング) に別れている。

事前処理 (フィルタの学習) スパムメール、正当メールを集める。すべてのメールをトークンに分解し、トークンごとのスパム確率を計算し、データベースに登録する。

判定処理 (フィルタリング) 判定するメールをトークンに分割する。得られたトークンのスパム確率をデータベースに問い合わせる。この中から特徴的なトークンを抽出し、複合確率を求める。複合確率が設定した閾値以上の場合、このメールをスパムメールと判断し、閾値未満なら正当メールと判断する。

特徴的なトークンとして、判定処理に適したトークンを用いる。スパム確率が 0.5 からより離れた確率を持つトークンを使う。スパム確率が 0.5 とは、どちらのメールともいえないトークンである。

5 SVM によるスパムフィルタ

SVM(Support Vector Machine) は、ベクトルで表されるデータ集合を 2 つのクラスに分類するためのアルゴリズムである。[12] SVM によるスパムフィルタでは、SVM を用いてメールをスパムメールと正当メールに分類する。

SVM は、入力としてベクトルで表されたデータ集合を使う。メールを SVM によって分類するには、メールデータをベクトル化する必要がある。テキストのベクトル化は、ベイジアン・スパムフィルタのときと同様にトークンに分割し、出現したトークンに対応するトークンコードとその出現頻度を求めることにより行う。トークンコードを定義するために、事前にメールに現れるトークンをすべて抽出しておく。出現頻度は、出現回数を数えたものや TF-IDF による定義などが考えられる。

SVM を用いたメールをスパムメールかどうか判定する手順は以下の通りである。手順は事前処理 (フィルタの学習) と判定処理 (フィルタリング) に別れている。

事前処理 (フィルタの学習) スパムメール、正当メールを集める。すべてのメールをトークンに分解し、トークンごとの出現頻度を求める。出現したトークンにトークンコードを定義する。トークンコードと出現頻度をもとにベクトル集合を作成する。ベクトル集合と、スパムメールか正当メールかのラベルを使い SVM により学習し、分類器 (フィルタ) を生成する。

判定処理 (フィルタリング) 判定するメールをトークンに分割し、トークンコードと出現頻度のベクトルを作成する。作成したベクトルをフィルタによりスパムメールか正当メールかを判定する。

6 スパム・フィルタの実装

ベイジアン・スパムフィルタと SVM スパムフィルタを実装し、性能を評価した。性能評価には、適合率と再現率を用いた。適合率と再現率は以下のように定義した。

$$rel = s/n \tag{3}$$

$$rep = s/c \tag{4}$$

ここで、

rel: 適合率

rep: 再現率

n: フィルタが正当メールと判定したメールの総数

c: 正当メールの総数

s: フィルタが正当メールと判定したメールで実際に正当メールだったメールの総数

とした。

適合率により、フィルタが正当メールであると判断したメールにおける実際の正当メールの割合を示す。再現率により、実際の正当メールにおけるフィルタが正当メールと判断したメールの割合を示す。

6.1 ベイジアン・スパムフィルタの実装

ベイジアン・スパムフィルタによるメールフィルタを実装し、性能評価を行った。ベイジアン・スパムフィルタとして bsfilter[14] を用いた。英語のトークンは、アルファベット、数字、アポストロフィ、ドルマークを構成要素と見なして、それ以外を区切り文字とした。

日本語のトークンは、bigram を用い、連続する漢字 2 文字、カタカナをトークンとした。正当メール、スパムメールを日本語、英語とも 150 通ずつ用意し、交差検定法にて性能評価を行った。実験結果を Table 1 に示す。

表 1: ベイジアン・スパムフィルタによる分類性能

対象	適合率 (%)	再現率 (%)
日本語のみ	96.71	98.00
英語のみ	73.89	100
日本語、英語	82.40	98.33
+追加処理あり	98.66	98.33

全体的に、高い再現率を得られた。英語のみの適合率が低いのは、良い英語正当メール、スパムメールを用意できなかったためだと考えられる。日本語、英語を同時に対象とするフィルタでは、適合率が 82.40% という結果が得られた。この結果に対し、以下の追加処理を行った結果、適合率を 98.66% に上げることができた。

- メール本文が空のものは無条件でスパムメールと判断する
- メール本文に URL があるがそのリンク先が切れているものを無条件でスパムメールと判断する
- リンクの切れていないのは URL プリフェッチ方式を適用する [13]

この時、スパムメールであるのに正常メールであると分類したメールを調べた。これらのメールは正当メール中に良く似たメールが含まれていることがわかった。似たような出現頻度の正当メールとスパムメールが含まれていたので、うまくフィルタを学習できなかったと考えられる。

6.2 SVM スパムフィルタの実装

SVM スパムフィルタによるメールフィルタを実装し、性能評価を行った。SVM の実装として、SVM^{light} を用いてフィルタを構築した。英語のトークンは、TreeTagger[16] を用いて語幹を抽出して用いた。日本語のトークンは、Chasen[7] を用いて語彙を抽出して用いた。実験には、日本語スパムメール 175 通、日本語正当メール 188 通、英語スパムメール 261 通、英語正当メール 300 通の合計 921 通を用いて、フィルタの学習を行った。学習後のフィルタの性能を Table 2 に示す。

表 2: SVM フィルタによる分類性能

対象	適合率 (%)	再現率 (%)
日本語のみ	98.00	98.00
英語のみ	100	98.04
日本語、英語	97.59	90.00

実験結果から、日本語のみ、英語のみの場合、高い再現率と適合率が得られた。日本のメールや英語のメールのみのメールに対して、高性能のスパムフィルタが構築可能であるといえる。しかし、日本語と英語の両方を含んだメール集合に対しては、再現率が低くなる結果が得られた。日本語のトークンと英語のトークンからなる長いベクトルを入力として取り込むため、冗長な情報によりフィルタを構築することになるからだと考えられる。このため、日本語と英語の双方に対応したシステムを構築する場合、日本語と英語を含んだベクトルを入力に用いるより、入力メールの言語を判断して、日本語なら日本語用のメールフィルタを、英語なら英語用のメールフィルタを用いるようにした方が、効率がいいと考えられる。メールの言語を判断するには、新たにフィルタを作成しなくても、メールヘッダの Content-Type を調べることにより、判断することが可能な場合が多いので、言語判定についての計算コストは無視できる。

7 GP によるテキスト分類手法の組み合わせ

実装したスパムフィルタの性能実験より、ベイジアン・スパムフィルタ、SVM フィルタとも高い適合率と再現率を示すことがわかった。しかし、両フィルタを比較すると、ベイジアン・スパムフィルタの方が再現率が若干高いが、適合率が低いことがわかる。また、両フィルタとも、日本語と英語を同時に分類すると適合率や再現率が下がってしまう。

そこで、両フィルタと使い分けながらフィルタリングすることにより、さらに高性能なフィルタリング行えるのではないかと考えた。どのように 2 つのフィルタを組み合わせるのかを遺伝的プログラミングにより学習させることにより、高性能なフィルタリングを行うメールフィルタリングシステムを提案する。ベイジアン・スパムフィルタと SVM フィルタを使い分ける

だけでなく、メールに応じて、日本語と英語で学習したフィルタを使い分けることができ、複数のフィルタを使うことで、単独のメールフィルタを使ったシステムよりも高い精度が出せるのではないかと考えている。また、ベイジアン・スパムフィルタは複数のクラスへの分類が行えるが、単独の SVM では、2 クラスへの分類しか行えない。複数の SVM フィルタを学習しておき、遺伝的プログラミングにより使い分けを学習することにより、複数クラスへの分類フィルタを構築することもできる。

複数のフィルタの組み合わせを学習するだけなら、決定木による学習なども考えられるが、遺伝的プログラミングを用いることにより、キーワードによるテキストフィルタやホワイトリスト・ブラックリストによるフィルタとの組み合わせも学習できるのではないかと考えている。

提案するシステムでは、関数ノードとして、それぞれのメールフィルタによる結果による分岐を示すノード、終端ノードとして、どのクラスに分類できるか (2 クラスの場合は、スパムメールかどうか) を定義することにより、使い分けの学習を行う。

現在、実験で使用するための学習データを整理している段階である。

8 おわりに

本論文では、テキスト分類に対しても遺伝的プログラミングを用いてテキスト分類手法の組み合わせによるテキスト分類システムの構築について提案した。対象問題として、スパムメールのフィルタリングに関する問題をテキスト分類問題として捕らえ、テキスト分類アルゴリズムを用いることによりフィルタを構築することを試みた。テキスト分類アルゴリズムとして、テキスト分類で良く用いられているベイズ理論と SVM(Support Vector Machine) を取り上げ、それらによるフィルタを用いて、スパムメールとそれ以外のメールを分類するシステムを構築した。単独のフィルタによる性能評価の結果から、フィルタの組み合わせによるシステムを検討した。現在、実験で使用する学習データを整理している段階であり、学習データがそろった段階で、遺伝的プログラミングにより学習により性能を向上させることができるか実験により確認する予定である。さらに、決定木学習などによるフィルタの組み合わせとの性能比較なども検討する予定である。

参考文献

[1] 新美 礼彦、安信 拓馬、田崎 栄一郎: 遺伝的プログラミングを用いたカテゴリごとのキーワード抽出法選択, 第 18 回 ファジィシステムシンポジウム論文集, pp.303–306, 2002

[2] 新美 礼彦: 遺伝的プログラミングを用いたデータマイニングアルゴリズムの組み合わせ手法, 第 19 回 ファジィシステムシンポジウム論文集, pp.815–818, 2003

[3] 新美 礼彦: 遺伝的プログラミングによるデータマイニングアルゴリズムの組み合わせ手法の改良. 第 20 回 ファジィシステムシンポジウム論文集: pp.273–277, 2004

[4] Ayahiko Niimi, Hirofumi Inomata, Masaki Miyamoto, Osamu Konishi: Evaluation of Bayesian Spam Filter and SVM Spam Filter. Joint 2nd International Conference on Soft Computing and Intelligent Systems and 5th International Symposium on Advanced Intelligent Systems (SCIS&ISIS 2004), Yokohama, Kanagawa, Japan: 5pages (in CD-ROM), 2004

[5] J.R. Koza: Genetic Programming, MIT Press, 1992

[6] 市村 由美、長谷川 隆明、渡部 勇、佐藤 光弘: テキストマイニング - 事例紹介, 人工知能学会誌, Vol.16, No.2,pp.192–200, 2001

[7] 松本 裕治、北内 啓、山下 達雄、平野 善隆、松田 寛、浅原 正幸:日本語形態素解析システム『茶筌』version 2.0 使用説明書 第二版, 1999

[8] 那須川 哲哉、河野 浩之、有村 博樹:テキストマイニング基盤技術, 人工知能学会誌, Vol.16, No.2, pp.201–211, 2001

[9] R. Agrawal, R. Srikant: Fast Algorithms for Mining Association Rules, the 20th International Conference on Very Large Databases, Santiago, Chile, 32pages, 1994

[10] 永田 昌明、平 博順: テキスト分類 - 学習理論の「見本市」, 情報処理, Vol.42, No.1, pp.32–37, 2001

[11] Paul Graham: A Plan for Spam, http://www.paulgraham.com/spam.html

[12] 平 博順、春野 雅彦: Support Vector Machine によるテキスト分類における属性選択, 情報処理学会誌, Vol.41, No.4,pp.1113-1123 (2000).

[13] 安東 孝二、河 正浩、安 在根、康 秀勳、北野 利治: SPAM メール対策における新方式の提案, マルチメディア, 分散, 協調とモバイル (DICOMO2003) シンポジウム (2003).

[14] nabeken: bsfilter / bayesian spam filter/ ベイジアン・スパムフィルタ, http://www.h2.dion.ne.jp/ nabeken/bsfilter/

[15] Thorsten Joachims: SVM - Light Support Vector Machine, http://svmlight.joachims.org/

[16] IMS Textcorpora and Lexicon Group: TreeTagger, http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

[問い合わせ先]

新美　礼彦
公立はこだて未来大学 システム情報科学部
情報アーキテクチャ学科
〒 041–8655 北海道函館市亀田中野町 116–2
Phone:0138–34–6222　FAX:0138–34–6301
E-mail:niimi@fun.ac.jp