# ベイズ学習アルゴリズムのスパムフィルタとウイルスフィルタへの適用の最適化

王 卉歓[†] 中谷直司[†] 小池竜一[†]
厚井裕司[†] 朴 美娘[††]

近年スパムによる被害に対抗するため，ベイズ学習アルゴリズムを用いたスパムフィルタが注目されている．また，同様にメールを媒介として多くの被害をもたらすコンピュータウイルスにおいても，既存の手法では対応困難な未知ウイルスに対し，ベイズ学習アルゴリズムを用いたウイルスフィルタの研究が行われている．しかし，ベイズ理論に基づくウイルスフィルタに関する研究は，十分な検討が行われたとはいえない状況である．そこで本論文では，現在スパムフィルタとして広く用いられている Paul Graham 方式，Gary Robinson 方式，naive 方式の 3 種類のベイズ方式を用いたスパムおよびウイルスフィルタとしての性能に関する考察と，メールに対しスパム検出と同時にウイルス検出を行ううえで実装コストの面で有利になる，スパムとウイルス両方のフィルタで高い性能を示す新しいベイズ方式の提案を行う．実験により提案方式は従来方式によるベイジアンフィルタよりも，同等あるいはより低い誤検出率を維持したまま，より高い検出率をスパムとウイルス両方において実現可能であることが示された．

# Optimization of Bayes Learning Algorithm to Spam Filter and Virus Filter

Huihuan Wang,[†] Naoshi Nakaya,[†] Ryuiti Koike,[†] Yuuji Koui[†] and Mi Rang Park[††]

The spam filter that used Bayes learning algorithm was paid attention in recent years as the countermeasure for damages of spam. In computer virus that causes a lot of damage through the medium of mail, the existing technique is difficult to take the countermeasure against the unknown virus. Some researches including us have studied and developed the virus filter that use the Bayes learning algorithm. But it seems that the enough research has been not done until now. In this paper, we compare the performance of spam filters and virus filters that use Paul Graham method, Gary Robinson method, naive method which have previously shown a good performance and widely have been used as spam filter. We also propose the new Bayes method that shows best performance of both spam filter and virus filter. It has advantage that we can detect a number of virus and spam mails at the same time in respect of the mounting cost. As the result, it is possible that the proposed method outperforms three original methods in exterminating both spam and virus with the same or lower false detection rate.

## 1. まえがき

IT 用語辞典[1] の定義によるとスパムとは，公開されている Web サイトなどから手に入れたメールアドレスに向けて，営利目的のメールを無差別に大量配信することである．すなわち，インターネットを利用したダイレクトメールであり，インターネットではメール受信のための通信料は受信者の負担になるため，スパムのように受信者の都合を考慮せず一方的に送られてくるこうしたメールは，きわめて悪質な行為とされている[2]．また，スパム行為は同内容のメールを一度に大量に配信するため，インターネットの公共回線に負荷がかかる点も問題となっている．本格的な電子社会の現代にあたって，スパム被害を未然に防ぐことはきわめて重要である．現在スパムフィルタには基本的にパターンマッチングフィルタとベイジアンフィルタがある．パターンマッチングフィルタは過去の情報から

† 岩手大学工学部
Faculty of Engineering, Iwate University
†† 三菱電機株式会社情報技術総合研究所
Information Technology R & D Center, Mitsubishi Electric Corporation

抽出されたスパムとしての特徴を検索する方法でスパムを検出する．ただしこの手法は，通常のメール（以下，ノンスパム）にもスパムとしての特徴が含まれることが多いため誤検出率が高い．ベイジアンフィルタはベイズ理論による機械学習を行う分類機であり，スパムに対してはパターンマッチング手法よりも有効である．その理由としては，(1) パターンマッチングと異なりすべての特徴を計算する，(2) 学習することで自己更新する，(3) どんな言語に対しても使える，(4) 無意味な単語の羅列などに騙され難い，などがあげられる[3),4)]．ベイジアンスパムフィルタは広く普及しており，なかでも Bogofilter[5)] や POPFile[6),7)] がよく知られている．

一方，インターネットをはじめとするネットワークが急速に発展するにつれて，コンピュータウイルス（以下，ウイルス）による被害は年々深刻なものとなっている[8)~10)]．近年では web サイト自体がウイルスに感染していたため，そのサイトを閲覧したクライアントがウイルスに感染してしまったという事件や，ウイルス感染により公的機関の PC から個人情報が流出するという事件が，一般的なニュース番組で大きくとりあげられるなど，ネットワーク管理者にとどまらず，一般の人々にもウイルスの危険性が知られるようになってきた[11),12)]．しかし，ウイルスによる被害が減少する傾向は一向に見られず，これらに対してさらなる対策を行うことが重要になってきている．ウイルスの検出はウイルス対策ソフトウェアを利用する手法が原則であるが，ウイルス対策ソフトウェアのデータベースに存在しないウイルスは検出することはできない．近年は新種ウイルスの発生速度にウイルス解析およびシグネチャ生成が間に合わなくなりつつある．そこで以前我々のグループは，ウイルス検出にスパムと同様にベイズ理論を用いたベイジアンウイルスフィルタを提案した[13)]．提案手法は定期的に既知のウイルスを学習させておくことで，それらと共通点を持つ未知ウイルスを検出するものであり，ベイズ学習アルゴリズムに Paul Graham 方式を用いた場合に，過去のウイルスと共通点を持つ未知のウイルスを高速に検出可能なことを示した．

現在スパムフィルタとして広く用いられているベイズ方式としては，Paul Graham 方式，Gary Robinson 方式，naive 方式が知られている．このうち，Paul Graham 方式による未知ウイルス検出の有効性については前述のとおりすでに確認されているが，他の方式の有効性については未確認であり，Paul Graham 方式よりも優れた性能を示す可能性は十分に考えられる．また，メールはウイルスの感染経路としては大きな割合を占めており，メールに対しスパム検出と同時にウイルス検出を行う需要は大きいものと思われる．スパムとウイルスの検出を同時に行う場合，その検出アルゴリズムは同一である方が実装コストの面で有利になる．特に，検出の高速化を目的にシステムのハードウェア化などを図るうえでは，2 つのベイジアンフィルタで同一のベイズ方式が利用可能であることのメリットは大きい．そこで本論文では，ベイズ学習アルゴリズムを用いたスパムとウイルス両方のフィルタにおいて有効に機能するベイズ方式に関する考察と，Gary Robinson 方式を改良することで既存の方式よりもより高い性能を実現した新しいベイズ方式の提案を行う．提案方式は従来方式によるベイジアンフィルタよりも，同等あるいはより低い誤検出率を維持したまま，より高い検出率をスパムとウイルス両方において実現可能であることが判明した．

以下，2 章では，スパム検出によく用いられる Paul Graham 方式，Gary Robinson 方式，naive 方式の 3 種類のベイズ学習アルゴリズムについて述べる．3 章では，3 種類の方式を用いたスパム検出について述べる．4 章では，同様に 3 種類の方式を用いたウイルスファイルの検出について述べる．5 章では，Gary Robinson 方式を改良することでスパムとウイルスの両方に対し高い性能を示すベイズ方式の提案と，その有効性について述べる．6 章は，むすびである．

## 2. ベイズ学習アルゴリズム

ベイズ理論とは，過去に起きた事象の確率を利用して未来を予測する手法である．この理論は現実世界から集められたデータだけに基づいて予測を行うため，データ数が多ければ多いほどその精度は向上し，さらに，データの変化に応じて結果も変わる[14)~17)]．ベイズ理論はメッセージを分類するのに非常に有効なため，近年メールフィルタリングの分野でよく用いられている．

最初のスパムフィルタは 1998 年 Pantel らにより提案された[18)]．その後，2002 年 8 月 Graham により A Plan For Spam[19),20)] という論文が発表され，ベイズ学習アルゴリズムを用いた新しいスパムフィルタが提案された．Paul Graham 方式では各特徴のスパム確率 $p(w_i)$ を次のように計算する．

$$p(w_i) = \frac{\frac{b}{n_{bad}}}{\frac{2*g}{n_{good}} + \frac{b}{n_{bad}}} \quad (1)$$

ここで，ある特徴 $w_i$ がスパムとして登場した回数を

$b$，ノンスパムとして登場した回数を $g$ とし，スパムのメール総数を $n_{bad}$，ノンスパムのメール総数を $n_{good}$ とする．なお，ノンスパムへの判定にバイアスをかけるために，ノンスパム中に登場した特徴の登場回数を 2 倍にしている．スパムにしか登場しない特徴は確率 0.99，ノンスパムにしか登場しない特徴は確率 0.01，データベースにない特徴のスパム確率として 0.4 を使用する．また，$2 * g + b$ が 5 より小さい特徴については計算しないことにする．最終的なスパム判定は，スパム確率が 0.5 から離れている特徴の上位 15 個（$p_1, p_2, \cdots p_{15}$ とする）を選んで，次のように複合確率 $S$ を計算する．

$$S = \frac{\prod_{i=1}^{15} p_i}{\prod_{i=1}^{15} p_i + \prod_{i=1}^{15}(1 - p_i)} \tag{2}$$

この複合確率が 0.9 を超えた場合にスパムと判定される．

Gary Robinson 方式[21),22)] は Paul Graham 方式に対して，特徴ごとのスパム出現確率 $p(w_i)$ をバイアスをかけずに求める点や，データベースに存在していない，あるいは登場回数がきわめて少ない特徴も計算で求める点などを改良したものである．特徴ごとのスパム確率を $f(w_i)$ とすると，$f(w_i)$ は以下のように求める．

$$f(w_i) = \frac{(s * x) + (n * p(w_i))}{s + n} \tag{3}$$

ここで，全特徴の $p(w_i)$ の平均値を $x$，特徴の出現回数を $n$，ある定数を $s$ とする．なお $s$ のデフォルト値は 1 である．メールのスパム確率 $S$ は次式で与えられる．

$$S = \frac{P - Q}{P + Q} \tag{4}$$

$$P = 1 - \left\{ \prod_{i=1}^{n} (1 - f(w_i)) \right\}^{1/n} \tag{5}$$

$$Q = 1 - \left\{ \prod_{i=1}^{n} f(w_i) \right\}^{1/n} \tag{6}$$

$S$ は $-1$ から 1 までの数字になるが，$S$ が 1 に近い値の場合にはスパムと判定され，$-1$ に近い値の場合にはノンスパムと判定される．

もう 1 つよく利用されているベイズ学習アルゴリズムは naive Bayes[23)] である．これはメールに含まれる特徴によって，特定のバケツにメールが分類される確率を計算するものである．バケツは特徴とそれぞれの特徴の頻度のリストで構成されている．ここで，$B_1$ から $B_n$ までの $n$ 個のバケツがあり，特定のメール $E$ には $W_1$ から $W_m$ までの $m$ 個の特徴があると仮定する．メール $E$ がバケツ $B_i$ に含まれる確率 $P(B_i|E)$ は次の式で計算する．

$$P(B_i|E) = \left\{ \prod_{j=1}^{m} P(W_j|B_i) \right\} * P(B_i) \tag{7}$$

$$P(W_m|B_i) = \frac{b + 1}{count_i + \sum_{i=1}^{n} v_i} \tag{8}$$

$b$ は $B_i$ に現れた $W_m$ の回数である．$count_i$ は $B_i$ に含まれるすべての特徴の登場回数の和である．$v_i$ は特定の集合に含まれるすべての特徴の数である．メールの分類は，それぞれのバケツについて $P(B_i|E)$ を計算し，最大の値になったバケツに分類するものとする．

上記 3 種類のベイズ方式の基本手法は，何らかの情報を利用して検出されたスパムから特徴を抽出して，スパムデータベースを生成する．また，ノンスパムから特徴を抽出して，ノンスパムデータベースを生成する．そして，フィルタリングしたい新たなメールについては，そのメールに含まれているすべての特徴について，スパムデータベースとノンスパムデータベースに登場する回数を取り出す．この値を用いて，3 種類の方式それぞれ固有の式でメールのスパム総合確率を計算し判断するものである．同様にウイルス検出においても，ウイルス実行ファイルや一般の実行ファイルから特徴を抽出することで，実行ファイルのウイルス総合確率を計算し判断することは可能である[13)]．そこで本論文では，実装コストなどを考慮して，スパムとウイルスの総合確率の計算式を共通化し，なおかつ上記の 3 種類の方式よりも優れた方式を提案する．

## 3. Bayesian Spam Filter

すでに述べたようにベイズ学習アルゴリズムは，従来スパムフィルタとしてよく利用されてきた．そこで，前述の 3 種類のベイズ方式をスパムフィルタとして用い，実際のスパムとノンスパムについて検出率と誤検出率を比較し，分析することからスパムとウイルス両方に適した新しいベイズ方式の議論へと移行する．

ベイズ学習アルゴリズムを用いたスパム検出フィルタは，スパムとノンスパムのデータ系列（単語）には大きな違いがある点を利用している．スパムはその性質上，商品なりサービスの宣伝を行っているため使用される単語に一定の特徴があり，また，どのスパムにも同様の単語が現れるという性質がある．そこで，過去の情報を有効に利用すればスパムを判定することが可能になる．オリジナルのベイジアンフィルタではスパムに頻繁に現れる単語を学習することで，それ

らの単語を多く含んだメールをスパムとして検出している．このようなスパム検出フィルタは，Bayesian Spam Filter と呼ばれている．

### 3.1 メールの特徴抽出とデータベースの生成

本研究では，3 種類の方式について公平かつ正確な比較を行うための理想的な環境を設定した．つまり，200 件のスパムの検出率と対応するのは，200 件のノンスパムについての誤検出率である．また，メール分割時に特徴情報の取りこぼしが起きないように，2 つ以上の単語を組み合わせる場合は N-gram という技術を用いた．N-gram を用いた例として，実際のスパムを 3 単語ごとに分割した場合，次のようになる．

```
Received: from localhost
from localhost (localhost
localhost (localhost [127.0.0.1])
......
```

この手法の長所は単純に単語ごとに分割する場合に比べ，特定の単語の組合せによって生じている特定の意味を抽出可能になることにある．

本研究では，過去に検出された 200 件のスパムと 200 件のノンスパムを，1 単語，2 単語，3 単語，... 22 単語ごとに分割し，各特徴の登場回数をカウントして，各方式の計算方法でそれぞれのデータベースを生成した．そして，生成した 1 単語から 22 単語までの各方式のデータベースと，対応した組合せ単語数で処理した実際のスパムとノンスパムに対し実験を行った．

### 3.2 実験結果と考察

本研究は 3 種類のベイズ方式を検出率，誤検出率の 2 つの指標から比較し，スパムフィルタとして，より最適な方式について考察する．検出率は学習に用いていない新たな 200 件のスパムについて，正しくスパムと判断された割合であり，誤検出率は学習に用いていない新たな 200 件のノンスパムについて，誤ってスパムと判断された割合である．

3 種類の方式の検出率を**図 1** で示す．横軸は単語を何単語ごとに分割したかを表す単語数（1–22）であり，縦軸は検出率（%）である．誤検出率を**図 2** に示す．横軸は単語を何単語ごとに分割したかを表す単語数（1–22）であり，縦軸は誤検出率（%）である．

naive 方式の場合には，検出率が最も高いのは 1 単語ごとに分割する場合で 96%になる．しかしこの場合，誤検出率も各分割数の中で一番高く 1.5%である．これは 1 単語ごとに分割する場合では，スパムをスパムらしい特徴を持つ 1 単語単位で検出できる一方で，単語の前後の繋がりを完全に無視していることに原因がある．つまり，単語の組合せによって生じている特定の意味を無視しているため，ノンスパムをスパムとして誤検出する可能性が高いということである．したがって 2 単語以上で分割する場合，誤検出率は 0.5%以下と良好な結果が得られている．特に 2 単語で分割する場合は検出率も 94%と 1 単語ごとの場合とさほど変わらず，有効に機能しているものと考える．なお，この naive 方式と後述する Gary Robinson 方式では誤検出率 0.5%，すなわち 1 つのノンスパムをスパムと誤検出している例が多く見受けられるが，このとき誤検出されているノンスパムはすべて同じものである．このノンスパム中には，スパムに多く見られる単語が連続して現れている部分が存在しており，今回の誤検出はこの部分の影響によるものと思われる．

Paul Graham 方式の場合には，分割した単語数すべてにおいて誤検出率は 0%である．3 種類のベイズ方式の中で誤検出率が一番低いのはこの方式である．この方式は特徴のスパム確率を計算するときに，ノンスパムがスパムと判断されることを防止することを目的に，特徴がノンスパムに登場する回数にバイアスを

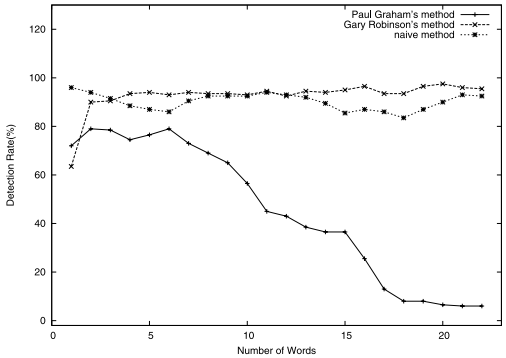


**図 1** 3 種類の方式のスパムの検出率
Fig. 1 Detection rate of spam of three kinds of methods.

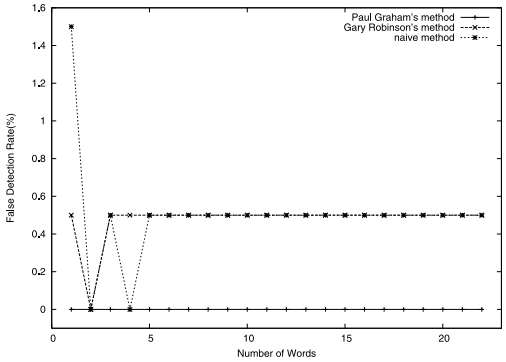


**図 2** 3 種類の方式のノンスパムの誤検出率
Fig. 2 False positive rate of nonspam of three kinds of methods.

かけている．そのため，全体的に特徴のスパム確率が低くなるため誤検出率は低くなるが，その一方で検出率も低いという結果になる．Paul Graham 方式では，2 単語ごとに分割する場合に検出率が一番高く 79%である．仮にバイアスをかけずに検出率を算出すると 86%になるが，誤検出率も同時に高くなって 0.5%になる．図 1 からみると，Paul Graham 方式の検出率は分割する単語数の増加に従って低くなる．原因の 1 つは $2 * g + b < 5$ の特徴については確率を計算しないという制限にあり，分割する単語数が増えれば増えるほど特徴の登場回数が少なくなるため，確率が計算されない特徴の増加により検出率が低くなっている．

Gary Robinson 方式の場合には，Paul Graham 方式とは逆に，分割する単語数が増えれば増えるほど検出率は高くなる傾向にあり，20 単語ごとに分割する場合の検出率は 97%に達した．一方，誤検出率はすべてにおいて 0.5%以下である．Gary Robinson 方式は他の 2 つの方式に比べ，検出率が高く，誤検出率は低いという注目すべき結果が得られた．さらに，Gary Robinson 方式には次のような特徴がある．

(1) 登場回数の少ない特徴でも確率を計算する．すなわち，すべての特徴を重視することで公平な評価が得られるはずである．
(2) データベースに存在しない特徴も平均値で計算する．3 種類のベイズ方式の中でこの方式だけ，スパムデータベースとノンスパムデータベース両方に存在しない特徴にも，学習データに応じた確率を割り当てている．

本研究の最終的な目標は，既知スパムとウイルスの特徴を学習することで，それらと共通点を持った未知スパムやウイルスを検出可能にすることである．しかし，未知スパムとウイルスは学習データには存在しない特徴をいくつも保有するはずであり，このような特徴を未知スパムとウイルスの検出に関係付けることは重要だと思われる．

## 4. Bayesian Virus Filter

ウイルス検出に本来スパム向けの検出アルゴリズムであるベイジアンフィルタを用いて検出が可能になる理由は，スパムもウイルスも「ある目的を達成するための単語，あるいは命令の集合体」であり，類似性があるためである．ウイルスも自己拡散し，多数の PC に感染していくという目的を持っている．それゆえ，ウイルスはその目的を達成するために様々な動作を行う[10]．ただし，同一目的のために行動している以上，ウイルス間で似たような行動をとることが多い，この傾向は亜種ウイルスの中ではさらに顕著であり，なかには送信するメールの内容，あるいは添付するウイルスのファイル名だけが異なるようなものも存在する．そこで，一定の手続きでウイルスから自身の動作に関する命令を取り出せば，一般の実行ファイルには低頻度で出現し，ウイルスにだけ高頻度で出現する命令が存在すると考えられる．このようにウイルス中にだけ含まれる命令が存在するため，ベイジアンフィルタがスパムに対して有効であるのと同様に，ウイルス検出においても既知ウイルスを学習することで，その後に発生する未知ウイルスを検出できる可能性がある．

従来の研究[13]では，ベイズ学習アルゴリズムに Paul Graham 方式を用いたウイルスフィルタ，Bayesian Virus Filter（BVFilter）が提案されている．しかしすでに述べたように，ベイズ学習アルゴリズムにはいくつかの種類が存在し，本論文の目的であるスパムとウイルスに共通した方式の提案には，Paul Graham 方式以外の方式を用いた BVFilter に関する検討も必要となってくる．そこで本論文では，BVFilter のベイズ学習アルゴリズムに前述の 3 種類のベイズ方式を用いる．基本的な手法はスパムの場合と同様となり，まず，何らかの情報を利用して検出されたウイルスから特徴を抽出して，ウイルスデータベースを生成する．なお，このウイルスデータベースは，一般的なウイルス対策ソフトウェアにおけるシグネチャに相当するものといえるので，以下では BVSignature と呼ぶこととする．また，ノンウイルスから特徴を抽出して，ノンウイルスデータベースを生成する．そして，フィルタリングしたい新たな実行ファイルについては，そのファイルに含まれているすべての特徴について，ウイルスデータベースとノンウイルスデータベースに登場する回数を取り出し，3 種類の方式それぞれ固有の式でファイルのウイルス総合確率を計算し判断する．

### 4.1 実行ファイルの特徴抽出とシグネチャの生成

ウイルスファイルの多くは実行可能圧縮という特殊な圧縮がなされており，そのままの状態では特徴を抽出することは難しい．そこで，本研究ではウイルスファイルは事前に解凍したものを用いた．そのうえで，ウイルスファイルとノンウイルスファイル（通常の Windows 実行ファイル）の特徴を，linux 環境で GNU strings コマンド[24]を利用して抽出した．

#### 4.1.1 GNU strings による特徴抽出方法

GNU strings コマンドは，与えられたファイルから表示可能なキャラクタの列を表示する．PE 形式の実行ファイル中には使用する DLL と API が *strings* として記述されている．実行ファイルが正常に動作する

ためには，OS に対して様々な処理を依頼する必要がありこれらの情報は必須である．また，実行ファイル中には自らの動作時に参照する文字が *strings* として含まれる．たとえば，メールを送信するようなウイルスなら，メール本文や使用する単語のリストを内部に持っており，さらには改竄を行うファイルのパスや，追加を行うレジストリのキーなども *strings* として内部に持っている．strings コマンドは，デフォルトでは 4 文字以上の長さのものを可能な限り表示するが，本研究では *strings* の最短の長さを 4–20 に設定して，*strings* の最短長と検出率および誤検出率の関係を考察した．

#### 4.1.2 BVSignature の生成と検出

BVFilter では，スパムフィルタにおいてメールを n 単語ごとに分割したものを特徴としてデータベースを生成したように，ウイルスファイルから抽出した *strings* を特徴として生成したデータベースを BVSignature とする．

実験では，32 種類のウイルスファイルそれぞれ 200 件ずつ合わせて 6,400 件と，ノンウイルスファイル 400 件を用意した．32 種類のウイルスとは，2004 年 8 月から 2005 年 2 月にかけて発見されたウイルスであり，その登場順番は Klez.H，Swen.A，Mydoom.A，Mydoom.F，Netsky.C，Netsky.D，Beagle.J，Beagle.K，Netsky.K，Netsky.N，Netsky.P，Netsky.Q，Lovgate.R，Netsky.S，Netsky.T，Netsky.X，Beagle.W，Beagle.X，Lovgate.W，Explet.A，Lovgate.X，Beagle.Y，Lovgate.Z，Lovgate.AC，Lovgate.AD，Beagle.AB，Mydoom.L，Mydoom.MM，Mota.B，Mydoom.R，Buchon.A，Sober.I である．登場順に各種類ごとに BVSignature を生成し，32 種類のウィルスファイルの検出を試みた．つまり，1 種類のウイルスファイルの 200 件を使用して BVSignature を生成し，6,400 件すべてのウイルスファイルと 200 件のノンウイルスファイルについて検出を実行した．

### 4.2 ウイルスファイルの検出率の評価指標

本研究はウイルスファイルについて，以前現れたウイルスファイルからウイルスらしい特徴点を抽出し，ベイズ学習アルゴリズムを利用して，新たなウイルスファイルの中にどれくらいウイルスらしい特徴を含んでいるか，確率を計算し判断する手法である．しかし，その性能を評価するにあたり単純にウイルスファイルの検出率を求めるだけでは，ベイズ学習アルゴリズムの完全に現実の世界から集められたデータに基づいて未知事象の生起確率を求めるという特長を十分に評価することはできない．すなわち本研究の場合，以前発生したウイルスの特徴を学習し，その学習結果に基づき未知ウイルスをどの程度検出できるかを評価するには，単純な検出率は適当ではない．このことは，実際に未知ウイルスを検出する場合において，未来に発生したウイルスを学習することで過去のウイルスが検出できたとしても，まったく意味をなさないことからも明らかである．そこで，この問題を解決するため Chain という概念を提案する．

Chain とは連続して検出が可能なウイルスの集合を示している．つまり，あるウイルス群をそれらが発生した順に BVFilter に入力した場合の Chain の数によって，未知ウイルスを検出できなかった回数が何回存在したのかを表現することができ，この Chain 数が少ないということは検出率が高いことを意味する．以下では，Chain の概念を**表 1** を用いて具体的に説明していく．

まず，Lovgate.R が未知ウイルスとして拡散すると想定する．その後，ウイルス対策ソフトウェアのシグネチャの更新により Lovgate.R が検出可能となるので Lovgate.R から BVSignature を生成する．すると，その後に Netsky.S と Netsky.T が未知ウイルスとして発生するものの 100%検出することができる．次に Netsky.X が発生するが，先ほど検出できた Netsky.S か Netsky.T から生成した BVSignature で 100%検出することができる．次に Netsky.Y が発生するが，これは以前に発見された BVSignature で検出が可能である．ただし，次に発生する Beagle.W は，これまでに生成した，どの BVSignature によっても検出することができない．よって，ここで Chain が 1 度切れたと解釈する．ウイルス対策ソフトウェアのシグネチャの更新により，Beagle.W がウイルスであることが確認される．その時点で Beagle.W から BVSignature を生成する．すると，次に発生する Beagle.X が検出可能となる．次に，Sober.G と Lovgate.W が発生するが，これは以前に生成した Netsky.S や Lovgate.R の BVSignature で検出が可能である．次に，Explet.A と Erkez.B が発生するが，これまでに生成したどの BVSignature によっても 100%の検出はできないので，2 度 Chain が切れたと解釈する．最後に Lovgate.X が発生するが，これも以前に生成しておいた BVSignature で検出が可能である．上記のような連続して検出が可能なウイルスどうしの集合は，式 (9) のように表せ，これによれば Lovgate.R，Beagle.W，Explet.A，Erkez.B の 4 カ所で Chain が切れる，すなわち Chain 数は 4 となり，言い換えるならば未知ウイルスを検出

**表 1** 各 BVSignature における検出率
Table 1 Detection rate of each BVSignature.

| BVSignature (ordered by virus appearance) | Input | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Lg.R | Ns.S | Ns.T | Ns.X | Ns.Y | Bg.W | Bg.X | So.G | Lg.W | Ex.A | Er.B | Lg.X | Nonvirus |
| Lovgate.R | 100.0 | 100.0 | 100.0 | 0.0 | 100.0 | 0.0 | 0.0 | 0.0 | 100.0 | 0.0 | 32.5 | 100.0 | 15.0 |
| Netsky.S | 23.5 | 100.0 | 100.0 | 100.0 | 100.0 | 0.0 | 0.0 | 100.0 | 0.0 | 0.0 | 32.5 | 100.0 | 3.0 |
| Netsky.T | 23.5 | 100.0 | 100.0 | 100.0 | 100.0 | 0.5 | 0.6 | 100.0 | 0.0 | 0.0 | 0.0 | 100.0 | 2.5 |
| Netsky.X | 0.0 | 100.0 | 100.0 | 100.0 | 100.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 32.5 | 0.0 | 2.5 |
| Netsky.Y | 23.5 | 100.0 | 100.0 | 100.0 | 100.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 32.5 | 100.0 | 0.0 |
| Beagle.W | 15.0 | 0.0 | 0.0 | 0.0 | 0.0 | 100.0 | 100.0 | 0.0 | 0.0 | 0.0 | 0.0 | 100.0 | 2.5 |
| Beagle.X | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 100.0 | 100.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 |
| Sober.G | 15.0 | 100.0 | 100.0 | 0.0 | 0.0 | 0.0 | 0.0 | 100.0 | 0.0 | 0.0 | 0.0 | 100.0 | 3.5 |
| Lovgate.W | 100.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 100.0 | 0.0 | 32.5 | 100.0 | 4.5 |
| Explet.A | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 100.0 | 0.0 | 0.0 | 2.5 |
| Erkez.B | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 100.0 | 0.0 | 0.0 |
| Lovgate.X | 100.0 | 100.0 | 100.0 | 0.0 | 100.0 | 100.0 | 3.1 | 100.0 | 100.0 | 0.0 | 32.5 | 100.0 | 13.0 |

Lg: Lovgate, Ns: Netsky, Bg: Beagle, So: Sober, Ex: Explet, Er: Erkez

**表 2** 3 種類の方式の Chain と最大誤検出率の比較
Table 2 Comparison of Chain and maximum false positive rate of three kinds of methods.

| strings の最短長 | Paul Graham 方式 | | Gary Robinson 方式 | | naive 方式 | |
|---|---|---|---|---|---|---|
| | Chain | 最大誤検出率 | Chain | 最大誤検出率 | Chain | 最大誤検出率 |
| 3 | 8 | 15.0% | 6 | 0.5% | 9 | 0.5% |
| 4 | 8 | 3.5% | 7 | 2.0% | 5 | 4.0% |
| 5 | 8 | 3.5% | 9 | 2.0% | 5 | 4.0% |
| 6 | 9 | 2.5% | 6 | 10.5% | 5 | 9.0% |
| 7 | 9 | 2.0% | 6 | 11.5% | 5 | 10.5% |
| 8 | 9 | 2.0% | 4 | 18.5% | 5 | 11.5% |
| 9 | 9 | 2.0% | 6 | 13.5% | 5 | 13.0% |
| 10 | 12 | 2.0% | 6 | 14.5% | 4 | 13.5% |
| 11 | 12 | 1.5% | 6 | 16.5% | 4 | 14.0% |
| 12 | 12 | 1.5% | 6 | 15.5% | 4 | 14.5% |
| 13 | 11 | 0.5% | 5 | 13.0% | 5 | 14.5% |
| 14 | 11 | 0.5% | 5 | 14.5% | 5 | 16.5% |
| 15 | 11 | 0.5% | 5 | 15.0% | 5 | 19.5% |
| 16 | 11 | 0.5% | 5 | 16.5% | 5 | 20.5% |
| 17 | 11 | 0.5% | 5 | 16.0% | 5 | 21.5% |
| 18 | 11 | 0.5% | 6 | 19.0% | 5 | 24.5% |
| 19 | 10 | 0.5% | 5 | 21.5% | 4 | 27.5% |
| 20 | 10 | 0.5% | 5 | 23.5% | 4 | 29.5% |

できない回数が 4 回だけ存在することを示している．つまり，表における順番でウイルスが発生した場合には，ウイルス対策ソフトウェアでの防御では，12 回の未知ウイルス発生を経験するところを，BVFilter では 4 回に軽減できたことになり約 67%の減少率である．この減少率は，亜種ウイルスが連続して発生した場合には大きくなる傾向がある．なお，Lovgate.X は $Chain_1$ に含まれる複数のウイルス，および $Chain_2$ に含まれる Beagle.W から生成される BVSignature で検出されるため，$Chain_1$，$Chain_2$ の双方に含まれる．

$$\begin{aligned} Chain_1 &= \{Lovgate.R, Netsky.S, Netsky.T, \\ &\quad Netsky.X, Netsky.Y, Sober.G, \\ &\quad Lovgate.W, Lovgate.X\} \\ Chain_2 &= \{Beagle.W, Beagle.X, Lovgate.X\} \\ Chain_3 &= \{Explet.A\} \\ Chain_4 &= \{Erkez.B\} \end{aligned} \tag{9}$$

### 4.3 実験結果と考察

本実験では，3 種類のベイズ方式において最も連続で検出できる，すなわち，Chain 数が少なく，そして，誤検出率も低いベイズ方式を判断する．

3 種類のベイズ方式ごとの *strings* の最短長の変化に対する，Chain 数と誤検出率の変化を**表 2** と**図 3**,

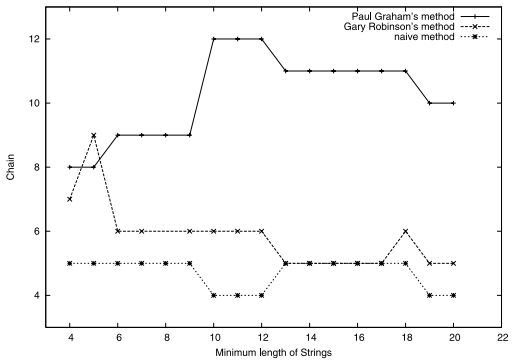


**図 3** 3 種類の方式のウイルスファイルの検出率
Fig. 3 Detection rate of virus of three kinds of methods.

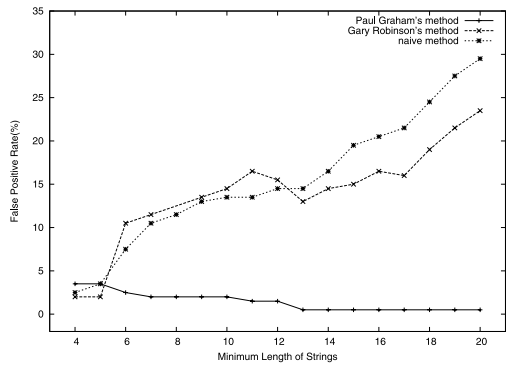


**図 4** 3 種類の方式のウイルスファイルの誤検出率
Fig. 4 False positive rate of virus of three kinds of methods.

**図 4** に示す．表 2 は 3 種類の方式の Chain 数と最大誤検出率を示したものである．図 3 の横軸は *strings* の最短長，縦軸は Chain 数である．図 4 の横軸は *strings* の最短長で，縦軸は最大誤検出率（%）である．誤検出率は 200 件のノンウイルスファイル中にウイルスファイルと判断されたファイルの割合であり，最大誤検出率は 32 種類のウイルスに対応する BVSignature 使用し，200 件ノンウイルスファイルについてそれぞれ検出した際の最も高い誤検出率の値である．

Gary Robinson 方式と naive 方式では，*strings* の最短長が長くなるに従って誤検出率が高くなる．その原因は，*strings* の長さが長い特徴のウイルス確率は高いことが多く，*strings* の長さが短い特徴のウイルス確率は低い，つまり確率が 0.01 の特徴が多いことによる．すなわち，*strings* の最短長を長く設定するとき，この最短長より短い特徴が，つまり，確率 0.01 の特徴がたくさん捨てられる．ノンウイルスファイルには確率 0.99 の特徴はめったに存在しないが，確率が 0.5 よりも大きい特徴が大量に存在する．結局，Gary Robinson 方式と naive 方式の計算式で計算すると，総合ウイルス確率は高くなる．

Paul Graham 方式は，逆に *strings* の最短長が長くなるに従って，誤検出率はだんだん低くなる．この理由は次のように考えられる．まず本研究では，基準値である 0.5 から離れている特徴の上位 15 個を選択するという本来の手法を用いると，ウイルスファイルがほとんど検出できないため基準値を 0.499 と設定している．これにより，たとえば *strings* の最短長が 4（以下では *strings*=4 と記述する）のときは確率 0.01 の特徴が大量に存在するが，0.499 から離れているもの 15 個を選ぶため，確率 0.99 の特徴は確率 0.01 の特徴より優先的に選択される．したがって，もしたくさんの特徴の中に確率 0.99 の特徴が 8 個だけ存在する場合でも，これら 8 個の 0.99 の特徴が優先的に選択されウイルスファイルと判断される可能性が高い．そして，逆に *strings* の最短長が長いときは，ノンウイルスファイルの中に確率の 0.99 の特徴はめったに含まれないため，0.01 は他の確率（たとえば 0.98，0.97，0.6 など）より優先的に選択される．つまり，たくさんの特徴の中に確率 0.01 の特徴が 8 個だけ存在する場合でも，ノンウイルスファイルと判断される可能性が高い．

図 3 から明らかなように，Gary Robinson 方式の検出率は Paul Graham 方式よりも高い．つまり，あるウイルスファイルは Gary Robinson 方式で検出できるが，Paul Graham 方式では検出できない．この原因を解明するために，Swen.A の 200 件のウイルスファイルを使用して BVSignature を生成し，Beagle.J の 1 つのウイルスファイルの検出を行った．この Beagle.J は Gary Robinson 方式では検出できるが，Paul Graham 方式では検出できないものである．このウイルスファイルの検出状況を詳細にみると，Paul Graham 方式で選ばれた 15 個の特徴の確率は，0.01 は 9 個，0.99 は 6 個となっており，そのためノンウイルスファイルと判断されていた．しかし，Gary Robinson 方式では平均値 $x$ が 0.409 となり，ウイルスファイルと正確に判断されていた．この原因は Paul Graham 方式は 0.01 や 0.99 のような極端な値を重視するため，0.01 の個数が 0.99 よりも少しでも多いとノンウイルスファイルと判断される可能性が高いことにある．一方，Gary Robinson 方式は 0.98，0.55 のような極端ではない数値も算出に用いるため，Paul Graham 方式のように確率 0.01 の特徴が多く存在するだけでノンウイルスとは判断しない．

図 4 に示すように，Paul Graham 方式はスパムの

場合と同様に，3 種類の方式の中で最大誤検出率は一番低いが検出率も一番低く，つまり Chain 数も一番多い．また Gary Robinson 方式と naive 方式において，*strings* の最短長の増加に従って誤検出率が著しく高くなる原因は，設定された最短長よりも短い情報が捨てられることで，情報が大量に失われるためである．なお，スパムの場合は単語の組合せを変えているだけで捨てられる情報は存在せず，組合せ単語の数が増えても誤検出率はほとんど同じになる．

## 5. 新しいベイズ方式の提案

表 2 を見ると，Chain 数が少ないと誤検出率は高くなる傾向がある．本実験では Chain 数が一番少ないのは naive 方式の *strings*=10，11，12 のときで，Chain 数は 4（以下では Chain=4 と記述する），最大誤検出率は約 13.5%になる．誤検出率が一番低いのは Paul Graham 方式の *strings*=11，12，13，14，15，16，17，18，19，20 のときで，最大誤検出率は 0.5%，Chain 数は Chain=11 と Chain=10 になる．検出率の高さは重要な要素ではあるが，同時に誤検出率の低さも重視する必要がある．そこで本研究では，Chain 数をなるべく少なくし，同時に最大誤検出率は 3%を超えないという範囲をフィルタとしての許容範囲とする．しかし，ここまでの実験結果においては 3 種類の方式ともこの範囲を満たしていない．

この問題を解決し，さらにスパムとウイルス両方に対応する方式を検討をするため，スパムにおいて naive 方式や Paul Graham 方式よりも検出率が高く，そして，平均値を使うことでデータベースに存在しない特徴も総合確率に反映させる Gary Robinson 方式に注目し，まず，この方式に存在するいくつかのパラメータと特徴抽出のパラメータを変化させることによって，より高い検出率とより低い誤検出率の実現を目指す．

### 5.1 最適なパラメータ設定

#### 5.1.1 *stirngs* の長さ

*strings* の最短長を 4 から 20 までに設定した場合，Gary Robinson 方式を使った検出結果は**表 3** のようになる．表 3 の結果から，最大誤検出率が相対的に低いのは *strings*=4 のときである．これは *strings* の最短長を長くした場合，その長さより短い特徴がすべて捨てられ重要な情報が大量に失われることで，誤検出率が高くなっているものと考えられる．Gary Robinson 方式はすべての特徴の確率を計算に用いるので，すべての情報を有効に利用して正確に判断するには情報が失われることは避けるべきである．そこで，本研究では *strings*=4 と設定する．

**表 3** Gary Robinson における strings の最短長の影響
Table 3 Influence of length of strings in Gary Robinson's method.

| *strings* の最短長 | Chain | 最大誤検出率 |
|---|---|---|
| 4 | 7 | 2.0% |
| 5 | 9 | 2.5% |
| 6 | 6 | 10.5% |
| 7 | 6 | 11.5% |
| 8 | 4 | 18.5% |
| 9 | 6 | 13.5% |
| 10 | 6 | 14.5% |
| 11 | 6 | 16.5% |
| 12 | 6 | 15.5% |
| 13 | 5 | 13.0% |
| 14 | 5 | 14.5% |
| 15 | 5 | 15.0% |
| 16 | 5 | 16.5% |
| 17 | 5 | 16.0% |
| 18 | 6 | 19.0% |
| 19 | 5 | 21.5% |
| 20 | 5 | 23.5% |

#### 5.1.2 パラメータ *s*

式 (3) 中のパラメータ $s$ の最適値を決定するために，*strings* の最短長が 4 から 20 までのそれぞれに対し，$s$ を 0.0，0.1，0.2，0.3，0.4，0.5，0.6，0.7，0.8，0.9，1.0 に設定し実験を行った．**表 4** には，その一部として *strings*=4，12，20 の結果を示す．結果から，パラメータ $s$ が大きくなると誤検出率も高くなることが判明した．そのため，本研究では $s$=0 と設定する．

### 5.2 計算式の改良

前述の実験から求めた最適なパラメータを設定しても，依然十分な性能を得ることは難しい．そこで，より少ない Chain 数と低い最大誤検出率を得るために，我々は次のように計算式を改良した新しいベイズ方式を提案する．

Gary Robinson 方式に最適な環境として *strings*=4，$s$=0 を設定した場合に，特徴ごとのウイルス出現確率を $p(w_i)$，全特徴の $p(w_i)$ の平均値を $x$ とすると，シグネチャに存在しない特徴のウイルス確率は $x$ が使用される．そのうえで，Gary Robinson 方式の計算式を以下のように改良した．

$$fpz = \left\{ \prod_{i=1}^{n} (1 - p(w_i)) \right\}^{1/n} \tag{10}$$

$$fqz = \left\{ \prod_{i=1}^{n} p(w_i) \right\}^{1/n} \tag{11}$$

$$p(w_i) = \frac{b/n_{bad}}{b/n_{bad} + g/n_{good}} \tag{12}$$

$$fqz > \frac{1-x}{1+x} * fpz \tag{13}$$

**表 4** Gary Robinson における $s$ の影響（*strings* の最短長=4，12，20）
Table 4 Influence of variable $s$ in Gary Robinson's method.

| $s$ | *strings* の最短長=4 | | *strings* の最短長=12 | | *strings* の最短長=20 | |
|---|---|---|---|---|---|---|
| | Chain | 最大誤検出率 | Chain | 最大誤検出率 | Chain | 最大誤検出率 |
| 0.0 | 8 | 0.5% | 6 | 8.0% | 5 | 15.0% |
| 0.1 | 8 | 0.5% | 6 | 10.5% | 5 | 18.0% |
| 0.2 | 8 | 0.5% | 6 | 10.5% | 5 | 19.0% |
| 0.3 | 8 | 1.0% | 6 | 11.0% | 5 | 21.0% |
| 0.4 | 8 | 1.0% | 6 | 11.0% | 5 | 21.5% |
| 0.5 | 8 | 1.0% | 6 | 12.0% | 5 | 22.0% |
| 0.6 | 8 | 1.0% | 6 | 12.5% | 5 | 22.0% |
| 0.7 | 8 | 1.5% | 6 | 13.0% | 5 | 22.5% |
| 0.8 | 8 | 1.5% | 6 | 13.5% | 5 | 22.5% |
| 0.9 | 8 | 2.0% | 6 | 14.5% | 5 | 23.0% |
| 1.0 | 7 | 2.0% | 6 | 15.5% | 5 | 23.5% |

このとき，式 (13) が成り立てば，そのファイルをウイルスファイルと判断する．なお，平均値 $x$ は判断するウイルスファイルの中に現れ，ウイルスシグネチャとノンウイルスシグネチャに存在するすべての特徴の確率の平均値であり，このファイルがウイルスである可能性をある程度示している．未知ウイルスファイルには，ウイルスシグネチャとノンウイルスシグネチャ両方ともに存在しない特徴がいくつか存在するはずだが，このような特徴を naive 方式のように単純に無視したり，Paul Graham 方式のように 0.4 で設定したりすると客観的な判断結果が得られないと考えられる．そこで，シグネチャに存在しない特徴の確率も計算により求める Gary Robinson 方式が有効となってくる．

一般にスパムフィルタにおいては，誤検出率を抑えることは重要である．すなわち使用者にとってはスパムファイルをノンスパムファイルと判定されるよりも，1 件のノンスパムファイルをスパムファイルとして削除されることの方が損失が大きいためである．しかしウイルスの場合には事情が異なり，ウイルスファイルがノンウイルスファイルと判断された場合，使用者のコンピュータは使用不能になる可能性もある．そこで，改良した計算式ではファイル検出の際に，ウイルスファイルかノンウイルスファイルかを強調するために，強調係数 $(1-x)/(1+x)$ を導入した（式 (13)）．これにより，平均値 $x$ のウイルス確率が高ければ高いほど，係数 $(1-x)/(1+x)$ の値が低くなり，$((1-x)/(1+x)) * fpz$ のノンウイルス確率も低くなるので，ウイルスファイルと判断されやすくなる．逆に $x$ が低いと，係数 $(1-x)/(1+x)$ は高くなり，$((1-x)/(1+x)) * fpz$ のノンウイルス確率も高くなるので，ノンウイルスファイルと判断されやすくなる．提案したベイズ方式を使った検出結果は，Chain=5，最大誤検出率=1.5%となり，想定した許容範囲内の性能となった．これは，Chain の少なさと低い最大誤検出率という両方から判断するに，最も良好な検出結果である．

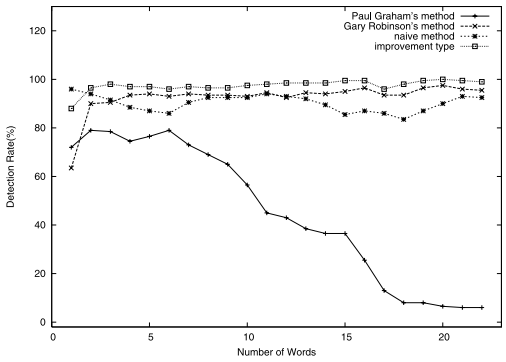


**図 5** 3 種類の方式と改良式のスパムファイルの検出率
Fig. 5 Detection rate of spam of three kinds of methods and improvement type.

提案したベイズ方式により，ウイルスフィルタについては検出結果の改善が見られたので，スパムフィルタについてもこの式を用い，3 章と同様の実験を行った．実験結果の検出率を**図 5** に示すが，生成したスパムフィルタは今まで主に利用されたベイジアンフィルタの 3 種類の方式，Paul Graham 方式，Gary Robinson 方式，naive 方式よりも，ほとんどの部分で検出率が高いという結果が得られた．また，誤検出率は従来の Gary Robinson 方式同様，0.5%以下という十分に低い値となった．以上のことから提案したベイズ方式に基づくベイズ学習アルゴリズムは，従来の 3 種類のベイジアンフィルタよりも，スパムとウイルス両方に対して優れた性能を示すフィルタとなることが確認された．

## 6. むすび

本論文では，ベイズ学習アルゴリズムを用いたスパムとウイルス両方のフィルタにおいて有効に機能するベイズ方式に関する考察と，Gary Robinson 方式を改良した既存の方式よりも高性能なベイズ方式の提案を行った．実際のスパムやウイルスファイルを用いた実験により，提案したベイズ方式は従来の Paul Graham 方式，Gary Robinson 方式，naive 方式に比べ，スパムフィルタとしては従来方式と同等の誤検出率を維持したまま，より高い検出率が実現可能であることが示された．さらにウイルスフィルタとしては，従来方式では誤検出率の上昇なしでは実現不可能であった高い検出率が，最大でも 1.5%という実用的なレベルの誤検出率で得られることが確認された．また，提案したスパムとウイルス両方において高い性能を示すベイズ方式は，メールに対しスパム検出と同時にウイルス検出を行うシステムのハードウェア化などを考えた場合，コストの面で大きなアドバンテージになるものと考えられる．

## 参考文献


1) IT 用語辞典 e-Words．http://e-words.jp/w/E382B9E38391E383A0.html
2) Burns, E.: The Deadly Duo: Spam and Viruses (June 2006). http://www.clickz.com/showPage.html?page=3622936
3) GFI Software: Why Bayesian filtering is the most effective anti-spam technology (2006). http://www.gfi.com/whitepapers/why-bayesian-filtering.pdf
4) Kantor, A.: Bayesian spam filters use math that works like magic (2004). http://www.usatoday.com/tech/columnist/andrewkantor/2004-09-17-kantor_x.htm
5) Greg's Bogofilter Page. http://www.bgl.nu/bogofilter/
6) POPFile-Automatic Email Classification. http://sourceforge.net/docman/display_doc.php?docid=13334&group_id=63137
7) POPFile Automatic Email Sorting using Naive Bayes. http://popfile.sourceforge.net/old.html
8) IT Security Center: *Computer virus incident report for the 3rd quarter* (*July to September*) *of 2004*, Online Publication, Information-technology Promotion Agency (IPA/ISEC) (2005).
9) IT Security Center: *Computer Virus Incident Reports* [*Details*], Online Publication, Information-technology Promotion Agency (IPA/ISEC) (2006). http://www.ipa.go.jp/security/english/virus/press/200605/virus200605-e.html
10) 中谷直司，小池竜一，厚井裕司，吉田等明：メール型未知ウイルス感染防御ネットワークシステムの提案，情報処理，Vol.45, No.8, pp.1908–1920 (2004).
11) McAfee: An Introduction to Computer Viruses (and Other Destructive Programs). http://www.mcafee.com/common/media/vil/pdf/av_white.pdf
12) Smith, J.: My Computer Virus Paper (2001). http://uhavax.hartford.edu/JUSMITH/paper.htm
13) 小池竜一，中谷直司，萩原由香里，厚井裕司，高倉弘喜，吉田等明：ベイズ学習アルゴリズムを用いた未知のコンピュータウイルス検出手法，情報処理，Vol.46, No.8, pp.1984–1996 (2005).
14) 渡部 洋：ベイズ統計学入門，福村出版 (1999).
15) 池田真雄，松井 敬，富田幸弘，馬場善久：統計学—データから現実をさぐる，内田老鶴圃 (1991).
16) 越 昭三：数理統計概論，学術図書出版社 (1983).
17) 森田優三：統計数理入門，日本評論社 (1968).
18) Pantel, P. and Lin, D.: A Spam Classification and Organization Program, *Proc. AAAI-98 Workshop on Learning for Text Categorization 95-98*, AAAI Press (1998).
19) Graham, P.: A Plan For Spam (2002). http://www.paulgraham.com/spam.html
20) Graham, P.: Better Bayesian Filtering (2003). http://www.paulgraham.com/better.html
21) Robinson, G.: Spam Detection (2002). http://radio.weblogs.com/0101454/stories/2002/09/16/spamDetection.html
22) Robinson, G.: A Statistical Approach to the Spam Problem (2003). http://www.linuxjournal.com/article/6467
23) Joachims, T.: A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization, *Proc. 14th International Conference on Machine Learning*, pp.143–151 (1997).
24) Focus on Linux. http://linux.about.com/library/cmd/blcmdl1_strings.htm


(平成 18 年 11 月 24 日受付)
(平成 19 年 3 月 1 日採録)

**王　卉歓**

2001 年高知大学大学院理学研究科博士前期課程修了．2005 年岩手大学大学院工学研究科博士後期課程入学，現在に至る．ネットワークセキュリティに関する研究に従事．

**中谷　直司**

1994 年埼玉大学工学部電子工学科卒業．1996 年同大学大学院理工学研究科博士前期課程修了．1999 年同大学院理工学研究科博士後期課程修了．同年岩手大学工学部情報システム工学科教務職員．2001 年同科助手，現在に至る．進化型アルゴリズム，ネットワークセキュリティに関する研究に従事．博士（学術）．電子情報通信学会会員．

**小池　竜一**（学生会員）

2003 年岩手大学工学部情報工学科卒業．2005 年同大学大学院工学研究科博士前期課程修了，同年同大学院工学研究科博士後期課程入学，現在に至る．ネットワークセキュリティに関する研究に従事．電子情報通信学会学生会員．

**厚井　裕司**（正会員）

1970 年東京理科大学理学部応用物理学科卒業．同年三菱電機（株）入社．2001 年岩手大学工学部情報システム工学科教授，現在に至る．主として，マルチメディアネットワーク，ネットワークセキュリティ，RF-ID タグに関する研究に従事．工学博士．IEEE，電子情報通信学会各会員．

**朴　美娘**（正会員）

1983 年漢陽大学工学部電子工学科卒業．同年漢陽大学工学部助手．1993 年東北大学大学院工学研究科情報工学専攻博士後期課程修了．同年東北大学電気通信研究所助手．1994 年三菱電機株式会社入社．現在，同社情報技術総合研究所勤務．通信プロトコル設計，ネットワークセキュリティ，移動通信ネットワーク等の研究に従事．博士（工学）．電子情報通信学会会員．