O-011

大規模スパムフィルタと実験環境の構築手法の提案

佐藤 一道　　脇田 建

東京工業大学情報理工学研究科
E-mail: {sato3, wakita}@is.titech.ac.jp

1 はじめに

米国IronPort System社の調査によると、2006年6月に世界で送信された未承諾広告メールの件数は550億通に達しており[9]、電子メールシステムの利便性に対する脅威となっている。未承諾広告メールには様々な呼び方があるが、本稿ではこのような迷惑なメッセージをスパム、スパム以外のメッセージをハムと呼ぶものとする。

スパムへの対策として、受信したメッセージをスパムとハムに分類するスパムフィルタがあり、さまざまな手法が提案されている[2]。IPアドレスに関するブラックリストフィルタやホワイトリストフィルタ、文献密度[8]を応用し、メッセージのベクトル空間上の密度を計算することによる手法[10]、IPアドレスやメッセージに含まれるURL、単語のマッチングなど複数のルールを用いて分類する手法[1]、強調フィルタリング[5]を用いた手法[4]やメッセージに出現する単語の頻度を統計的に解析するベイジアンスパムフィルタ[3, 6, 7]などが挙げられる。スパムフィルタの多くは個人向けであるが、スパムが増加してきたことによってスパムフィルタをメールサーバ上に設置し、多人数をサポートするような動きがでてきた。このようなスパムフィルタは大量のメッセージを処理しなければならないので、分類精度よりも処理速度を優先する傾向にある。一方で、個人向けに開発されたベイジアンスパムフィルタは高い分類精度と柔軟性を持っているが、処理速度の問題から大規模化は困難である。

われわれはベイジアンスパムフィルタを大規模化し、分類精度に関して妥協しない大規模スパムフィルタの構築を目標としている。大規模ベイジアンスパムフィルタを構築する上での最大の障害は効率的なメモリの使用である。既存のベイジアンフィルタ[11]では個々のプロファイルが十数MBに及んでしまう。そのため安易に大規模化すると、数GBのメモリを持つ一般的なサーバではメモリ上にすべてのプロファイルを表現することができず、処理速度が劇的に低下してしまう。そこで、本稿ではベイジアンスパムフィルタのユーザプロファイルを共有することによる大規模化手法を提案する。この結果、嗜好の類似したユーザ同士が同じプロファイルを共用することになり、分類精度を犠牲にせずメモリの使用量を劇的に減らすことができる。これによって頻繁に使用されるプロファイルを2次キャッシュに収めることができ、処理速度が向上する。

このような大規模スパムフィルタの有効性を評価するにあたり、統計的に意味のある実ユーザの協力が不可欠であるが、プライバシーの問題から困難である。そこで、本稿ではメーリングリストを用いた仮想ユーザの作成手法を提案する。この提案によって近似的ではあるが大規模な実験評価環境を構築することができる。

本稿では仮想ユーザの作成手法を用いて1,000人規模および3,000人規模の評価環境を構築し、プロファイル共有の有効性を評価した。実験の結果1,000人規模において36個、3,000人規模において26個のプロファイルを用いれば十分な分類精度を得られることが確認でき、メモリ使用量を劇的に削減することができた。

2 ユーザプロファイル

本節ではベイジアンスパムフィルタで用いられるユーザプロファイルの構造について簡単に述べる。

ベイジアンスパムフィルタはまず、あらかじめスパムとハムに分類されたメッセージの集合を学習し、ユーザプロファイルを作成する。ユーザu_iのプロファイルDB_iはu_iが持つメッセージに含まれる単語とそれに対応するスパム確率の組の集合であり、スパム確率はスパム、ハムに含まれる単語の出現頻度からベイズの定理によって計算される。新たにメッセージが届いた場合、メッセージに含まれる単語のスパム確率をプロファイルから取得し、その複合確率の高さによってスパム判定を行う。

単語を文字列として処理することは非効率的である。そこで、本研究ではメッセージから単語を抽出する際にハッシュ関数を用いて単語を32ビットの自然数値で表現し、プロファイルを単語をキー、スパム確率を値とするハッシュテーブルとすることで処理の高速化を行った。本研究ではプロファイルのエントリ数を約170万とした。また、スパム確率は通常64ビットの浮動小数点数で表現されるが、実際にメッセージを分類する上でこれ程の精度は必要ない。そこで、スパム確率の精度を抑え16ビットで表現することによってプロファイルの圧縮を行った。これによって1ユーザのプロファイルのサイズを13.6MBから3.4MBまで圧縮することができる。

3 共有アルゴリズム

本節ではユーザu_iのDB_iをu_jと共有することによるメモリ使用量の削減手法を提案する。

DB_iをu_jのものとして代用できるということは、DB_iを用いてu_jに届くメッセージを高い精度で分類できるということである。そこで、以下のようなu_jのメッセージM_jに対するDB_iの分類精度の指標$Error(i,j)$を導入する。

$$Error(i,j) = \frac{fp(DB_i, M_j) + fn(DB_i, M_j)}{|Spam(M_j)| + |Ham(M_j)|}$$

ここで、$fp(DB_i, M_j)$、$fn(DB_i, M_j)$はそれぞれDB_iを用いてM_jを分類したときにハムをスパムとする誤

検出数とスパムをハムとする見逃し数である。また、$Spam(M_j)$、$Ham(M_j)$ はそれぞれ u_j の持つ全メッセージの中でスパム、ハムに分類されるメッセージの集合である。

$Error(i,j)$ が十分に小さい場合、DB_i を u_j のプロファイルとして代用できると考えられる。そこで、十分に小さい $\varepsilon\,(>0)$ を用いて $Support(\varepsilon) = \{(i,j)|Error(i,j) < \varepsilon\}$ というユーザ間の二項関係を定義する[1]。ここで、$(i,j) \in Support$ ならば、u_i は u_j をサポートすると言う。

プロファイルを共有するために、直感的には全ての 2 ユーザの組み合わせに対して総当りで $Error(i,j)$ を計算し、$(i,j) \in Support$ となる組 (i,j) を見つけ類型的ユーザ u_i を決定すればよいと考えられるが、計算コストが $O(n^2)$ となってしまうため現実的ではない。そこで、サポートする人数が多いと考えられる候補者をあらかじめ見つけておき、その候補者群の中から類型的ユーザを見つける手法を提案する。

まず、全ユーザから無作為に k 人の仮想ユーザを選択する。これらのユーザを標本ユーザと呼ぶ。次に全ユーザの標本ユーザ群に対するサポート人数を調べ、多ければ候補者とする。候補者の選出の後、候補者と全ユーザとの $Support$ を調べ、類型的ユーザを決定する。この手法を総当りに対して絞込みと呼ぶ。n 人のユーザから候補者が l 人選出されたとすると、$Error(i,j)$ の計算回数は $(k+l)\times n$ となる。k と l を n に対して十分小さくすることにより総当りと比較して計算コストを削減することができる。

ここで、総当り、絞込みの手法において u_j をサポートするユーザが複数出現する可能性がある。このような場合、$Error(i,j)$ が最も小さいユーザのプロファイルを u_j のものとする。

4 メーリングリストを用いた仮想ユーザの作成手法

大規模スパムフィルタの評価には、多数の実ユーザの協力が必要であるが、プライバシーの問題から現実的には困難である。本節では大規模評価環境の構築手法を提案する。

現実のメールユーザが受信するメッセージは言葉遣いや内容がそれぞれ異なっている。このことをメッセージの個性と表現する。理想的な実験環境構築に求められることとして、各仮想ユーザの持つハムに個性を与えなければならないというものがある。ただし、スパムは同じメッセージを不特定多数のユーザに送信しているため、個性を与えなくてもよい。

そこで、メーリングリストを用いた仮想ユーザの作成手法を提案する。メーリングリストはコミュニケーションの場として広く用いられており、それぞれのテーマも多岐にわたっている。つまり、メーリングリストに流れるメッセージには十分な個性があるため、多数のメーリングリストから様々な個性を持つメッセージを容易に収集することができる。

[1] $Support$ に対して対称律、推移律が成立しないことが実験で確認できた

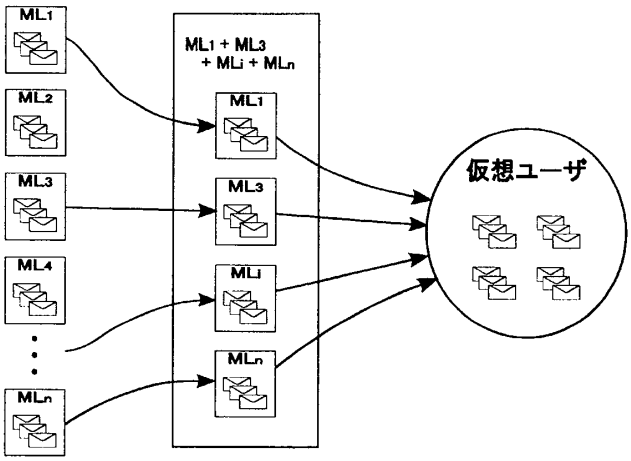

図 1 仮想ユーザの作成手法

仮想ユーザの作成手法として、まず n 個のメーリングリストに登録する。次にそのメーリングリストから k 個選び集合を生成する。そのメーリングリストの集合に含まれる全メッセージを仮想ユーザのハムとする。図 1 は仮想ユーザの作成手法の流れを表している。

この手法を用いることによって、第 0 次近似的ではあるが、多数の実ユーザの協力を得ることなく大規模スパムフィルタの実験環境を構築することができる。

しかし、この手法によって作成された仮想ユーザのメールヘッダの中には、題名や送信経路など常に同じ単語が含まれているものがある。これによってベイジアンスパムフィルタの分類精度に大きな影響を与えてしまい、正しい評価結果が得られない。そこで、本稿ではメールヘッダの中で分類精度に影響を及ぼさず、重要な情報である *From*、*Content-Type*、*Content-Transfer-Encoding* および重要ではないが分類精度に影響がない *Mime-Version*、*Status* のみを解析することによって正しい評価結果を得ることができるようにした。

5 評価

本節では提案したプロファイル共有手法の評価を行う。実験環境は Intel Xeon 2.8GHz 2CPU × 2 スレッド、2MB の 2 次キャッシュ、8GB のメモリ、Linux2.6 の計算機、実装には Java 言語を用いた。評価環境としてハム、スパムにそれぞれ約 300 通与えた仮想ユーザを 3,000 人作成して評価実験を行った。また、$Support(0.01)$ とした。ただし、共有後にスパムの誤検出が多く発生してしまうのを防ぐために、誤検出が発生した場合は $Support$ は成立させないものとした。被テストユーザ数は全仮想ユーザ数の 20%、候補者の選出条件は被テストユーザ群のうち 80%以上の人数をサポートすることとした。

5.1 共有手法の潜在的有効性

3 節で提案した共有手法を評価する前に、潜在的にプロファイルの共有が可能であるかを確認した。潜在的有効性を評価するために、1,000 仮想ユーザに対して総当りで $Support$ を調べ、プロファイルの共有を行った。表 1、2、3 はそれぞれ共有結果、誤検出率、見逃し率を比

表 1　ナイーブなフィルタと総当り、絞込みでのプロファイル共有の結果の比較

	ナイーブ		総当たり	絞込み (10 回平均)		絞込み (最良結果)	
仮想ユーザ数（人）	1,000	3,000	1,000	1,000	3,000	1,000	3,000
被テストユーザ数 (人)	—	—	—	50	150	50	150
候補者数 (人)	—	—	1000	47	18	43	7
プロファイル数 (個)	1,000	3,000	13	31	26	32	20
プロファイルの総容量 (MB)	3,400	10,200	44	105	88	109	68
$Error(i,j)$ の計算回数 (回)	—	—	1.0×10^6	9.7×10^4	5.0×10^5	9.3×10^4	4.7×10^5
類型的ユーザ発見に要した時間 (分)	—	—	617	46	237	43	183

表 2　スパムの誤検出率の比較

誤検出率	ナイーブ		総当たり	絞込み	
	1000	3000	1000	1000	3000
0.0%	837	2511	855	838	2462
0.0% - 0.1%	0	0	0	0	0
0.1% - 0.2%	1	1	1	1	2
0.2% - 0.3%	66	207	61	68	210
0.3% - 0.4%	73	227	62	73	262
0.4% - 0.5%	22	53	20	20	64
0.5% -	1	1	2	0	0
平均 (%)	0.054	0.052	0.048	0.052	0.058

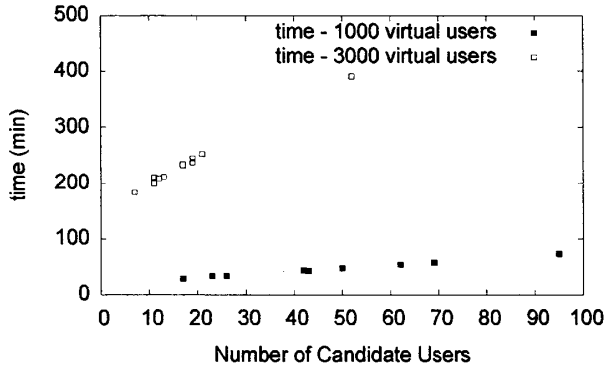

図 2　絞込みにおける候補者数と類型的ユーザ発見に要する時間

較したものであり、自身のプロファイルを用いた理想的なスパムフィルタをナイーブなフィルタと呼ぶ。

表 1 より、共有後のプロファイル数を 13 個まで減らせることが確認できた。また表 2、3 より、プロファイル数が約 1/77 であるのにもかかわらず分類精度が低下していないことがわかる。ナイーブなフィルタと比較して誤検出数が減少し、見逃し数が増加している原因としては *Support* の成立条件として誤検出数に重点をおいていることが考えられる。これらの結果より潜在的にプロファイルの共有が可能であることが確認できた。本稿では 3,000 人規模に対して絞込みによるプロファイル共有を行っているが、総当りによるプロファイル共有には 92 時間程度の時間が必要であると考えられるため行わなかった。

表 3　スパムの見逃し率の比較

見逃し率	ナイーブ		総当たり	絞込み	
	1000	3000	1000	1000	3000
0%	246	682	152	209	342
0% - 1%	403	1453	335	384	1281
1% - 2%	275	646	352	293	946
2% - 3%	58	157	104	83	304
3% - 4%	11	31	33	21	87
4% - 5%	4	16	22	6	34
5% -	3	15	2	4	6
平均 (%)	0.80	0.80	1.15	0.94	1.14

5.2　1,000 仮想ユーザにおける実験

1,000 仮想ユーザに対して絞込みを用いてプロファイルを共有させ、同じ仮想ユーザ群を総当りで共有させた場合との比較を行った。ただし、絞込みでは実験ごとに被テストユーザ群や選出される候補者が異なる。本研究では絞込みに対しては 10 回の実験を行った。表 1 には 10 回の実験結果の平均とプロファイル共有後の分類精度が最も良い結果を、表 2、3 は分類精度が最良の結果のみを記した。

実験の結果、共有後のプロファイル数は平均 31 個、分類精度が最良のもので 32 個となった。類型的ユーザがサポートする人数の中で誤検出率が 0.5%以上、または見逃し率が 5%以上の仮想ユーザを共有失敗として自身のプロファイルを使用させると、最良結果において 36 個のプロファイルで十分な分類精度が得られることがわかる。これは総当りと比較すると約 2.4 倍のプロファイル数となったが、ナイーブなフィルタと比較してプロファイル総容量を 3.4GB から 122MB へと減らすことができ、96.4%の圧縮ができた。

10 回の実験の中で最も高い分類精度を示したものと総当りでの分類精度を比較すると、誤検出率は同程度、見逃し率は減少する結果となった。これは総当りよりも共有後のプロファイル数が多くなったことによるものであると考えられるが、10 回の実験の中で総当りよりも良い分類精度を持つものは 1 つだけであった。この原因は候補者が全ユーザ数に対して小さいため、1 ユーザをサポートするユーザが少なくなってしまいより良い分類精度を持つプロファイルを見逃しているためである。これは類型的ユーザがサポートする人数で最大のものが総当りでは 492 人であるのに対し絞込みでは平均 502 人と、

1つの類型的ユーザに多くのユーザがサポートされていることからも他により良い類型的ユーザが存在することがわかる。

類型的ユーザ発見のために $Error(i,j)$ を計算した回数は総当りと比較して約1/10となり、結果として計算時間も46分と総当りと比較して実用的な時間でプロファイルを共有することができた。

5.3 3,000仮想ユーザにおける実験

3,000仮想ユーザに対して絞込みによってプロファイル共有を行い、1,000仮想ユーザを絞込みで共有させた場合との比較を行った。

実験の結果、共有後のプロファイル数は平均26個、分類精度が最良のもので20個となった。1,000仮想ユーザの場合と同様に誤検出率が0.5%以上、見逃し率が5%以上のものは自身のプロファイルを使用することとすると、最良結果において26個のプロファイルで十分な分類精度が得られることが確認できた。これは1,000人規模の実験と比較しても大きく変化しない。すなわち、本方式がユーザ数の増加に対して優秀なスケーラビリティを持つといえる。

分類精度に関してナイーブなフィルタと比較すると、誤検出率は同程度、見逃し率は増加する結果となったが、これは1,000人規模と同様に、*Support* の成立のために誤検出数に重点をおいているためであると考えられる。

類型的ユーザ発見のために要した時間は平均で237分となり、3,000人規模を総当りで共有させた場合の予想時間5,553分(92時間)に対して約1/23の時間で共有を行うことができた。

5.4 候補者数と類型的ユーザ発見の発見時間

最後に絞込みによる10回のプロファイル共有実験における候補者数と類型的ユーザ発見に要した時間に関して考察する。図2より、1,000人規模での10回の実験において候補者数は17人から95人発見され、3,000人規模では7人から52人の候補者が発見されていることがわかり、発見された候補者数が実験ごとに大きく異なっていることがわかる。これによって類型的ユーザの発見に要する時間にも幅がでてしまい、実験によっては効率的に見つけられないことがわかった。また、ユーザの規模と実験に要した時間との比較をすると、規模が3倍になったものと比較して、実験に要した時間は平均で5.11倍と大きく増加していることがわかる。これは絞込みを用いたプロファイル共有手法は総当りと比較して非常に短い時間で共有が可能であるが、ユーザの規模が大きくなるにつれ実用的な時間で処理しきれなくなることを示している。このことからより効率的に候補者を発見するアルゴリズムの提案が要求される。

6 まとめ

本稿で提案したベイジアンスパムフィルタのプロファイル共有手法によって数GBのメモリを持つ一般的なサーバにおいて3,000人規模をサポートするスパムフィルタが構築可能であることを証明した。

実験の結果、1,000人規模においてプロファイルを36個、3,000人規模において26個までプロファイル数を削減することができ、メモリ使用量をそれぞれ96.4%、99.9%削減することができた。また、実験によって得られた共有プロファイル数はユーザの規模に依存せず提案手法に高いスケーラビリティがあることを確認した。

今後の課題として、より大規模な実験を行うこと、有効な候補者を効率的に発見することができるように共有アルゴリズムを改善すること、時系列的にメッセージを変更し、仮想ユーザの個性を変化させた場合の動的な共有ユーザの変更が挙げられる。また、ナイーブなスパムフィルタとプロファイル共有後のスパムフィルタの処理速度の比較、実験におけるパラメータの最適化や、ベイジアンスパムフィルタのスパム判定の閾値をユーザごとに自動生成することも今後の課題としていきたい。

本研究の一部は文部科学省科学研究費助成金(18300041号)の援助を受けています。本稿の草稿について貴重なコメントをいただいた吉田悦郎さんに感謝します。

参考文献

[1] The Apache SpamAssassin Project. http://spamassassin.apache.org/.

[2] J. Goodman, G. V. Cormack, and D. Heckerman. Spam and the ongoing battle for the inbox. *Communications of the ACM*, Vol. 50, No. 2, pp. 24–33, 2007.

[3] P. Graham. A Plan For Spam, 2002. http://www.paulgraham.com/spam.html.

[4] J. S. Kong, P. O. Boykin, B. A. Rezaei, N. Sarshar, and V. P. Roychowdhury. Scalable and Reliable Collaborative Spam Filters: Harnessing the Global Social Email Networks. In *Conference on Email and Anti-Spam*, p. 8, 2005. http://www.ceas.cc/papers-2005/143.pdf.

[5] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl. Grouplens: Applying collaborative filtering to Usenet news. *Communication of ACM*, Vol. 40, No. 3, pp. 77–87, 1997.

[6] G. Robinson. Spam Detection, 2002. http://radio.weblogs.com/0101454/stories/2002/09/16/spamDetection.html.

[7] G. Robinson. A Statistical Approach to the Spam Problem. *Linux Journal*, No. 107, 2003. http://www.linuxjournal.com/article/6467.

[8] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval.* McGraw Hill, 1983.

[9] IronPort System. Spammers Continue Innovation: IronPort Study Shows Image-based Spam, Hit & Run, and Increased Volumes Latest Threat to Your Inbox, 2006. http://www.ironport.com/company/ironport_pr_2006-06-28.html.

[10] K. Yoshida, F. Adachi, T. Homma, and H. Fujikawa. Density-Based Spam Detector. *Proc. 10th Conference on Knowledge Discovery and Data Mining*, pp. 486–493, August 2006.

[11] 鍋谷研一. bsfilter / bayesian spam filter / ベイジアンスパムフィルタ. http://bsfilter.org/.