2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology

Symbiotic Data Mining for Personalized Spam Filtering

Paulo Cortez, Clotilde Lopes
Dep. of Information Systems/Algoritmi
University of Minho
4800-058 Guim. Portugal
{*pcortez,clopes*}*@dsi.uminho.pt*

Pedro Sousa, Miguel Rocha
Dep. of Informatics/CCTC
University of Minho
4710-059 Braga, Portugal
{*pns,mrocha*}*@di.uminho.pt*

Miguel Rio
Dep. of Electrical Engineering
University College London
WC1E 7JE UK
m.rio@ee.ucl.ac.uk

Abstract

Unsolicited e-mail (spam) is a severe problem due to intrusion of privacy, online fraud, viruses and time spent reading unwanted messages. To solve this issue, Collaborative Filtering (CF) and Content-Based Filtering (CBF) solutions have been adopted. We propose a new CBF-CF hybrid approach called Symbiotic Data Mining (SDM), which aims at aggregating distinct local filters in order to improve filtering at a personalized level using collaboration while preserving privacy. We apply SDM to spam e-mail detection and compare it with a local CBF filter (i.e. Naive Bayes). Several experiments were conducted by using a novel corpus based on the well known Enron datasets mixed with recent spam. The results show that the symbiotic strategy is competitive in performance when compared to CBF and also more robust to contamination attacks.

1. Introduction

Nowadays, it is easy to collect, process and share data. The field of Data Mining (DM) deals with the extraction of knowledge (e.g. patterns) from raw data and it has been successfully used in distinct domains (e.g. marketing, medicine or Internet) [1]. Moreover, the Internet growth opened room for important communication and information sharing services (e.g. Web, e-mail). However, this expansion also led to severe problems, such as information overload and unsolicited e-mail (known as spam).

The cost of sending spam is tiny and it is easy to reach a high number of potential consumers [2]. Spam emerged very quickly after e-mail itself and currently over 120 billion of these messages are sent each day [3]. While e-mail is the most known form of spam, this phenomenon also affects other services, such as instant messaging (spim). Spam is an intrusion of privacy and many messages are of adult content, online fraud or viruses. Moreover, spam has costs in terms of traffic fees and time spent reading unwanted messages.

Currently, there are two major approaches to fight spam [4]: Collaborative Filtering (CF) and Content-Based Filtering (CBF). CF is based on sharing information about spam messages, while CBF uses a DM classifier (e.g. Naive Bayes) that learns to discriminate spam from message features (e.g. common spam words). CF can be based on blacklists [5], which contain IP addresses of known spam senders, or fingerprints extracted from spam messages [6]. Current research on spam CBF relies mainly on improving individual classifier performance, by a better preprocessing [4] or enhancement of the learning algorithm [7]. Ensembles that combine distinct spam classifiers have also been proposed [8].

Both CF and CBF have drawbacks. CF often suffers from sparsity of data (e.g. users may classify few messages) and first-rater problem (e.g. an e-mail cannot be classified unless a user has rated it before). Moreover, people have personal views of what is spam and CF often discards this issue [9]. On the other hand, in CBF there may be a large gap between low-level features (e.g. bit color) and high-level concepts (e.g. spam images). Furthermore, poor performances may be achieved by new users, since CBF requires several representative training examples. CBF is also vulnerable to dictionary or focused attacks, where the adversary can exploit DM models by contaminating the training set (e.g. by sending spam with a large amount of normal words) [10]. By fusing these two views there is a potential for a better personalized filtering. However, the number of studies that unify CBF and CF is scarce and mainly addressed towards recommendation systems that run at centralized systems [11].

We propose a novel Distributed DM (DDM) approach, based on a hybrid CBF-CF view and named Symbiotic Data Mining (SDM). The idea is to join distinct entities (e.g. e-mail users) interested on similar DM goals (e.g. spam filtering). Rather than exchanging data (e.g. messages), these entities will share information about what each local filter has learned (e.g. DM models). The aim of SDM is to foster mutual relationships, where all or most members benefit. Under SDM, each user is interested in improving filtering at a personal level. The Internet is used to gather collaborators among these (high number of) users. Users may dynamically join or leave this collaboration and there are privacy issues regarding what can be shared. The classic DDM approach [12] is targeted to obtain a scalable global DM model for a single entity by aggregating several local DM analyses. SDM is different from pure DDM and the centralized CBF-CF works (e.g. [11]) since SDM data and models are distributed through distinct entities. Hence, there are issues of user management (e.g. adding or removing a user), privacy, security and motivation (e.g. each user should

978-0-7695-3801-3/09 $26.00 © 2009 IEEE
DOI 10.1109/WI-IAT.2009.30

benefit from the collaboration).

In this paper, we apply SDM to spam detection and compare it with a local CBF filter (i.e. Naive Bayes). The remainder of this paper is structured as follows. Section 2 presents the e-mail data, local and symbiotic filtering methods, and evaluation metrics. Next, the results are presented and discussed (Section 3). Finally, closing conclusions are drawn (Section 4).

2. Materials and Methods

2.1. Spam data

Several public benchmark datasets to evaluate anti-spam filters have been proposed [13], such as : the Ling-spam[1], SpamAssassin[2], Spambase[3] and TREC 2005 Spam Track[4]. The first dataset mixed spam with legitimate messages (ham) collected from public archives (e.g. newsgroups), the second uses public fora or donations by users, the third includes only preprocessed features (e.g. word frequencies) and the fourth employed several filters to discard spam of the Enron e-mail [14] collection. As pointed by [13], none of these datasets is fitted for personalized filtering. Ham messages from Ling-spam are more topic-specific than what normal users tend to receive. In contrast, SpamAssassin ham messages are less-specific, since users tend to donate general (non-sensitive) messages. Spambase does not contain raw data (e.g. message date). Finally, the Spam Track was used for a global evaluation of filters, merging all user messages into a single corpus.

To evaluate SDM, ideally there should be real spam/ham messages collected from distinct users (possibly from a social network) during a given time period. Yet, due to logistic and privacy issues, it is quite difficult to obtain such data and make it public. Hence, we will use a synthetic mixture of real spam and ham messages, in a strategy similar to what has been proposed in [6], [13]. The ham messages will be related to five Enron employees with the largest mailboxes collected during the same time period. In particular, we will use the cleaned-up form provided by [14] of the mailboxes: kaminski-v (kam), farmer-d (far), beck-s (bec), lokay-m (lok) and kitchen-l (kit). Since these employees worked at the same organization, it is reasonable to assume that they would know each other, i.e. belong to a social network. We will also use the spam collection of Bruce Guenter[5], which is based in spam traps (i.e. fake emails published in the Web), during the years of 2006 and 2007 (our dataset was built in 2008). Only messages with Latin character sets were selected, because the ham messages use this type of character coding and non-Latin mails would be easy to detect. Also, since this collection contains several copies of the same messages (due to the use of multiple traps), we removed duplicates by comparing MD5 signatures of the body messages.

1. http://www.iit.demokritos.gr/skel/i-config/downloads/
2. http://spamassassin.apache.org/publiccorpus/
3. http://archive.ics.uci.edu/ml/datasets/Spambase
4. http://plg.uwaterloo.ca/~gvcormac/treccorpus/
5. http://untroubled.org/spam/

We propose a mixture algorithm that is based on the time that each message was received (date field, using the GMT time zone), which we believe is more realistic than the sampling procedure adopted in [13]. Since the Enron data is from a previous period (see Table 1), we first added 6 years to the date field of all ham messages. Let S_t denote a spam message received at time t, $S_{i,f} = (S_{t_i}, S_{t_{i+1}}, \ldots, S_{t_f})$ the time ordered sequence of the Bruce Guenter spam, $H_{u,i,f}$ and $S_{u,i,f}$ the sequences of ham and spam messages for user u from time t_i to t_f. For a given time period $t \in (t_i, \ldots, t_f)$, the algorithm randomly selects $|S'_{i,j}|$ spam messages from $S_{i,j}$. Then, $S_{u,i,j}$ is set by sampling messages from $S'_{i,j}$ with a probability of P for each message selection. The size of $S'_{i,j}$ (cardinality) is given by:

$$|S'_{i,j}| = \frac{R \cdot \sum_{i=1}^{L} |H_{u,i,j}|}{P \cdot L} \quad (1)$$

where L denotes the total number of users available at the time period and R is the overall (i.e. including all user and time data) spam/ham ratio. Since the time periods are different for each user (Table 1), four time sequences (i.e. t_i and t_j values) were used by the algorithm (Figure 1).

Table 1. Summary of the S-Enron corpus

user	ham size	spam size	time period	spam /ham
kam	4363	2827	[12/05,05/07]	0.6
far	3294	2844	[12/05,05/07]	0.9
bec	1965	2763	[01/06,05/07]	1.4
lok	1455	2202	[06/06,05/07]	1.5
kit	789	623	[02/07,05/07]	0.8

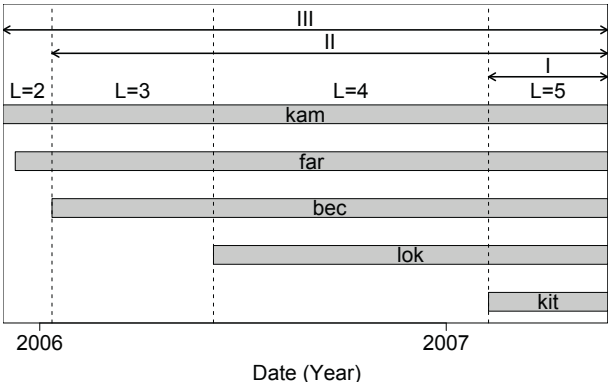

Figure 1. Temporal view of the S-Enron mailboxes.

The mixture is affected by the R and P parameters. Since a high number of experiments is addressed in this work,

we will fix these parameters to reasonable values. While the global spam/ham ratio is $R = 1$, the individual ratios range from 0.6 to 1.5. Also, the spam/ham ratios fluctuate through time (as shown in Figure 3). On the other hand, the probability of spam selection affects the percentage of common spam between users. If two users have similar profiles (e.g. e-mail exposure), then they should receive similar spam. We assume that this scenario is expected for the Enron employees and thus set $P = 0.5$. Under this setup and for a given time period, any 2 users will receive around 50% of similar spam, 3 users will share around 25% of spam and so on. The resulting corpus is named S-Enron and it is public available in its raw form at: http://www3.dsi.uminho.pt/pcortez/S-Enron.

2.2. Local filter

We will address only textual content (i.e. word frequencies) of e-mail messages. This popular approach (e.g. Thunderbird filter) has the advantage of being generalizable to wider contexts (e.g. spim detection). While different algorithms can be adopted for spam filtering, such as Support Vector Machines (SVM) [2], we will use the simpler Naive Bayes (NB), which is widely adopted by anti-spam filtering tools [4], [15]. As both individual and symbiotic strategies will be compared using the same learning algorithm, we believe that most of the results presented in this paper can be extended to other text classifiers. We will also adopt the preprocessing proposed in [16]:

1) The word frequencies are extracted from the subject and body message (with the HTML tags previously removed). Each message j is encoded into a vector $\mathbf{x}_j = (x_{1j}, \ldots, x_{mj})$, where x_{ij} is the number of occurrences of token X_i in the text.
2) The feature selection is applied, which consists in ignoring any words when $x_{ij} < 5$ in the training set and then selecting up to the 3000 most relevant features according to the Mutual Information ($MI(X_i)$) criterion:

$$MI(X_i) = \sum_{c \in \{s, \neg s\}} p(X_i|c) log \frac{p(X_i|c)}{p(X_i)p(c)} \quad (2)$$

   where c is the message class (s - spam or $\neg s$ - ham), $p(X_i|c)$ is the probability of finding token X_i in e-mails from class c, $p(X_i)$ and $p(c)$ are the proportions of X_i terms and c class examples present in the data.
3) Each x_{ij} value is transformed into: $x'_{ij} = log(x_{ij} + 1)$ (TF transform), $x''_{ij} = x'_{ij} \cdot log(k/\sum_k \delta_{ik})$ (IDF transform) and $x'''_{ij} = x''_{ij}/\sqrt{\sum_l (x_{lj})^2}$ (length normalization), where δ_{ik} is 1 if the token i exists in the message k and 0 otherwise.

The NB computes the probability that a document j is spam (s) for a filter trained over $\mathcal{D}_u$ email data from user u, according to:

$$p(s|\mathbf{x}_j, \mathcal{D}_u) = \alpha \cdot p(s|\mathcal{D}_u) \prod_i^m p(X_i|s, \mathcal{D}_u) \quad (3)$$

where α is normalization constant that ensures that $p(s|\mathbf{x}, \mathcal{D}_u) + p(\neg s|\mathbf{x}, \mathcal{D}_u) = 1$, $p(s|\mathcal{D}_u)$ is the $p(s)$ of dataset $\mathcal{D}_u$. The $p(X_i|s, \mathcal{D}_u)$ estimation depends on the NB version. In this work, we will use the multi-variate Gauss NB (as implemented in the **R** tool, see Section 3) [13]:

$$p(X_i|c, \mathcal{D}_u) = \frac{1}{\sigma_{i,c}\sqrt{2\pi}} e^{-\frac{(x'''_{ij} - \mu_{i,c})^2}{2\sigma^2_{i,c}}} \quad (4)$$

where $\mu_{i,s}$ and $\sigma_{i,s}$ are the mean and standard deviation estimated from the $c = s$ or $c = \neg s$ messages of $\mathcal{D}_u$.

In [10], it has been shown that local spam filters are vulnerable to dictionary and focused contamination attacks. The former attack is used to reduce the CBF efficiency, leading the victim to read spam, while the latter can be used to prevent the victim from reading an important email. Both attacks can be achieved by sending spam messages mixed with normal words. Once the victim labels these messages as spam, the training set is contaminated and the filter will be affected the next time it is retrained. A dictionary aggression consists in sending a large amount of normal words, while the focused assault assumes that the attacker has some knowledge of a specific message that the victim will receive in the future (e.g. a competing offer for a given contract).

2.3. Symbiotic filter

While sharing models is less sensitive than exchanging e-mail messages, there are still privacy issues. For instance, if user A has access to the filter of user B, then A may feed a given token (or set of tokens) into the model and thus know some probability that such token was classified by B as spam or ham. To solve this problem, we propose an anonymous distribution of the filters, under two possibilities: using a trustable application or a secure server. The former can be used in a Peer-2-Peer (P2P) setting, where the users trust the software to blindly share filters among all members, while the latter works under a centralized model where the anonymization is achieved by an intermediate secure server. Figure 2 shows example of both scenarios, where user C receives the local filters from users A and B without knowing who built each model. Since in SDM there will be typically a large number of users that dynamically may join or leave the collaboration, it will be very difficult to "guess" who created each model. The degree of privacy can be increased further if each user does not know who belongs to the symbiotic group. Regarding the filter sharing, a standard format should be adopted, such as the Predictive Model

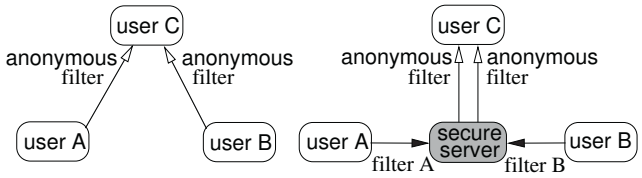

Figure 2. Blind exchange of filters by using a trustable application (left) or secure server (right).

Markup Language [17], which is compatible with a large number of DM products.

In SDM, the individual predictions can be combined by using a collaborative ensemble of the local filters. While several combination functions can be adopted (e.g. average), we propose a hierarchical learning, where the outputs of the local DM models are used as the inputs of another (meta-level) learner. Hence, each SDM user would have a local meta-learner that is dynamically trained to get a high accuracy on its past personal data. When compared with the equal weights (average) function, the meta-learner is more fitted for SDM, since it can dynamically (i.e. through time) assign different weights to distinct users (see Figure 5). While several algorithms could be used for the hierarchical learning (e.g. SVM), we will adopt the same NB described in the previous section. The rationale is that NB is commonly adopted by anti-spam solutions, thus incorporating SDM into these tools would be simpler by reuse of code.

We assume that each user u trains a local filter $\theta_{u,t}$ over her/his $\mathcal{D}_u$ training data. Filters can be trained asynchronously and L filters will be available for each user at time t: $\{\theta_{1,t}, \ldots, \theta_{L,t}\}$. The Symbiotic NB (SNB) meta-model spam probability is given by:

$$\begin{aligned} p(s|\mathbf{x}_j, \mathcal{D}'_u) &= \alpha \cdot p(s|\mathcal{D}'_u) \prod_{i=1}^{L} p(\theta_{i,t}|s, \mathcal{D}'_u) \\ p(\theta_{i,t}|c, \mathcal{D}'_u) &= \frac{1}{\sigma_{i,c}\sqrt{2\pi}} e^{-\frac{(p(s|\mathbf{X}_j, \theta_{i,t}, \mathcal{D}'_u) - \mu_{i,c})^2}{2\sigma^2_{i,c}}} \end{aligned} \quad (5)$$

where $\mathcal{D}'_u$ is the SNB training set and $p(s|\mathbf{x}_j, \theta_{i,t}, \mathcal{D}'_u)$ is the probability given by the filter $\theta_{i,t}$, as computed in Equation 3. To reduce memory and computational requirements, we allow that $\mathcal{D}'_u \subseteq \mathcal{D}_u$, where $M = |\mathcal{D}'_u|$ denotes the most recent messages from u mailbox. It should be noted that any token from $\mathbf{x}_j$ that is not considered by $\theta_{i,t}$ will simply be discarded by the filter from user i. Similarly, any input attribute from $\theta_{i,t}$ that is not included in $\mathbf{x}_j$ will be set to 0.

2.4. Evaluation

Since spam detection evolves through time (i.e. there is a concept drift), we will adopt the more realistic incremental retraining evaluation procedure, where a mailbox is split into batches $b_1, \ldots, b_n$ of K adjacent messages ($|b_n|$ may be less than K) [13]. For $i \in \{1, \ldots, n-1\}$, the filter is trained with $\mathcal{D}_u = b_1 \cup \ldots \cup b_i$ and tested the messages from b_{i+1}.

Figure 3 shows an example the evolution of the kam mailbox spam/ham ratio over different batches, with $K = 100$ and during the time period III of Figure 1.

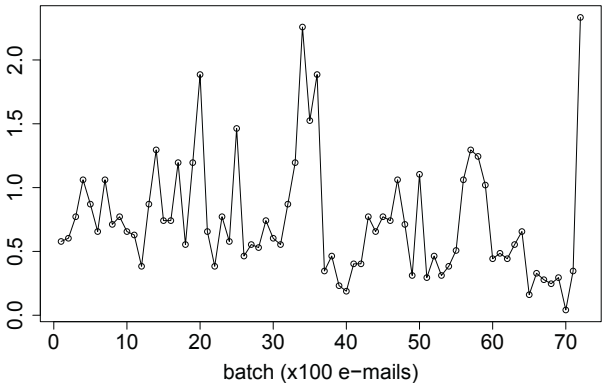

Figure 3. Evolution of the spam/ham ratio for the kam mailbox.

For a given probabilistic filter, the predicted class is given by s if $p(s|\mathbf{x}_j, \mathcal{D}_u) > D$, where $D \in [0.0, 1.0]$ is a decision threshold. For a given D and test set, it is possible to compute the true (TPR) and false (FPR) positive rates:

$$\begin{aligned} TPR &= TP/(TP + FN) \\ FPR &= FP/(TN + FP) \end{aligned} \quad (6)$$

where TP, FP, TN and FN denote the number of true positives, false positives, true negatives and false negatives, respectively. The receiver operating characteristic (ROC) curve shows the performance of a two class classifier across the range of possible threshold (D) values, plotting FPR (x-axis) versus TPR (y-axis) [18]. The global accuracy is given by the area under the curve ($AUC = \int_0^1 ROC dD$). A random classifier will have an AUC of 0.5, while the ideal value should be close to 1.0. Since the cost of losing normal e-mail (FP) is much higher than receiving spam (FN), D is usually set to favor points in the low false-positive region of the ROC. Thus, we will also adopt the metric TPR at a specific $FPR = r$ (denoted as TPR@FPR=r), where r is close to 0.0 [7]. With the incremental retraining procedure, one ROC will be computed for each b_{i+1} batch and the overall results will be presented by adopting the vertical averaging ROC (i.e. according to the FPR axis) algorithm presented in [18]. Statistical confidence will be given by the t-student test at the 95% confidence level [19].

3. Results

All experiments were conducted in the **R** environment, an open source and high-level programming language for data analysis [20]. In particular, the NB algorithm described in Section 2.2 is implemented by the naiveBayes function of the **e1071** R package, while the text preprocessing uses several functions from the **tm** package [21].

During the all experiments, we set $K = 100$ (a reasonable value also adopted in [13], [15]). For the SNB, we used a similar number for the hierarchical training set size, i.e. $M = 100$. This small value has the advantage of reducing memory requirements (the user only needs 100 messages in his mailbox) and some initial experiments with larger values of M revealed no gain in performance. All TPR@FPR values will be computed with $r = 0.01$ (1%).

3.1. Fixed symbiotic group

Two distinct scenarios will be tested, according to the time periods I and II of Figure 1. Given the S-ENRON corpus characteristics, in this work we will explore a small number of fixed symbiotic users: $L = 5$ for I and $L = 3$ for II.

The incremental retraining method (Section 2.4) was applied to both scenarios, by considering all messages within the corresponding time period. Thus, the number of kam, far and bec batches (n) will be different for I and II. The obtained results are summarized as the mean of all test sets (b_{i+1}, $i \in \{1, \ldots, n-1\}$) and shown in Table 2 and Figure 4. The best values are in **bold**, while underline denotes a statistical significance. In Figure 4, bars denote 95% t-student confidence intervals and only the most interesting region of FPR is shown for the ROC curves.

For the first scenario (I), the symbiotic strategy outperforms the local filter for all users and metrics, except for lok and TPR@FPR. A similar behavior occurs for the II setting, where SNB is better than NB except for kam and AUC. As false positives have higher costs in spam detection, the TPR@FPR results are particularly important. Thus, it is interesting to notice that there is a high TPR@FPR improvement given by the symbiotic method in several cases (kam, far and kit for I and bec for II).

To demonstrate the SNB dynamics, Figure 5 shows the first two consecutive graphs of the SNB input importances under scenario II. Each edge represents the influence (in %) of the NB filter (the origin) in the symbiotic model (the destination), as measured by applying a sensitivity analysis procedure [22]. The text in bold (e.g. $\mathbf{b_2}$) denotes the last batch used to train the NB classifier. For example, the first SNB model of user far (left graph) uses a NB filter from kam that was trained using 200 messages ($\mathcal{D}_{\text{kam}} = b_1 \cup \mathbf{b_2}$).

3.2. Incremental symbiotic group

A more realistic scheme is adopted for the time period III, where users join the symbiotic group in an incremental fashion, at different time stages according to Figure 1. Thus, L will grow from 2 to 5. The results are presented in Table 3. As expected, the symbiotic strategy (SNB) clearly favors newcomers, which have small mailboxes and thus benefit from the collaboration. In effect, the TPR@FPR differences are quite large, such as in bec and lok for $L = 4$ and $L = 5$

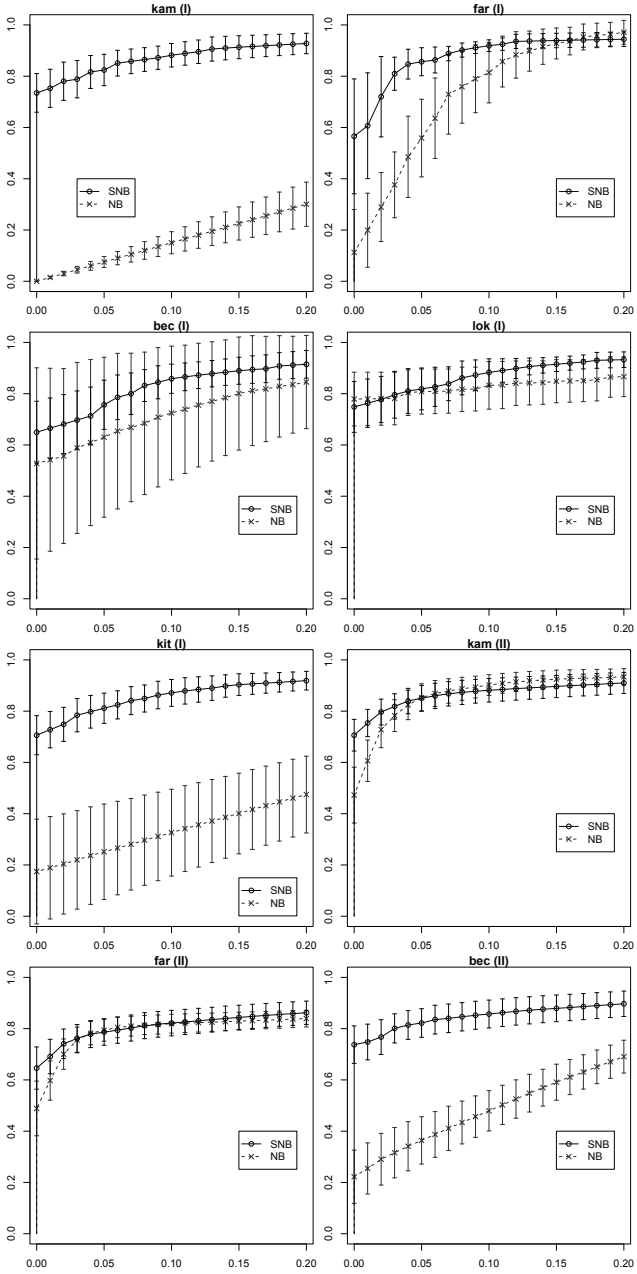

Figure 4. ROC curves for scenarios I and II.

and kit for $L = 5$. For demonstration purposes, the ROC curves are plot for bec, when $L = 3$ and $L = 5$ (Figure 6). However, the results show that even "veteran" users gain from the symbiotic relation when the number of users grow. For instance, the kam and far TPR@FPR results for $L = 5$ are 13.7 and 8.5 pp better.

3.3. Contamination attacks

SDM should be more robust to contamination, since it aggregates responses from several (possible unknown) users

Table 2. Summary of the results for I and II

	I					II				
		AUC		TPR@FPR			AUC		TPR@FPR	
user	n	**NB**	**SNB**	**NB**	**SNB**	n	**NB**	**SNB**	**NB**	**SNB**
kam	16	62.1	**95.6**	1.5	**75.3**	70	**94.7**	94.3	60.7	**75.4**
far	13	93.5	**95.1**	19.9	**60.7**	60	89.4	**91.7**	59.7	**69.1**
bec	9	91.5	**94.0**	54.2	**66.6**	48	83.5	**93.4**	25.5	**74.8**
lok	9	91.4	**95.2**	**78.0**	76.3	-	-	-	-	-
kit	15	74.6	**95.3**	18.9	**72.8**	-	-	-	-	-

Table 3. Summary of the results for III

	L=2				L=3				L=4				L=5			
	AUC		TPR@FPR		AUC		TPR@FPR		AUC		TPR@FPR		AUC		TPR@FPR	
user	**NB**	**SNB**	**NB**	**SNB**	**NB**	**SNB**	**NB**	**SNB**	**NB**	**SNB**	**NB**	**SNB**	**NB**	**SNB**	**NB**	**SNB**
kam	**94.4**	88.8	40.6	**51.8**	**91.5**	87.4	53.3	**60.8**	95.4	**96.8**	79.5	**79.7**	**98.4**	98.0	67.5	**81.2**
far	**89.9**	82.1	**60.7**	56.1	86.6	**87.0**	50.7	**59.9**	88.7	**94.7**	58.3	**74.6**	89.6	**95.5**	65.5	**74.0**
bec	-	-	-	-	80.4	**87.6**	58.4	**66.8**	84.5	**96.9**	15.3	**80.5**	85.2	**97.4**	3.6	**73.3**
lok	-	-	-	-	-	-	-	-	73.1	**96.2**	7.0	**74.0**	63.6	**97.3**	1.4	**79.2**
kit	-	-	-	-	-	-	-	-	-	-	-	-	74.6	**97.9**	18.9	**74.9**

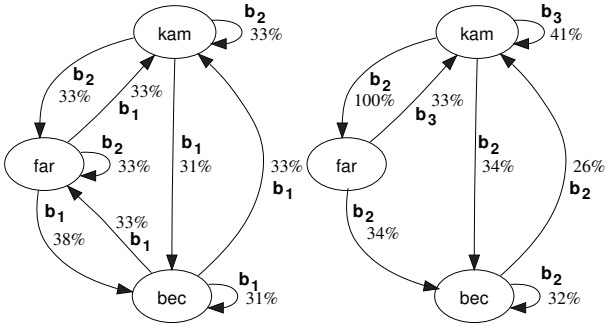

Figure 5. Examples of SNB input importances for II.

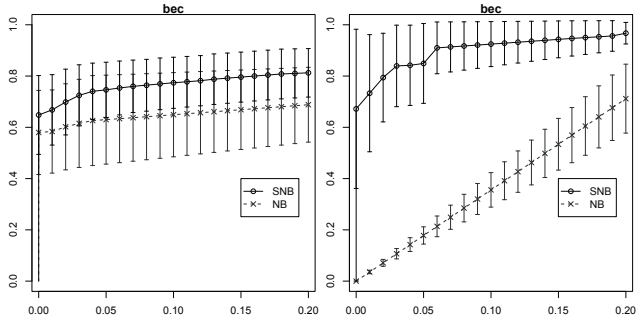

Figure 6. ROC curves for user bec and L=3 (left) and L=5 (right).

and targeting a specific victim may be easy but contaminating the whole symbiotic group is not. We will repeat the experiments of Section 3.1, by considering only scenario I and user bec to test the effects of mailbox contamination. The dictionary assault is simulated by replacing the first 10 spam emails at batch 4 from bec by the GNU aspell (http://aspell.net/) English dictionary (version 6.0, with 138599 tokens). In Figure 7, gray lines denote the behavior of NB and SNB without the attack (i.e. results of Section 3.1), black lines show the performance under the attack and the dot-dashed vertical line shows when the attack starts[6]. Local CBF is highly affected. Only 10 messages were replaced and yet the filter detection capability is reduced to a random classifier (since AUC=0.5) through all remaining batches. In contrast, the symbiotic method is only initially affected, since as time goes by the performance gets closer to the no attack scenario. Also, the remaining symbiotic users maintain their spam detection capabilities, as shown by the kit results, which is a representative example. This behavior is explained by the SNB algorithm, which simply discards a given filter if it does not help to predict the recent past M messages of the user. Hence, this experiment shows that SDM is robust also to saboteurs, i.e. if a particular user intentionally feeds the group with a random or bad filter then this filter will be simply ignored.

The dictionary attack can be solved by performing a roll-back (i.e. returning to the previous filter) or using the RONI defense [10], which rejects training examples that have a large negative impact in spam detection. However, focused assaults are much more difficult to prevent and finding an adequate defense is still an open problem [10]. We believe SDM is an interesting solution due to the same rationale presented for the dictionary aggression, i.e. the combination of multiple filters should overcome the limitations of a single model contamination.

6. For the bec user, the filters are not trained with contaminated messages at batch 4, yet these messages appear in the test set and thus the NB and SNB performances suffer a moderate decay. The true effect of the attack is only visible at batch 5.

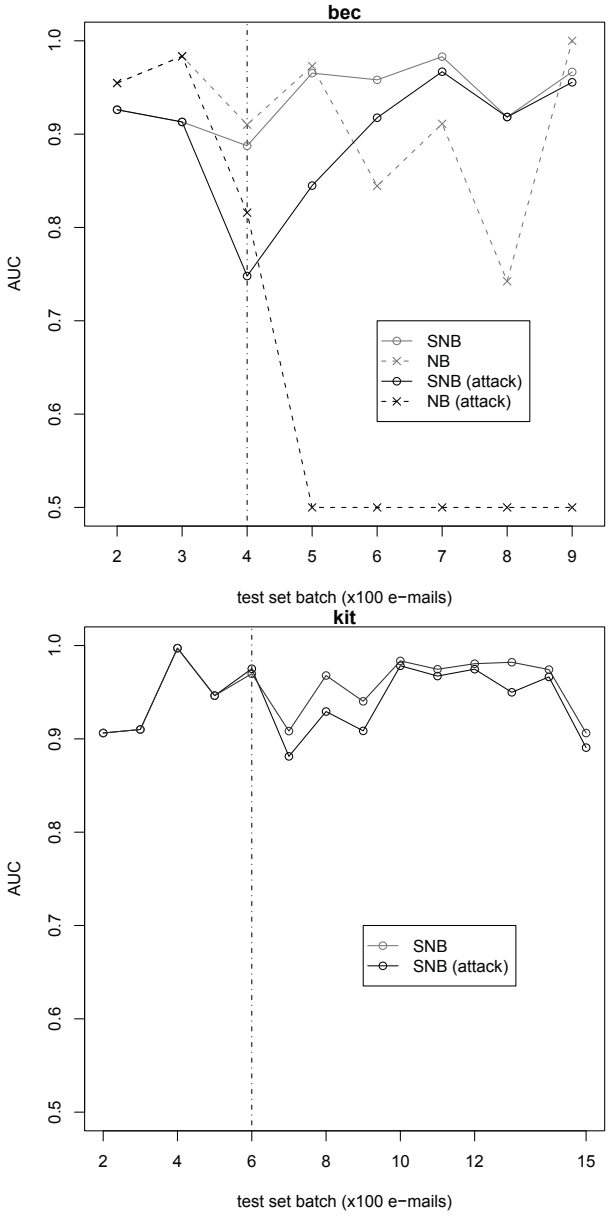

Figure 7. The dictionary attack effect.

A new set of experiments was devised, using again scenario I and bec mailbox. During a given run, a legitimate message was randomly selected, from batches 6 to 9, as the target text. We assume that the attacker is confident about the target content and thus can guess 50% of the target words. At batch 4, 10 spam emails were replaced by the contaminated messages. We repeated this procedure during 20 runs. The effect of this attack on spam is minimal and thus we will only show the effect on the target ham e-mails. Figure 8 plots the filter spam probability (y-axis) for each target message (total of 20 runs, x-axis). Since all target messages are ham, a robust filter should present low spam probabilities, near the zero horizontal axis. The results show that local filter (NB) is much more vulnerable than the symbiotic strategy (SNB). The spam probability mean values of NB and SNB are 0.69 and 0.32 (the differences are statistically significant). For example, when using a decision threshold of $D = 0.5$, 14 (of 20) messages are classified by NB as spam, while this number lowers to 6 for SNB. Even if D is raised to 0.999, NB predicts 13 spam e-mails and SNB only detects 5.

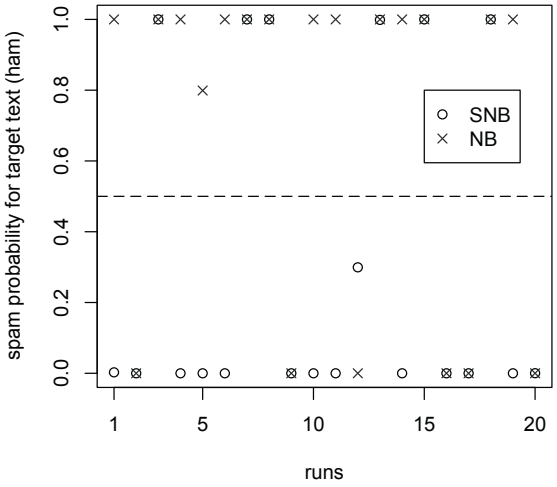

Figure 8. The focused attack effect.

4. Conclusions

We proposed a novel distributed data mining approach, called Symbiotic Data Mining (SDM), that unifies Content-Based Filtering (CBF) with Collaborative Filtering (CF). The goal is to reuse local filters from distinct entities in order to improve personalized filtering while maintaining privacy. As a case study, we apply the SDM strategy to spam e-mail detection. Several experiments were carried out under the new S-Enron corpus, which uses a realistic mixture of legitimate messages from five Enron employees with recent spam based in spam traps. We compared the performance of SDM and CBF using symbiotic groups with fixed and incremental number of users. Also, we simulated the effect of contamination attacks. Our results show that for a small number of users (i.e. 3 to 5), SDM outperforms personalized text classifiers. Furthermore, SDM is more robust to both dictionary and focused attacks. Within our knowledge, this is the first time a solution is proposed for the latter type of contamination.

Spammers and anti-spammers are in an arms race. As argued by [8], tuning a single classifier to perfection will

encourage spammers to eventually defeat it. By dynamically combining filters from distinct users, we believe that a stronger protection is achieved. In effect, SDM puts emphasis on accessing (indirectly) more data rather than using more complex local filters. As future work, we intend to study scalability issues. Under a large group, this could be achieved by adopting user selection algorithms (e.g. clustering user profiles). We believe SDM is also potentially useful in other personalized filtering scenarios, such as spam over instant messaging (spim) detection and Web page blocking (e.g. offensive content).

Acknowledgment

This work is supported by FCT grant PTDC/EIA/64541/2006.

References

[1] E. Turban, R. Sharda, J. Aronson, and D. King, *Business Intelligence, A Managerial Approach*. Prentice-Hall, 2007.

[2] V. Cheng and C. Li, "Personalized Spam Filtering with Semi-supervised Classifier Ensemble," in *IEEE/WIC/ACM International Conference on Web Intelligence*, 2006.

[3] C. Kanich, C. Kreibich, K. Levchenko, B. Enright, G. Voelker, V. Paxson, and S. Savage, "Spamalytics: An Empirical Analysis of Spam Marketing Conversion," in *Computer and Communications Security Conference (CCS'08)*. ACM, 2008, pp. 27–31.

[4] J. Méndez, I. Cid, D. Glez-Peña, M. Rocha, and F. Fdez-Riverola, "A Comparative Impact Study of Attribute Selection Techniques on Naïve Bayes Spam Filters," in *8th Industrial Conference on Data Mining*, Springer, Ed., vol. LNAI 5077, 2008, pp. 213–227.

[5] A. Ramachandran and N. Feamster, "Understanding the Network-Level Behavior of Spammers," in *SIGCOMM'06*, ACM, Ed., 2006, pp. 291–302.

[6] Z. Zhong, L. Ramaswamy, and K. Li, "ALPACAS: A Large-scale Privacy-Aware Collaborative Anti-spam System," in *IEEE INFOCOM*, 2008, pp. 556–564.

[7] M. Chang, W. Yih, and C. Meek, "Partitioned Logistic Regression for Spam Filtering," in *14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 97–105.

[8] S. Hershkop and S. Stolfo, "Combining Email Models for False Positive Reduction," in *11th ACM SIGKDD Int. Conference on Knowledge discovery and data mining*, 2005, pp. 21–24.

[9] A. Gray and M. Haahr, "Personalised, Collaborative Spam Filtering," in *1st Conference on E-Mail and Anti-Spam CEAS*, 2004.

[10] B. Nelson, M. Barreno, F. Chi, A. Joseph, B. Rubinstein, U. Saini, C. Sutton, J. Tygar, and K. Xia, "Exploiting Machine Learning to Subvert Your Spam Filter," in *1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*. ACM Press, 2008, pp. 1–9.

[11] K. Yu, A. Schwaighofer, V. Tresp, W. Ma, and H. Zhang, "Collaborative Ensemble Learning: Combining Collaborative and Content-Based Information Filtering via Hierarchical Bayes," in *19th International Conference on Uncertainty in Artificial Intelligence (UAI)*. ACM, 2003, pp. 353–360.

[12] F. Provost, *Advances in Distributed and Parallel Knowledge Discovery*. MIT Press, 2000, ch. Distributed data mining: Scaling up and beyond.

[13] V. Metsis, I. Androutsopoulos, and G. Paliouras, "Spam Filtering with Naive Bayes – Which Naive Bayes?" in *Third Conference on Email and Anti-Spam (CEAS)*, 2006, pp. 125–134.

[14] R. Beckermann, A. McCallum, and G. Huang, "Automatic categorization of email into folders: benchmark experiments on Enron and SRI corpora," University of Massachusetts Amherst, IR-418, 2004.

[15] R. Segal, "Combining global and personal anti-spam filtering," in *Forth Conference on Email and Anti-Spam (CEAS)*, 2007.

[16] A. Kosmopoulos, G. Paliouras, and I. Androutsopoulos, "Adaptive Spam Filtering Using Only Naive Bayes Text Classifiers," in *CEAS 2008 - Fifth Conference on Email and Anti-Spam*, August 2008.

[17] R. Grossman, M. Hornick, and G. Meyer, "Data Mining Standards Initiatives," *Communications of ACM*, vol. 45, no. 8, pp. 59–61, 2002.

[18] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, pp. 861–874, 2006.

[19] A. Flexer, "Statistical Evaluation of Neural Networks Experiments: Minimum Requirements and Current Practice," in *Proceedings of the 13th European Meeting on Cybernetics and Systems Research*, vol. 2, Vienna, Austria, 1996, pp. 1005–1008.

[20] R Development Core Team, *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-00-3, http://www.R-project.org, (Accessed 26 March 2008).

[21] I. Feinerer, K. Hornik, and D. Meyer, "Text Mining Infrastructure in R," *Journal of Statistical Software*, vol. 25, no. 1-54, 2008.

[22] R. Kewley, M. Embrechts, and C. Breneman, "Data Strip Mining for the Virtual Design of Pharmaceuticals with Neural Networks," *IEEE Trans Neural Networks*, vol. 11, no. 3, pp. 668–679, May 2000.