Semi Supervised Image Spam Hunter: A Regularized Discriminant EM Approach

Yan Gao[1], Ming Yang[2], and Alok Choudhary[1]

[1] Dept. of EECS, Northwestern University,
Evanston, IL, USA
[2] NEC Laboratories America,
Cupertino, CA, USA
{ygao@cs,choudhar@eecs}.northwestern.edu, myang@sv.nec-labs.com

Abstract. Image spam is a new trend in the family of email spams. The new image spams employ a variety of image processing technologies to create random noises. In this paper, we propose a semi-supervised approach, regularized discriminant EM algorithm (RDEM), to detect image spam emails, which leverages small amount of labeled data and large amount of unlabeled data for identifying spams and training a classification model simultaneously. Compared with fully supervised learning algorithms, the semi-supervised learning algorithm is more suitedin adversary classification problems, because the spammers are actively protecting their work by constantly making changes to circumvent the spam detection. It makes the cost too high for fully supervised learning to frequently collect sufficient labeled data for training. Experimental results demonstrate that our approach achieves 91.66% high detection rate with less than 2.96% false positive rate, meanwhile it significantly reduces the labeling cost.

1 Introduction

Spam is e-mail that is both unsolicited by the recipient and sent in nearly identical form to numerous recipients. Research reveals that 96.5% of incoming e-mails received by businesses were spam by June 2008 [1], and spam management costs U.S. businesses more than $70 billion annually [2]. As of 2007, image spam accounted for about 30% of all spam [3]. Image spam has become a more and more deteriorating issue in recent years [4].

Most current content-based spam filtering tools treat conventional email spam detection as a text classification problem, utilizing machine learning algorithms such as neural networks, support vector machine (SVM) and naïve Bayesian classifiers to learn spam characteristics [5–10]. These text-based anti-spam approaches achieved outstanding accuracy and have been widely used. In response, spammers have adopted a number of countermeasures to circumvent these text-based filters. Embedding spam messages into images, usually called “image spam”, is one of the most recent and popular spam construction techniques.

R. Huang et al. (Eds.): ADMA 2009, LNAI 5678, pp. 152–164, 2009.

Fig. 1. Sample spam images

Typically the image spam contains the same types of information advertised in traditional text-based spams, while similar techniques from CAPTCHA (Completely Automated Public Turing Test to Tell Computers and Humans Apart) are applied to manipulate the texts in the image. These techniques include adding speckles and dots in the image background, randomly changing image file names, varying borders, randomly inserting subject lines, and rotating the image slightly. Figure 1 shows some examples. The consequence is an almost infinite number of image-based spams that contain random variants of similar spam content. This kind of spam images is typically attached to or embedded in text with randomly generated good words or content lifted from famous literature. Through this, image spam has successfully bypassed text-based spam filters and presented a new challenge for spam researchers.

In the early stage, there are several organizations and companies working on filtering image-based spam using Optical Character Recognition (OCR) techniques [11, 12]. SpamAssassin (SA) [13] pulls words out of the images and uses the traditional text-based methods to filter spams. This strategy was unavoidably defeated by the appearance of CAPTCHA. Therefore, it is an urgent need to develop a fully automatic image content based spam detection system. Several recent research works are targeting on it, such as the image spam hunter proposed by Gao et al. [14], Dredze et al's fast image spam classifier [15], and near duplicate image spam detection [16, 17]. Most of them leverage supervised machine learning algorithms to build a classifier for filtering spam images [14, 15] by using image-based features.

However, in an adversary classification problem [18] like spam detection, it is not sufficient to just train a classifier once. The reason resides in the fact that any machine learning algorithms are estimating models based on the data statistics, and the assumption is that the statistics of the data used for training are similar to the data statistics in testing. However, spammers are always trying to counteract them by adapting their spamming algorithms to produce image spam emails with feature statistics different from what the anti-spam algorithms have been trained upon. Therefore, the anti-spam algorithms may need to be re-trained from time to time to capture the adversary changes of spam statistics.

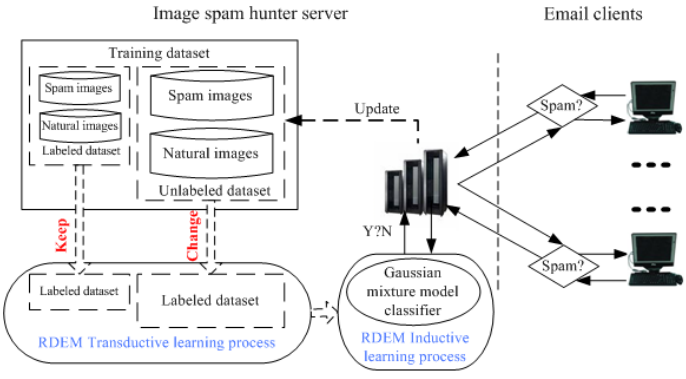

Fig. 2. Prototype system diagram

Furthermore, collecting sufficient labeled spam images for re-training the classifiers is a very labor intensive and costly task. Therefore, it is not desirable, if not possible, to label a large amount of images attached in emails for re-training the classifiers each time, especially when it has to be done very frequently to keep the pace with the spammers. In order to avoid such high cost of labeling large amount of data, a semi-supervised learning scheme [19] is a more efficient choice, where we can leverage small amount of labeled image data and large amount of unlabeled data for training the classifiers.

In this paper, we propose a regularized discriminant EM algorithm (RDEM) to perform semi-supervised spam detection. RDEM improves discriminant EM (DEM) algorithm by leveraging semi-supervised discriminant analysis. In particular, we regularize the cost function of multiple discriminant analysis (MDA) [20] in DEM with a Laplacian graph embedding penalty. Inherited from the DEM, our approach performs both *transductive* learning and inductive learning. We test the proposed approach on an image spam dataset collected from Jan 2006 to Mar 2009, which contains both positive spam images collected from our email server, and negative natural images randomly downloaded from Flickr.com and other websites by performing image search using Microsoft Live Search. Our approach achieves 2.96% false positive rates with 91.66% true positive rates with less than 10% labeled data on average. Comparison results with previous literature demonstrate the advantages of our proposed approach.

2 System Framework of RDEM Image Spam Hunter

In this section, we describe a semi-supervised Image Spam Hunter system prototype to differentiate spam images from normal image attachments. Figure 2 shows the system diagram. We first randomly choose and label a small percentage of spam images as the positive samples and general photos as the negative samples to form the labeled training dataset. The unlabeled training dataset is randomly chosen from the mixed pool of spam images and normal photos. There is no need for clustering the spam images and normal photos into groups in our prototype system, because the Gaussian

mixture model (GMM) in our algorithm is able to deal with the multi-class categorization problem automatically.

The RDEM algorithm, which will be further detailed in Section 3.3, is then applied to the training dataset, which includes both small amount of labeled training data and large amount of unlabeled training data, to build a model for distinguishing the image spams from good emails with image attachments. The unlabeled training data are labeled in this process, which is the *transductive* learning part of the proposed RDEM algorithm. A Gaussian mixture model is induced simultaneously in a discriminative embedding space, which could be further used to classify new data. This is the *inductive* learning part of the RDEM algorithm. Because of this joint transductive and inductive learning process, our proposed semi-supervised image spam hunter is robust to the random variations that exist in current spam images, and easy to adapt to the new changes for image spams in terms of the low labeling cost.

It is worth noting that our semi-supervised spam hunter also fits well as a helpful component running at the beginning of other supervised anti-spam systems to boost the small amount of labeled data. Once enough labeled data is generated through the component, a fully supervised classifier could be further trained for automated spam detection. In a sense, the proposed semi-supervised spam detection scheme could also be functioned as a bootstrap system for a fully supervised spam hunter such as the one proposed by Gao et al. [14].

3 Regularized Discriminant EM Algorithm

We improve the discriminant EM (DEM) [21] algorithm for semi-supervised learning, which introduces a graph Laplacian penalty [22, 23] to the discriminant step of the DEM algorithm. We call it regularized DEM algorithm (RDEM). In the rest of this section, we first introduce the classical unsupervised EM algorithm [24], then present the details of the DEM algorithm [21] and RDEM algorithm, respectively.

3.1 EM Algorithm

EM [24] algorithm is an iterative method to perform maximum likelihood parameter estimation with unobserved latent variables in a probabilistic model. Formally, let $D = \{(x_i, z_i)\}_{i=1}^{N}$where $x_i \in R^n$is the observed data, z_i is the unobserved data, and θ is the parameter vector which characterizes the probabilistic model of D. Denote $Z = \{z_i\}_{i=1}^{N}$, $X = \{X_i\}_{i=1}^{N}$, and the log likelihood function by $L(X, Z, \theta)$. In our formulation, we assume that the data model is a Gaussian mixture model (GMM) of k components, therefore $\theta = \{(\omega_j, \mu_j, \sum_j)\}_{j=1}^{k}$, where ω_j, μ_j and $\sum_j$ are the mixture probability, mean, and covariance matrix of the j -th Gaussian component $G(x|\omega_j, \mu_j, \sum_j)$, respectively. Furthermore, we define $z_i = \{z_{ij}\}_{j=1}^{k}$where $0 \le z_{ij} \le 1$ represents how likely data point $x_i$belongs to the j-th Gaussian component. Let θ^{t-1} be the estimated parameter at the iteration $t-1$ of the EM algorithm, at iteration t, the EM algorithm runs the following two steps to estimate the Gaussian mixture model:

E-Step: Calculate the expected value of $L(X,Z,\theta)$ w.r.t. $p(z|x,\theta^{t-1})$, given the current θ^{t-1}, i.e., $Q(\theta|\theta^{t-1}) = E(L(X,Z,\theta)|x,\theta^{t-1})$. We have

$$z_{ij}^t = \frac{\omega_j^{t-1} G\left(x_i \middle| \omega_j^{t-1}, \mu_j^{t-1}, \sum_j^{t-1}\right)}{\sum_k \omega_k^{t-1} G(x_i|\omega_k^{t-1}, \mu_k^{t-1}, \sum_k^{t-1})}, \tag{1}$$

$$\begin{aligned} Q(\theta|\theta^{t-1}) = \sum_{i=1}^{N}\sum_{j=1}^{k} z_{ij}^t [\log \omega_j^{t-1} \\ -\frac{1}{2}\log\left|\sum_j^{t-1}\right| - \frac{1}{2}(x_i-\mu_j)^T \sum\nolimits_j^{t-1^{-1}} (x_i-\mu_j) - \frac{n}{2}\log 2\pi]. \end{aligned} \tag{2}$$

M-Step: Find the parameter θ^t such that $\theta^t = \arg\max_\theta Q(\theta|\theta^{t-1})$. We have

$$\omega_j^t = \frac{1}{n}\sum_i z_{ij}^t, \mu_j^t = \frac{\sum_i z_{ij}^t x_i}{\sum_i z_{ij}^t}, \tag{3}$$

$$\sum_j^t = \frac{\sum_i z_{ij}^t (x_i-\mu_j^t)(x_i-\mu_j^t)^T}{\sum_i z_{ij}^t}. \tag{4}$$

Estimating GMM using EM is a popular approach for unsupervised clustering of data. The EM iterations are guaranteed to find a local optimal estimation.

3.2 Discriminant EM Algorithm

Discriminant EM [21] (DEM) is a semi-supervised extension of the original EM algorithm. It assumes that the data can be categorized into c different classes, and the structure of the data can be captured by a GMM with c components in a $c-1$ dimensional discriminant embedding space. Let $D = \{(x_i, l_i)\}_{i=1}^{N_1}$ be the set of labeled data, where $l_i \in \{1,2,\cdots,c\}$ is the label of the data $x_i \in R^n$. Let $U = \{(u_i, z_i)\}_{i=1}^{N_2}$ be the set of unlabeled data, where $z_i = \{z_{ij}\}_{j=1}^{c}$ is the unknown soft labels of $u_i \epsilon R^n$. Moreover, let $\widetilde{x_i}$ and $\widetilde{u_i}$ be the projection of x_i and u_i in the $c-1$ dimensional discriminant embedding represented by an n × (c − 1) projection matrix W, i.e., $\widetilde{x_i} = W^T x_i$ and $\widetilde{u_i} = W^T u_i$. Also let $\left(\widetilde{\omega_i}, \widetilde{\mu_i}, \widetilde{\sum_i}\right)$ be the parameters of the i-th Gaussian component $G\left(x\middle|\widetilde{\omega_j}, \widetilde{\mu_j}, \widetilde{\sum_j}\right)$ of the GMM in the embedding space. The DEM algorithm is composed of the following three steps:

E-Step: Estimate the probabilities of the class labels for each unlabeled data u_i, i.e.,

$$z_{ij}^t = \frac{\widetilde{\omega}_j^{t-1} G\left(\widetilde{u}_i^{t-1} \middle| \widetilde{\omega}_j^{t-1}, \widetilde{\mu}_j^{t-1}, \widetilde{\sum}_j^{t-1}\right)}{\sum_k \widetilde{\omega}_k^{t-1} G\left(\widetilde{u}_i^{t-1} \middle| \widetilde{\omega}_k^{t-1}, \widetilde{\mu}_k^{t-1}, \widetilde{\sum}_k^{t-1}\right)}. \tag{5}$$

D-Step: Perform multiple discriminant analysis [20] based on the labeled data D and soft labeled data U, by solving the following optimization problem to identify the optimal embedding W^t, i.e.,

$$w^t = arg\max_w \frac{w^T S_b w}{w^T S_w w}, \tag{6}$$

where

$$S_b = \sum_{j=1}^{C} (m - m_j)(m - m_j)^T, \tag{7}$$

$$S_w = \sum_{i=1}^{N_1} (x_i - m_{l_i})(x_i - m_{l_i})^T + \sum_{i=1}^{N_2}\sum_{j=1}^{C} z_{ij}\,(u_i - m_j)(u_i - m_j)^T, \tag{8}$$

$$m = \frac{1}{N_1 + N_2}\left(\sum_{i=1}^{N_1} x_i + \sum_{i=1}^{N_2} u_i\right), \tag{9}$$

$$m_j = \frac{1}{\sum_{i=1}^{N_1}\delta(l_i, j) + \sum_{i=1}^{N_2} z_{ij}}\left(\sum_{i=1}^{N_1}\delta(l_i, j)x_i + \sum_{i=1}^{N_2} z_{ij}u_i\right), \tag{10}$$

and $\delta(l_i, j)$ is the Dirac delta function which takes value one when $l_i$equals j and zero otherwise. $W^t$is composed of the eigen vectors corresponding to the largest $C - 1$ eigen values of the generalize eigen system $S_b w = \lambda S_w w$. Then both the labeled and unlabeled data are projected into the embedding, i.e.,

$$\tilde{x}_i^t = {W^t}^T x_i, \tilde{u}_i^t = {W^t}^T u_i. \tag{11}$$

M-Step: Estimate the optimal parameters of the GMM in the embedding space, i.e.,

$$\tilde{\omega}_j^t = \frac{1}{N_1 + N_2}\left(\sum_{i=1}^{N_1}\delta(l_i, j) + \sum_{i=1}^{N_2} z_{ij}^t\right), \tag{12}$$

$$\tilde{\mu}_j^t = \frac{\sum_{i=1}^{N_1}\delta(l_i, j)\,\tilde{x}_i^t + \sum_{i=1}^{N_2} z_{ij}^t\,\tilde{u}_i^t}{\sum_{i=1}^{N_1}\delta(l_i, j) + \sum_{i=1}^{N_2} z_{ij}^t}, \tag{13}$$

$$\sum_j^t = \frac{\sum_{i=1}^{N_1}\delta(l_i, j)\,(\tilde{x}_i^t - \mu_j^t)(\tilde{x}_i^t - \mu_j^t)^T + \sum_{i=1}^{N_2} z_{ij}^t\,(\tilde{u}_i^t - \mu_j^t)(\tilde{u}_i^t - \mu_j^t)^T}{\sum_{i=1}^{N_1}\delta(l_i, j) + \sum_{i=1}^{N_2} z_{ij}^t}. \tag{14}$$

These three steps are iterated until convergence. As we have already discussed, although DEM itself is a semi-supervised algorithm, the D-step is a purely supervised step. This is not desirable because it fully trusts the labels estimated from the E-step. We proceed to replace it with a semi-supervised discriminant analysis algorithm.

3.3 Regularized Discriminant EM Algorithm

Cai et al. [25] and Yang et al. [26] propose a semi-supervised discriminant analysis algorithm to leverage both labeled and unlabeled data to identify a discriminant embedding for classification. Following the common principle of learning from

unlabeled data, which is to respect the structure of the data, semi-supervised discriminant analysis introduces a graph Lapacian regularization term into multiple discriminant analysis, based on the regularized discriminant analysis framework proposed by Friedman [27]. The intuition of applying the graph Laplacian regularization is that in a classification problem, data points which are close to one another are more likely to be categorized in the same class. More formally, for the unlabeled data set U, let $U = [u_1, u_2, \cdots, u_{N_2}]$ be the $n \times N_2$data matrix, we define

$$s_{kl} = \begin{cases} 1, & u_k \in N_p(u_l) \| u_l \in N_p(u_k) \\ 0, & otherwise \end{cases}, \tag{15}$$

where $N_p(u)$ indicates the p-nearest neighbors of the data point u. Let $S = [s_{kl}]$, and $D = diag[d_{kk}]$ where $d_{kk} = \sum_{l=1}^{N_2} s_{kl}$. Both are $N_2 \times N_2$ matrices. S defines a p-nearest neighbor graph. Following previous work on spectral cluster [22, 23], the graph Laplacian is naturally defined as

$$J(w) = \sum_{k=1}^{N_2} \sum_{l=1}^{N_2} s_{kl}(w^T u_k - w^T u_l)^2 \tag{16}$$

$$= 2\sum_{k=1}^{N_2} w^T u_k d_{kk} u_k^T w - 2\sum_{k=1}^{N_2}\sum_{l=1}^{N_2} w^T u_k s_{kl} u_l^T w \tag{17}$$

$$= 2w^T U(D - S)U^T w = 2w^T ULU^T w, \tag{18}$$

there L is the Laplacian matrix [22, 23]. It is clear minimizing $J(w)$ with respect to w would result in that data close to one another would be also close to one another in the embedding space. Following the regularized discriminant analysis [27], we introduce this graph Laplacian [22] regularization term into the multiple discriminant analysis cost function (i.e., Equation 6), i.e.,

$$w^t = \arg\max_w \frac{w^T S_b w}{w^T S_w w + \beta w^T ULU^T w}, \tag{19}$$

where $S_b$and $S_w$are defined in Equation 7-10, and β is a control parameter to balance between the supervised term and unsupervised term, respectively. In the D-Step, we shall replace Equation 6 with Equation 19 and perform a semi-supervised discriminant analysis. We denote $S'_w = S_w + \beta ULU^T$. Because of the graph Laplacian regularization term, W□is composed of the C eigenvectors corresponding to the C largest eigen values in the generalized eigen system $S_b w = \lambda S'_w w$. Keep the other two steps unchanged in the DEM algorithm discussed in the previous subsection, we propose an improved DEM algorithm. Named after the regularized discriminant analysis, we call it regularized DEM algorithm (RDEM). One thing we should notice that there is a small difference between Equation 19 and the formulation proposed by Cai et al. [25], because our formulation also takes the soft labels of the unlabeled data into consideration. It is clear that when β= 0, Equation 19 degenerates to Equation 6. In a sense, DEM is a special case of the regularized DEM algorithm.

Inherited from the DEM, our proposed approach performs both *transductive* learning and *inductive* learning. It performs transductive learning since the unlabeled

data will be labeled after the training process by the maximum a posteriori estimation. Meanwhile, the induced GMM model in the discriminative embedding space can be used straightforwardly for classifying new data samples.

4 Image Features

We adopt an effective set of 23 image statistics [28–30] integrating color, texture, shape, and appearance properties for image spam detection.

Color Statistics: We build a 10^3 dimensional color histogram in the joint RGB space, i.e., each color band is quantized into 10 different levels. The entropy of this joint RGB histogram is the first statistics we adopted. We also build an individual 100 dimensional histogram for each of the RGB band, 5 statistics are calculated from each of these three histograms, including the discreteness, mean, variance, skewness, and kurtosis. The discreteness is defined as the summation of all the absolute differences between any two consecutive bins. The other four are all standard statistics. Hence we adopt 16 color statistics in total.

Texture Statistics: We employ the local binary pattern (LBP) [31] to analyze the texture statistics. A 59 dimensional texture histogram is extracted. It is composed of 58 bins for all the different uniform local binary patterns, i.e., those with at most two $0\sim1$ transitions in the 8-bit stream, plus an additional bin which accounts for all other non-uniform local binary patterns. The entropy of the LBP histogram is adopted as one feature. This adds in 1 texture statistics.

Shape Statistics: To account for the shape information of the visual objects, we build a $40\times8 = 320$ dimensional magnitude-orientation histogram for the image gradient. The difference between the energies in the lower frequency band and the higher frequency band are used as 1 feature. The entropy of the histogram is another feature. We further run a Canny edge detector [32] and the total number of edges and the average length of the edges are adopted as two statistics. These produce 4 shape statistics in total.

Appearance Statistics: We build the spatial correlogram [33] of the grey level pixels within a 1-neighborhood. The average skewness of the histograms formed from each slice of the correlogram is utilized as one feature. Another feature is the average variance ratio of all slices, where the variance ratio is defined as the ratio between the variance of the slice and the radius of the symmetric range over the mean of the slice that accounts for 60% of the total counts of the slice. These add up to 2 appearance statistics.

5 Experiments

Three quantities, i.e., *recognition accuracy, true positive rate, and false positive rate*, are adopted to compare the different approaches. The recognition accuracy stands for the overall classification accuracy of both spam and non-spam images. The true

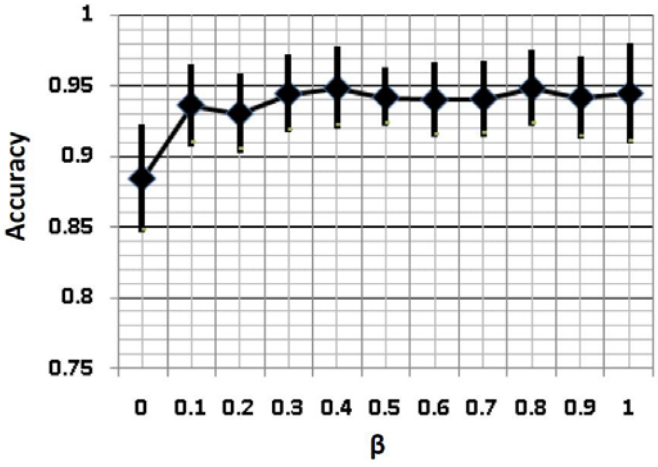

Fig. 3. The recognition accuracy of the RDEM algorithm with different setting of β. Each bar presents the average recognition accuracy and the standard deviation over 20 random label/unlabel splits. Note $\beta = 0$ is equivalent to run the original DEM algorithm.

positive rate represents the portion of the spam images being classified as spams, while the false positive rate indicates the portion of the non-spam images being classified as spam. We have to remark that most often a spam detection system would prefer to work with low false positive rate, and very few missing detections of spams are acceptable.

5.1 Data Collection

We collected two sets of images to evaluate our semi-supervised spam hunter system: normal images and spam images. We collected 1190 spam images from real spam emails received by 10 graduate students in our department between Jan 2006 and Feb 2009. These images were extracted from the original spam emails and converted to jpeg format from bmp, gif and png formats. Since we anticipate the statistics of normal images will be similar to those photo images found in social networking sites and image search results from popular search engines, we collected 1760 normal images by either randomly downloading images from Flickr.com or fetching the images from other websites from the search results on Microsoft Live Image Search (http://www.live.com/?scope=images).

5.2 Comparison to DEM

To simulate the real application scenarios, we randomly sample 10% (small portion) of the images from the data we collected to represent the labeled data, and the rest are regarded as the unlabeled data. We call one such random sample as 1 split. Since β in Equation 19 controls the impact of the graph Laplacian regularization in the RDEM algorithm, we first explore the impact of it. The recognition accuracy is calculated based on the maximum a posteriori label estimate.

Figure 3 presents the recognition accuracy with different settings of β. We test 10 different settings to vary β from 0.1 to 1.0 with a stride of 0.1. Each black marker presents the average recognition accuracy over 20 random splits. The bar overlayed on each marker presents standard deviation of the recognition accuracy over 20 splits. As we can clearly observe, the recognition accuracy of RDEM is not that sensitive to the setting of β. For $\beta \neq 0$, the average detection accuracies are all above 90%. Indeed, the marker corresponding to $\beta = 0$ is exactly the recognition accuracy of the DEM algorithm. It shows that RDEM is superior to DEM with all the different settings of β.

In the experiments, we use 3 Gaussian components to model the spam images, and another 3 Gaussian components to model ordinary images. Since β = 0.8 presents the best recognition accuracy of 94.84% for RDEM, we use it for all the other comparisons. We further compare RDEM and DEM in terms of the average true positive rate and false positive rate in Table 1. As we can see the RDEM outperforms the original DEM algorithm significantly, i.e., it achieves both higher true positive rate and lower false positive rate than the DEM. It is also worth noting that the detection results of the RDEM algorithm are also more consistent, i.e., the standard deviation of the detection rates from it are smaller. This manifests that RDEM is statistically more stable.

Table 1. Comparison of RDEM with DEM algorithm.

Method	Ave. True Positive	Ave. False Positive
DEM	85.38%±5.20%	9.69%±7.00%
RDEM($\beta = 0.8$)	91.66%±2.33%	2.96%±1.45%

5.3 Comparison to Supervised Learning Methods

Table 2 shows the comparison of the accuracy of our RDEM method against two popular supervised learning methods, the Boosting tree [34] and SVM [35], with different amount of labeled data. The Boosting tree [34] was leveraged by Gao et al. [14] to detect the spam images, and SVM [35] has demonstrated to be the optimal classifier in many applications. We can observe that RDEM demonstrates consistent performance gain over the Boosting tree and SVM with either 1%, 5% or 10% of labeled data. For example, RDEM can still achieve 88.40% true positive rate with a false positive rate of 5.61%, given the labeled data only accounts for 1% of the total data in our data collection (i.e., 12 spam images and 18 normal photos).

As we also observe, when the number of labeled data is small, the variances of the true positive rates of both the Boosting tree and the SVM are much higher than that of the RDEM algorithm. This is quite understandable since for strong supervised learning algorithms such as boosting tree and SVM, lacking of labeled training data would make the learning process very brittle and unstable. Hence they show very high variances. Our preliminary results show a very good cost performance of RDEM. The small number of labeled data in the training stage is extremely valuable for the real client-side email spam detection system, as it avoids annoying the end users by the tedious task of labeling a lot of image spams, and provides the spam detection system a good start.

Table 2. Comparison of the RDEM against the Boosting tree and SVM methods with different amounts of labeled data

Method	Ave. True Positive Rate		
	1.0%	5.0%	10.0%
RDEM ($\beta = 0.8$)	88.40%±3.19%	90.89%±3.57%	91.66%±2.33%
Boosting tree	67.09%±38.92%	72.99%±36.64%	86.87%±5.71%
SVM	19.39%±33.74%	51.59%±25.21%	68.51%±17.48%
Method	Ave. False Positive Rate		
	1.0%	5.0%	10.0%
RDEM ($\beta = 0.8$)	5.61%±4.09%	3.61%±2.82%	2.96%±1.45%
Boosting tree	4.85%±4.15%	5.06%±3.94%	3.44%±1.94%
SVM	12.65%±29.72%	9.80%±13.48%	9.25%±9.60%

6 Conclusion and Future Work

We proposed a semi-supervised system prototype based on a regularized discriminant EM algorithm to detect the spam images attached in emails. The proposed method employs a small amount of labeled data and extracts efficient image features to perform both transductive and inductive learning to detect the spam images, and achieves promising preliminary results. Future research will be focusing on further improving the computational efficiency of the RDEM algorithm, and exploring more discriminative image features.

References

1. Sophos Plc: http://www.sophos.com/pressoffice/news/articles/2008/07/dirtydozjul08.html
2. Necleus Research: http://nucleusresearch.com/research/notes-and-reports/spamthe-repeat-offender/
3. McAfee: http://www.avertlabs.com/research/blog/index.php/2007/05/25/arespammers-giving-up-on-image-spam/
4. Hayati, P., Potdar, V.: Evaluation of spam detection and prevention frameworks for email and image spam a state of art. In: Proc. Conf. on Information Integration and Web-based Application and Services, Linz, Austria (November 2008)
5. Sahami, M., Dumais, S., Heckerman, D., Horvitz, E.: A bayesian approach to filtering junk e-mail. In: Proc. AAAI Workshop on Learning for Text Categorization, Madison, Wisconsin (July 1998)
6. Drucker, H., Wu, D., Vapnik, V.N.: Support vector machines for spam categorization. IEEE Transactions on Neural Networks 10, 1048–1054 (1999)
7. Carreras, X., Salgado, J.G.: Boosting trees for anti-spam email filtering. In: Proc. the 4th International Conference on Recent Advances in Natural Language Processing, Tzigov Chark, BG, pp. 58–64 (2001)
8. Boykin, P.O., Roychowdhury, V.P.: Leveraging social networks to fight spam. Computer 38(4), 61–68 (2005)

9. Blosser, J., Josephsen, D.: Scalable centralized bayesian spam mitigation withbogofilter. In: USENIX LISA (2004)
10. Li, K., Zhong, Z.: Fast statistical spam filter by approximate classifications. In: ACM SIGMETRICS, pp. 347–358 (2006)
11. Fumera, G., Pillai, I., Rolir, F.: Spam filtering based on the analysis of text information embedded into images. Journal of Machine Learning Research 6, 2699–2720 (2006)
12. Biggio, B., Fumera, G., Pillai, I., Roli, F.: Image spam filtering using visual information. In: ICIAP (2007)
13. SpamAssassin: `http://spamassassin.apache.org`
14. Gao, Y., Yang, M., Zhao, X., Pardo, B., Wu, Y., Pappas, T., Choudhary, A.: Imagespam hunter. In: Proc. of the 33rd IEEE International Conference on Acoustics, Speech, and Signal Processing, Las Vegas, NV, USA (April 2008)
15. Dredze, M., Gevaryahu, R., Elias-Bachrach, A.: Learning fast classifiers for imagespam. In: Proc. the 4th Conference on Email and Anti-Spam (CEAS), California, USA (August 2007)
16. Mehta, B., Nangia, S., Gupta, M., Nejdl, W.: Detecting image spam using visual features and near duplicate detection. In: Proc. the 17th International World Wide Web Conference, Beijing, China (April 2008)
17. Wang, Z., Josephson, W., Lv, Q., Charikar, M., Li, K.: Filtering image spam with near-duplicate detection. In: Proc. the 4th Conference on Email and Anti-Spam (CEAS), California, USA (August 2007)
18. Dalvi, N., Domingos, P., Mausam, S.S., Verma, D.: Adversarial classification. In: Tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 99–108 (2004)
19. Zhu, X.: Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison (2005)
20. Fisher, R.A.: The use of multiple measurements in taxonomic problems. Annals of Eugenics 7, 179–188 (1936)
21. Wu, Y., Tian, Q., Huang, T.S.: Discriminant-em algorithm with application to image retrieval. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Hilton Head Island, SC, USA, June 2000, vol. I (2000)
22. He, X., Niyogi, P.: Locality preserving projections. In: Thrun, S., Saul, L., Schölkopf, B. (eds.) Advances in Neural Information Processing Systems 16. MIT Press, Cambridge (2004)
23. He, X., Yan, S., Hu, Y., Niyogi, P., Zhang, H.: Face recognition using laplacianfaces. IEEE Transaction on Pattern Analysis and Machine Intelligence 27(3), 328–340 (2005)
24. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society. Series B (Methodological) 39(1), 1–38 (1977)
25. Cai, D., He, X., Han, J.: Semi-supervised discriminant analysis. In: Proc. the 11th IEEE International Conference on Computer Vision (ICCV), Rio de Janeiro, Brazil (October 2007)
26. Yang, J., Yan, S., Huang, T.: Ubiquitously supervised subspace learning. IEEE Transactions on Image Processing 18(2), 241–249 (2009)
27. Friedman, J.H.: Regularized discriminant analysis. Journal of the American Statistical Association 84(405), 165–175 (1989)
28. Ng, T.T., Chang, S.F.: Classifying photographic and photorealistic computer graphic images using natural image statistics. Technical report, Columbia University (October 2004)

29. Ng, T.T., Chang, S.F., Hsu, Y.F., Xie, L., Tsui, M.P.: Physics-motivated features for distinguishing photographic images and computer graphics. In: ACM Multimedia, Singapore (November 2005)
30. Ng, T.T., Chang, S.F., Tsui, M.P.: Lessons learned from online classification of photo-realistic computer graphics and photographs. In: IEEE Workshop on Signal Processing Applications for Public Security and Forensics (SAFE) (April 2007)
31. Mäenpä, T.: The local binary pattern approach to texture analysis extensions and applications. Ph.D thesis, Infotech Oulu, University of Oulu, Oulu, Finland (August 2003)
32. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. 8(6), 679–698 (1986)
33. Huang, J., Kumar, S.R., Mitra, M., Zhu, W.J., Zabih, R.: Image indexing using color correlograms. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, Los Alamitos (1997)
34. Tu, Z.: Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering. In: Tenth IEEE International Conference on Computer Vision (2005)
35. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20(3), 273–297 (1995)