A Comprehensive Approach to Image Spam Detection: From Server to Client Solution

Yan Gao, Alok Choudhary, *Fellow, IEEE*, and Gang Hua, *Member, IEEE*

***Abstract*—Image spam is a type of e-mail spam that embeds spam text content into graphical images to bypass traditional text-based e-mail spam filters. To effectively detect image spam, it is desirable to leverage image content analysis technologies. However, most previous works of image spam detection focus on filtering the image spam on the client side. We propose a more desirable comprehensive solution which embraces both server-side filtering and client-side detection to effectively mitigate image spam. On the server side, we present a nonnegative sparsity induced similarity measure for cluster analysis of spam images to filter the attack activities of spammers and fast trace back the spam sources. On the client side, we employ the principle of active learning where the learner guides the users to label as few images as possible while maximizing the classification accuracy. The server-side filtering identifies large image clusters as suspicious spam sources and further analysis can be performed to identify the real sources and block them from the beginning. For those spam images which survived the server-side filter, our active learner on the client side will further guide the users to interactively and efficiently filter them out. Our experiments on an image spam data-set collected from the e-mail server of our department demonstrate the efficacy of the proposed comprehensive solution.**

***Index Terms*—Active learning, clustering, image recognition, image spam, spam filtering, sparse representation.**

I. Introduction

GLOBAL spam volume increased very fast over the past five years. E-mail spam accounted for 96.5% of incoming e-mails received in businesses by June 2008 [1], and cost more than $70 billion in management expenses for the U.S. Government annually. The success of text document classification techniques on e-mail spam detection [2]–[4] has driven spammers to explore new variations of spam e-mails, among which image spam e-mail has become a new popular weapon, which accounts for approximately 30% of all e-mail spams, as reported by McAfee [5] in 2007.

Manuscript received January 11, 2010; revised July 22, 2010; accepted September 07, 2010. Date of publication September 27, 2010; date of current version November 17, 2010. This work was supported in part by the National Science Foundation (NSF) under Grant IIS-0905205, Grant OCI-0956311, Grant CCF-0938000, Grant CCF-0621443, Grant OCI-0724599, Grant CCF-0833131, and Grant CNS-0830927, and in part by the DOE under Grant DE-FC02-07ER25808/A000, Grant DE-FG02-08ER25848/A001, and Grant DE-SC0001283. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Darko Kirovski.

Y. Gao and A. Choudhary are with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60208 USA (e-mail: yangao2009@u.northwestern.edu; choudhar@eecs.northwestern.edu).

G. Hua is with IBM Research T. J. Watson Center, Hawthorne, NY 10532 USA (e-mail: ganghua@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2010.2080267

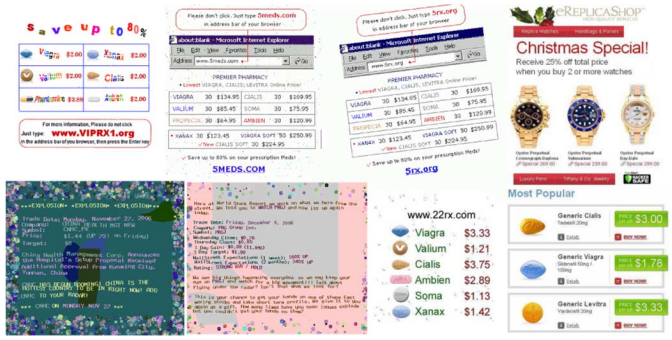


Fig. 1. Example of spam images: spammers usually generate a set of varieties from a single image source using image processing and manipulation algorithms.

Image spam stands for those spam e-mails which embed the spam text messages into the graphical content of image attachments. Unlike image attachment such as company logos, the embedded text messages are intended for massive distribution, such as advertising cheap VIGRA. Since most e-mail clients will render a graphics image automatically, image spam can successfully deliver the intended message to the end users.

To avoid being traced by an exact Hash signature such as MD5 and to avoid the embedded text message to be recognized robustly by an optical character recognition (OCR) system, when the spammers render the textual content into an image, they impose various image processing and manipulation techniques on the image, such as those tricks utilized in CAPTCHA (Completely Automated Public Turing Test to Tell Computers and Humans Apart). These different tricks include, but are by no means limited to, adding speckles and dots in the image background, varying borders, randomly inserting subject lines, and rotating the images slightly and so on. Fig. 1 shows some examples of image spams.

Although a large amount of end users receive different image spams, these images are substantially visual variations from a small number of spam image sources. By appending texts containing randomly generated words based on normal natural language statistics in the text body of the e-mail or subject lines with the spam images, the image spam can successfully bypass text-based spam filters and make it perform practically no function. Therefore, we have to leverage image content analysis or computer vision algorithms to visually recognize these spam images.

Some early work such as SpamAssassin [6] has tried to pull out the embedded texts in the spam images by using optical character recognition (OCR), and then applying text-based spam filtering techniques. However, highly accurate OCR on spam im-

ages may be by itself a more difficult problem than spam image classification, especially when the spammers are performing the aforementioned adversarial manipulation of the image content. This is probably the reason why many recent works have been focusing on directly classifying e-mail image attachments as either spam or nonspam, such as the different image spam hunters [7], [8], fast image spam classifiers [9], and near duplicate image spam detection [10]. A supervised or semisupervised learning machinery is usually leveraged in these image spam classifiers.

Although supervised learning algorithms have achieved good accuracy for the task of image spam detection, getting sufficient labeled images for robust training is always expensive, especially for the adversarial classification problem in which retraining the model needs to be done quite often. One possible solution might be utilizing semisupervised learning to save some labeling cost, such as the semisupervised image spam hunter proposed by Gao *et al.* [8]. However, neither a fully supervised classifier nor semisupervised classifier can guarantee 100% accuracy, so human intervention is still needed from time to time to avoid erroneously discarding valuable e-mail messages with image attachments.

Because it is unavoidable to have human interaction, in order to further save the labeling cost, we propose to leverage active learning [11]–[15] for guiding the users to label as few spam images as possible while maximizing high classification accuracy for the users. This way, we can drastically reduce the labeling cost by identifying the most informative examples for users to label. This has largely motivated our work on employing active learning for image spam filtering [16].

However, all these direct classification algorithms, whether it is a supervised learner [7], [9], a semisupervised learner [8], or an active learner [9], largely focus on classifying each individual image attachment on the client side. It lacks, however, more global analysis of the corpus of image attachments as a whole. It is obvious that such holistic analysis of the image spam corpus can only be carried out on the e-mail server. More specifically, unsupervised cluster analysis of the image corpus on the server side may provide more information on the sources of the spam images. For example, if a majority of e-mail users of an e-mail server received image attachments from the same cluster, then it is highly likely that they are spam images. Further analysis can then be performed to identify the source senders and block them from the server side directly in the future.

Nevertheless, to effectively cluster images, it is essential to have a good visual similarity measure for different images. Previous work has designed different image signatures from diverse image features to define either $L1$ or $L2$ norm, weighted, or unweighted, in the feature space as the similarity measures [10], [17]. However, they are not able to adapt to the manifold structure of the image features, as pointed out by Cheng *et al.* [18].

We propose a *nonnegative sparsity induced similarity measure* and apply it for the task of cluster analysis of spam images. The basic proposition we make is that an image should be able to be effectively reconstructed by a small number of other images from the same cluster. This is especially true for spam images because many spam images are generated from a limited number of source images. We design a quadratic program to calculate such nonnegative sparse representation and a similarity measure is further derived from such a representation.

It is easy to understand that server-side spam filtering is largely complementary to client-side spam filtering. Therefore, we proposed a comprehensive approach, which combines both server-side cluster analysis of spam images and client-side active learning spam classification. On the server-side cluster analysis, we employ the nonnegative sparsity induced similarity measure discussed above and use a spectrum clustering algorithm proposed by Song *et al.* [19]. Those large image clusters are highly suspicious ones which can be further analyzed by the administrator. If a common source is identified, the spam source will be blocked on the server side from the beginning. For spam images which survived the server-side filtering, we further utilize our active learning image spam hunter to effectively deal with them in an interactive and efficient way. We present and compare two active learning classifiers in our system. One is based on an SVM and the other is based on a Gaussian process classifier, respectively.

The remainder of the paper is organized as follows. In Section II, we present the whole system design of the proposed comprehensive approach to image spam filtering, including both server-side and client-side components. The algorithmic design of our server-side image spam detection system is presented in Section III. We present the detailed algorithms for client-side image spam filtering system in Section IV. We further introduce the dataset as well as evaluation criteria we used for experiments in Section V. Extensive experimental evaluations of the system are presented and discussed in Section VI. We conclude the paper and discuss possible future work in Section VII.

II. From Server to Client: A Comprehensive System Design

Fig. 2 presents the overall system flowchart of the proposed comprehensive solution. Given a batch of image attachments from an e-mail server, our system would first extract invariant visual features to represent each image. The visual representation we finally adopted and some analysis and visualization of the goodness of the adopted visual feature representation is discussed in detail in Section V-B. After that, for server-side filter, cluster analysis is performed on the image set. Since spam images are usually sent in bulk, larger clusters are more suspicious to be spam image groups and can be further analyzed to identify the sources. For example, the adminstrator can further identify the spam sources so that we can block them from the server side in a very early stage. We shall remark here that this is hard to achieve if we only do client-side spam filtering. In this paper, we solely focus on how to identify these suspicious larger clusters. We will present our detailed algorithmic design of our cluster analysis system in Section III.

Those image attachments which can pass the cluster analysis on the server side will be sent to the end users. Then on the client side, we further design an active learning classification system where the learner will guide the users to efficiently and interactively classify the spam images which have survived

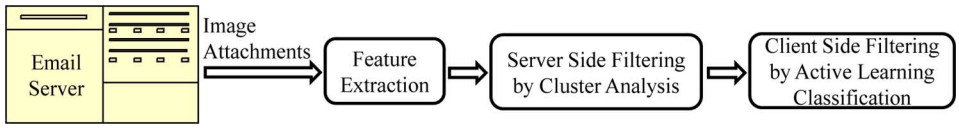


Fig. 2. Design flowchart of the proposed comprehensive solution to image spam filtering.

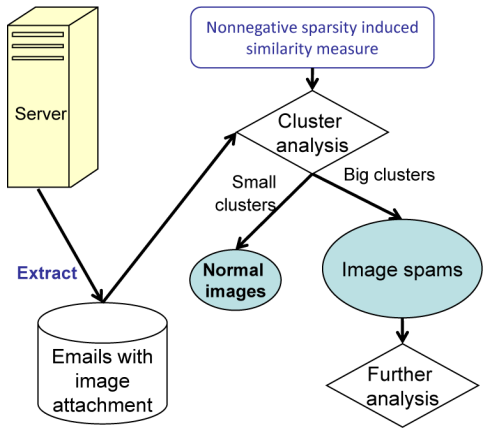


Fig. 3. System flowchart of a server-side image spam detection system by cluster analysis.

the server-side filtering. Detailed algorithmic design of the proposed active learning classification system will be presented in Section IV, in which we also explore and compare the different learning algorithms to design the active learning classifier. To the best of our knowledge, we are the first to have proposed to use active learning for the task of image spam detection.

III. Server-Side Image Spam Filtering

In this section, we will present our server-side image spam filtering subsystem. We first present the flowchart of the subsystem in Section III-A, followed by the details of a nonnegative sparsity induced similarity measure for cluster analysis of spam images.

A. System Flowchart

Fig. 3 presents the flowchart of a server-side image spam detection system by cluster analysis. Given a set of image attachments extracted from the e-mail server, we cluster them by leveraging a nonnegative sparsity induced similarity measure, which we shall discuss in more detail in Section III-B. With server-side cluster analysis and source blocking, we hope that the spam e-mails received by end users are minimized. Those smaller clusters are most often normal images so that they will be passed to the client users in the end. There may be false negatives, but they are in small bulk and less annoying to the end users. Moreover, the client-side spam image filters would be able to further capture them.

B. Nonnegative Sparsity Induced Similarity Measure for Clustering

Any cluster analysis relies on a good similarity measure. We proceed to present our nonnegative sparsity induced similarity measure in this section. Assume $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$ is the feature vectors of the N images obtained from a batch of e-mails in an e-mail server, where $\forall i, \mathbf{x}_i \in \mathbf{R}^n$. Our nonnegative sparsity induced similarity is based on a basic assumption. That is, any data sample or feature vector in the corpus can be well represented by the nonnegative linear combination of a small number of samples from the same cluster. Nevertheless, for $\mathbf{x}_i$, we do not know beforehand which samples are in the same cluster, not to mention which small set of samples would reconstruct it well.

To successfully identify the potential small sample set to nonnegatively linear reconstruct $\mathbf{x}_i$, let $\mathbf{X}_i = [\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_N]$, we propose to solve the following optimization problems:

$$\min_{\mathbf{c}} \quad \frac{1}{2}\|\mathbf{x}_i - \mathbf{X}_i \cdot \mathbf{c}\|^2 + \frac{\beta}{2}\|\mathbf{c}\|^2 + \lambda \sum_{j=1}^{n} c_j \tag{1}$$

$$\text{s.t.} \quad \forall j = 1 \ldots N, c_i \geq 0 \tag{2}$$

where $\mathbf{c} = [c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_N]^T$ are the reconstruction coefficients, and β is a small constant to weight the ridge regression cost to penalize $\mathbf{c}$ with large L2 norm.[1] Since we constrain each c_i to be nonnegative, $f(\mathbf{c}) = \sum_{j=1}^{n} c_j$ is equivalent to an L1 norm Lasso penalty [20]. Therefore, solving the above constrained optimization problem would naturally result in $\mathbf{c}$ to be a sparse vector, i.e., a vector with only a small number of nonzero entries. λ is the control parameter of the Lasso penalty, which directly determines how sparse $\mathbf{c}$ will be.

After easy mathematic derivation, it is straightforward to observe that the above formulation (1) can be rearranged as

$$\min_{\mathbf{c}} \quad \mathbf{c}^T(\mathbf{X}_i^T\mathbf{X}_i + \beta\mathbf{I})\mathbf{c} + (\lambda\mathbf{1} - \mathbf{x}_i^T\mathbf{X}_i)^T\mathbf{c} \tag{3}$$

$$\text{s.t.} \quad \forall j = 1 \ldots N, c_i \geq 0 \tag{4}$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{1}$ is a vector with all elements being 1. This is a standard quadratic program with linear constraints and can be solved by a standard active set method. We employ the MINQ [21] Matlab library in our implementation to solve it. Notice that the difference of our formulation compared with those of Benaroya [22] is the additional ridge regression term, which is to regularize the solution of linear regression to be more stable.

Naturally, after we have identified the sparse vector $\mathbf{c}$, we define the similarity of $\mathbf{x}_i$ to all the other data samples to be

$$w_{ij} = \frac{c_j}{\sum_{k=1, k\neq i}^{N} c_k}. \tag{5}$$

Since the w_{ij} induced above may not be symmetric, i.e., $w_{ij} \neq w_{ji}$, our final similarity measure s_{ij} is forced to be symmetric by setting $s_{ij} = (w_{ij} + w_{ji})/(2)$. After we have successfully identified the similarity matrix $S = [s_{ij}]$, we may run any spectral clustering algorithm [19] or a simple hierarchical agglomerative clustering algorithm to cluster the data.

[1] We fix $\beta = 0.01$ in our experiments.

We remark here that our nonnegative sparsity induced similarity measure is partly motivated by the work of Cheng *et al.* [18]. The most obvious difference is that we introduce the nonnegative constraints into the formulation, while their formulation allows the reconstruction coefficients to be negative, which may not be desirable since it is in conflict with one of the two assumptions the authors made, i.e., a sample can be linearly reconstructed from a small set of samples from the same cluster. This is probably the reason that the negative coefficients have to be forcefully set to zero in their algorithm when defining the final distance measure. Similar to [18], one may also pick up the k nearest neighbors of $\mathbf{x}_i$ to form $\mathbf{X}_i$ instead of using all the other $n-1$ data samples, to save the expensive computational cost.[2]

With this nonnegative sparsity induced similarity measure, many different clustering algorithms can be employed. In particular, we leverage a spectrum clustering algorithm proposed by Song *et al.* [19] to obtain the clustering results from the similarity matrix calculated from a set of images.

Last but not least, based on our current algorithmic design, it may be difficult to adapt the clustering results with the adding of new images without recalculating all the affected similarities. Fortunately in our system design the server-side cluster analysis only need to run in a batch mode. In other words, the cluster analysis will be performed on a set of image attachments received in an e-mail server within a period time. It is not necessary to regroup the cluster results when the next batch of e-mails are received.

IV. Client-Side Image Spam Detection

We present our client-side image spam detection subsystem based on active learning in this section. In particular, we present the system framework in Section IV-A. Then in Section IV-B, we present two candidate active learning algorithms for our subsystem, one based on a support vector machine (SVM) and the other based on a Gaussian process classifier. Their performances will be compared in our experiments.

A. System Framework

The system diagram of our client-side filtering subsystem is shown in Fig. 4. To differentiate spam images from normal image attachments, the whole dataset is split into two: the labeled dataset and unlabeled dataset. The labeled dataset is denoted as $\mathcal{X}_L = \{\mathbf{x}_i \,|\, i \in L\}$, with labels $\mathcal{Y}_L = \{y_i \in \{-1,+1\} \,|\, i \in L\}$, where 1 represents the spam image and -1 represents the nonspam image, respectively. The unlabeled dataset is denoted as $\mathcal{X}_U = \{\mathbf{x}_i \,|\, i \in U\}$. We assume $L = [1,n]$ and $U = [n+1, N]$. Let $\mathcal{X} = \mathcal{X}_L \cup \mathcal{X}_U$. When the system is first used, $\mathcal{X}_L$ is an empty set ϕ and $\mathcal{X}_U$ may cover the full dataset $\mathcal{X}$. We randomly choose a few (<10) spam images and nonspam images to label and take them as the initial labeled dataset for training the first round classifier.

The core of this prototype system is an active learning algorithm with a data sample choosing criterion $\mathrm{AL}(y(\mathbf{x}))$, where $y(\mathbf{x})$ is the classifier induced from the learning algorithm.

[2]We fix $k = 100$ in our experiments.

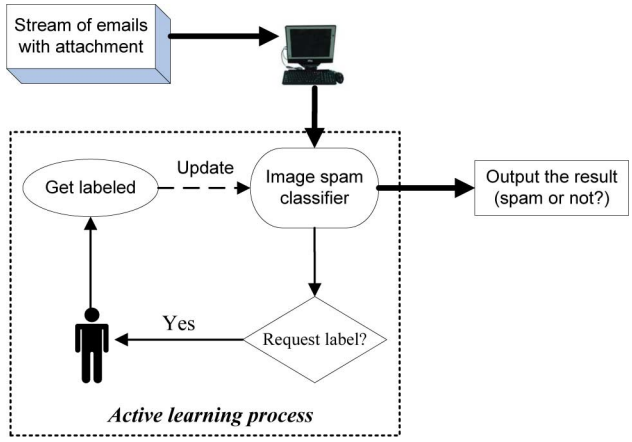


Fig. 4. Prototype system diagram of active learning image spam detection.

As long as an appropriate mathematic quantity $\mathrm{AL}(y(\mathbf{x}))$ is defined, we can make any supervised learning algorithm to be an active learning algorithm. The active learning criterion $\mathrm{AL}(y(\mathbf{x}))$ efficiently guides the users to label as few images as possible while maximizing the recognition performance of the classifier.

More formally, at each step of the active learning algorithm, we first perform the supervised learning algorithm with the current $\mathcal{X}_L$, and build the image spam classifier $y(\mathbf{x})$. Next we select

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathcal{X}_U} \mathrm{AL}(y(\mathbf{x})) \tag{6}$$

to label and get

$$\mathcal{X}_L \Leftarrow \mathcal{X}_L + \mathbf{x}^* \tag{7}$$

$$\mathcal{X}_U \Leftarrow \mathcal{X}_U - \mathbf{x}^*. \tag{8}$$

With the new $\mathcal{X}_L$, the above active learning step is iterated until the recognition accuracy of the classifier reaches a satisfactory level. We will discuss the selection of iteration times in our experiments in Section VI-B. In this way, the continuously adaptive classifier is generated and ready to filter the incoming batch of new e-mails with image attachment.

B. Active Learning Algorithms

We present two different active learning algorithms in this section. One is adapted from the probabilistic output of an SVM [23], [24]. The other is built on top of a Gaussian process (GP) classifier [25], [26]. We shall remark here that in this active image classification problem, it is extremely important that all the operations be running in real-time. Meanwhile the active learning process need to quickly achieve very high true positive rate and maintain extremely low false positive rate at the same time. From the users' perspective, it is even more annoying to dig a misclassified e-mail with normal image attachment out of the trash box. On the other hand, the users will not bear delayed responses from the active learner so it is extremely important that the computing of the visual representation and the learning algorithm be very efficient.

1) Active Learning SVM: Given the labeled data set $\mathcal{X}_L = \{\mathbf{x}_i, y_i\}_{i=1}^n$, the primal problem of a linear SVM solves the following quadratic program for obtaining the maximum margin linear classifier [27], [28], i.e.:

$$\min_w \quad \frac{\|\mathbf{w}\|^2}{2} + C\sum_i \xi_i \tag{9}$$

$$\text{s.t.} \quad y_i(\mathbf{x}_i \cdot \mathbf{w} + b) \le 1 - \xi_i \quad \text{and} \quad \xi_i \le 0 \,\forall i \tag{10}$$

where $\mathbf{w}$ is the linear projection, and each ξ_i is a slack variable to deal with the situation when the data set is not completely separable. The solution of the above constrained optimization problem is usually obtained by solving the Wolfe dual problem

$$\max \quad \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j \mathbf{x}_i \cdot \mathbf{x}_j \tag{11}$$

$$\text{s.t.} \quad 0 \le \alpha_i \le C \,\forall i \quad \text{and} \quad \sum_i \alpha_i y_i = 0 \tag{12}$$

where each α_i is corresponding to a Lagrangian multiplier. It shows that the solution is given by

$$\mathbf{w} = \sum_{i=1}^{N_s} \alpha_i y_i \mathbf{x}_i \tag{13}$$

where N_s indicates the number of support vectors for the classifier. Therefore, the classification result of a new data vector $\mathbf{x}$ is

$$y = \text{sign}\left\{\sum_{i=1}^{N_s} \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b\right\}. \tag{14}$$

It is easy to observe that in both the Wolfe dual problem (11) and the final classifier (14), the data vectors only present in the form of dot product. This enables us to construct nonlinear SVM by leveraging the kernel tricks [28], i.e., to solve the following problem:

$$\max \quad \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) \tag{15}$$

$$\text{s.t.} \quad 0 \le \alpha_i \le C \,\forall i \tag{16}$$

$$\sum_i \alpha_i y_i = 0 \tag{17}$$

where $k(\mathbf{x}_i, \mathbf{x}_j)$ is a kernel function which defines the dot product of the nonlinear transformed data vectors $\phi(\mathbf{x}_i)$ and $\phi(\mathbf{x}_j)$ in a reproducing kernel Hilbert space (we use Gaussian radial basis kernel in our experiments), i.e.,

$$k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j). \tag{18}$$

Similarly, the final nonlinear SVM classifier is

$$y = \text{sign}\left\{\sum_{i=1}^{N_s} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + b\right\}. \tag{19}$$

Note that we do not need to explicitly define the nonlinear transformation $\phi(\mathbf{x})$ since both the optimization problem in (15) and the solution in (19) only involve the kernel function. As shown by Madevska-Bogdanova *et al.* [23], we could transform the function output from an SVM to be a posterior distribution by applying a Sigmoid function, i.e.,

$$p(y = 1 \,|\, \mathbf{x}) = \frac{1}{1 + \exp\left\{a\left(\sum_{i=1}^{N_s} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + b\right)\right\}} \tag{20}$$

where a is a constant quantity which could be estimated from the training data. With this posterior probability of the predicted label given the data point, a natural active learning criterion would be based on the uncertainty of the predicted label given a data point. Let $p_1 = p(y = 1 \,|\, \mathbf{x})$. The uncertainty is naturally defined by an entropy term

$$H(y(\mathbf{x})) = -p_1 \log p_1 - (1 - p_1)\log(1 - p_1). \tag{21}$$

Therefore, for this active learning SVM, we define

$$\text{AL}(f(\mathbf{x})) = H(y(\mathbf{x})). \tag{22}$$

The rationale behind the criterion is that the active learning algorithm should guide the users to label the image for which the classifier is least confident to recognize.

2) Active Learning Gaussian Process Classifier: Given the labeled dataset $\mathcal{X}_L$, an unlabeled data $\mathbf{x}_u$, and $\mathcal{X}_{Lu} = \mathcal{X}_L + \mathbf{x}_u$, we introduce a latent variable z_i, which is the soft label of the data point $\mathbf{x}_i$. We denote $\mathcal{Z}_{Lu} = \{z_i \,|\, i \in L + u\}$. In a Gaussian process classifier, the joint distribution of $\mathcal{Z}_{Lu}$ is assumed to be a joint Gaussian with zero mean and covariance defined by a kernel function $k(\cdot, \cdot)$ applied to $\mathbf{x}_i$ and $\mathbf{x}_j$, i.e.,

$$p(\mathcal{Z}_{Lu} \,|\, \mathcal{X}_{Lu}) \sim \mathcal{N}(\mathbf{0}, \mathbf{K}) \tag{23}$$

where $\mathbf{K}$ is an $N \times N$ matrix with the element $k_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. We denote K_{LL} to be the submatrix of $\mathbf{K}$ that is induced by $\mathcal{X}_L$. Following Kapoor *et al.* [26], we assume $p(y \,|\, z)$ is a Gaussian distribution $\mathcal{N}(y, \sigma^2)$. We immediately have

$$p(\mathcal{Z}_{Lu} \,|\, \mathcal{X}_{Lu}, \mathcal{Y}_L) \propto p(\mathcal{Z}_{Lu} \,|\, \mathcal{X}_{Lu}) p(\mathcal{Y}_L \,|\, \mathcal{Z}_{Lu}) \tag{24}$$

$$= p(\mathcal{Z}_{Lu} \,|\, \mathcal{X}_{Lu}) \prod_{i \in L} p(y_i \,|\, z_i). \tag{25}$$

Denoting y_u to be the label of $\mathbf{x}_u$ we would like to predict, we are interested in inferring the following quantity:

$$p(y_u \,|\, \mathcal{X}_{Lu}, \mathcal{Y}_L) = \int_{\mathcal{Z}_{Lu}} p(y_u \,|\, \mathcal{Z}_{Lu}) p(\mathcal{Z}_{Lu} \,|\, \mathcal{X}_{Lu}, \mathcal{Y}_L) d\mathcal{Z}_{Lu}. \tag{26}$$

Denoting $\mathbf{k}(\mathbf{x}_u) = [k(\mathbf{x}_u, \mathbf{x}_1), k(\mathbf{x}_u, \mathbf{x}_2), \ldots, k(\mathbf{x}_u, \mathbf{x}_n)]^T$, and $\mathbf{I}$ to be the identity matrix, by following [26], we have

$$p(y_u \,|\, \mathcal{X}_{Lu}, \mathcal{Y}_L) = \mathcal{N}\left(\bar{y}_u, \bar{\sigma}_u^2\right) \tag{27}$$

where

$$\bar{y}_u = \mathbf{k}(\mathbf{x}_u)^T (\sigma^2 \mathbf{I} + \mathbf{K}_{LL})^{-1} \mathcal{Y}_L \tag{28}$$

$$\bar{\sigma}_u^2 = k(\mathbf{x}_u, \mathbf{x}_u) - \mathbf{k}(\mathbf{x}_u)^T (\sigma^2 \mathbf{I} + \mathbf{K}_{LL})^{-1} \mathbf{k}(\mathbf{x}_u) + \sigma^2. \tag{29}$$

Denoting $p_1 = p(y_u = 1 \mid \mathcal{X}_{Lu}, \mathcal{Y}_L)$, we can define the entropy by using (21), and the active learning criterion would exactly take (22). It is worth noticing that Kapoor *et al.* [26] defined their active learning criterion with this GP classifier to be

$$\mathrm{AL}(y(\mathbf{x})) = -\frac{|\bar{y_u}|}{\bar{\sigma}_u}. \tag{30}$$

In this binary classification problem, it is easy to verify that this is equivalent to our entropy uncertain measure.

V. Dataset, Feature Representation, and Evaluation Criteria

A. Data Set

We collected an image dataset which contains 1190 spam images and 1760 normal images. The spam images are extracted from real spam images received by ten graduate students in our department between January 2006 and March 2009. These spam images were extracted from the original spam e-mails and all of them are converted to JPEG format. For normal image attachments, we collect photo images by either downloading from photo sharing sites such as Flikr.com, or fetching the photo images from popular image search engines such as Microsoft Bing search (http://www.bing.com/images?FORM=Z9LH4). This dataset was first utilized and reported in [8].

B. Image Features

We extract an effective set of 23 discriminant image statistical features [29] for our image spam filter tasks, both on the server and client side. They cover the properties of color, texture, shape, and appearance.

For color statistics, we first build a 10^3-dimension color histogram in the joint RGB space by quantizing each color band into 10 different levels. The *entropy* of this histogram is computed as the first statistics. We further set up one 100-dimensional histogram for each of the three color channels. Then the *discreteness*, *mean*, *variance*, *skewness*, and *kurtosis* for each of the three histograms are calculated, which adds another $5 \times 3 = 15$ statistics. Here the discreteness is the summation of all the absolute differences between any two consecutive bins. So altogether we collect 16 color statistics.

A local binary pattern (LBP) [30] is used to analyze the texture statistics. We extract a 59-dimensional texture histogram, including 58 bins for all the different uniform local binary patterns, i.e., the pattern of at most two 0–1 transitions in a 8-bit stream, and an additional bin for all other nonuniform local binary patterns. The *entropy* of the LBP histogram is calculated as 1 texture statistics.

Shape information is also considered as an important feature in our system. A $40 \times 8 = 320$ dimensional gradient magnitude-orientation histogram is built to describe the shape information. The *entropy* of the histogram is the first shape feature, and the second feature is the difference between the energies in the lower frequency band and the higher frequency band. Then we use the total amount of edges and the average length of the edges as another two shape feature by running a Canny edge detector [31]. Thus there are four shape statistics in total.

Last but not least, we use the spatial correlogram [32] of the gray level pixels within 1-neighborhood to represent appearance information. The first feature is the *average variance ratio* of all the slices of the correlogram, which is 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. Then histograms are built from each slice of the correlogram, and the *average skewness* of the histograms is calculated as the second feature.

These features are motivated by the fact that spam images usually present different visual statistics when compared with natural or normal images. Therefore, adopting them as visual representations may naturally discriminate spam images from normal or natural images. To illustrate this well, we randomly pick up a set of images from our data collection. It contains 200 spam images and 200 nonspam images. We can then plot the two distributions of each of the 23 feature values in the two classes to visualize how discriminative each feature is. Due to the space limitation, we present four such figures, as shown in Fig. 5. It clearly presents that the adopted statistic visual features can separate the normal images from spam images very well. As we can clearly observe, there is clear modes separation between normal and spam images from all the feature distributions we plotted.

As we have discussed, it is very important that the visual computing part, i.e., the calculation of these image statistics, to be efficient. As a matter of fact, in our experiments, the average computing time for extracting these 23 visual statistics from an 320×240 image is less than 10 ms.

C. Evaluation Criteria

1) Server-Side Evaluation Criterion: Assume we have ground truth cluster labels of our data collection; we use two criteria to evaluate the performance of all the clustering results [33]. The first criterion is the average clustering accuracy CAC, which is defined as

$$\mathrm{CAC} = \frac{1}{n}\sum_{i=1}^{n} \delta(m(cl_i) - l_i) \tag{31}$$

where n is the total number of data points, $\delta(\cdot)$ is the Dirac-Delta function, cl_i and l_i are the cluster id and the labeled cluster id of data point i, respectively, and $m(\cdot)$ is the best map of cl_i to the ground truth cluster id, which can be optimally resolved by the Kuhn-Munkres algorithm [34].

The second evaluation criterion we adopt is the normalized mutual information [33] between the cluster results C' and the ground truth clusters C, which is defined as

$$\mu\mathrm{MI} = \frac{\mathrm{MI}(C, C')}{\max(H(C), H(C'))} \tag{32}$$

where $H(\cdot)$ represents the entropy of the cluster set and $\mathrm{MI}(C, C')$ is the mutual information between the two cluster sets, i.e.,

$$\mathrm{MI}(C, C') = \sum_{c_i \in C, c'_j \in C'} p(c_i, c'_j) \log \frac{p(c_i, c'_j)}{p(c_i)p(c'_j)}. \tag{33}$$

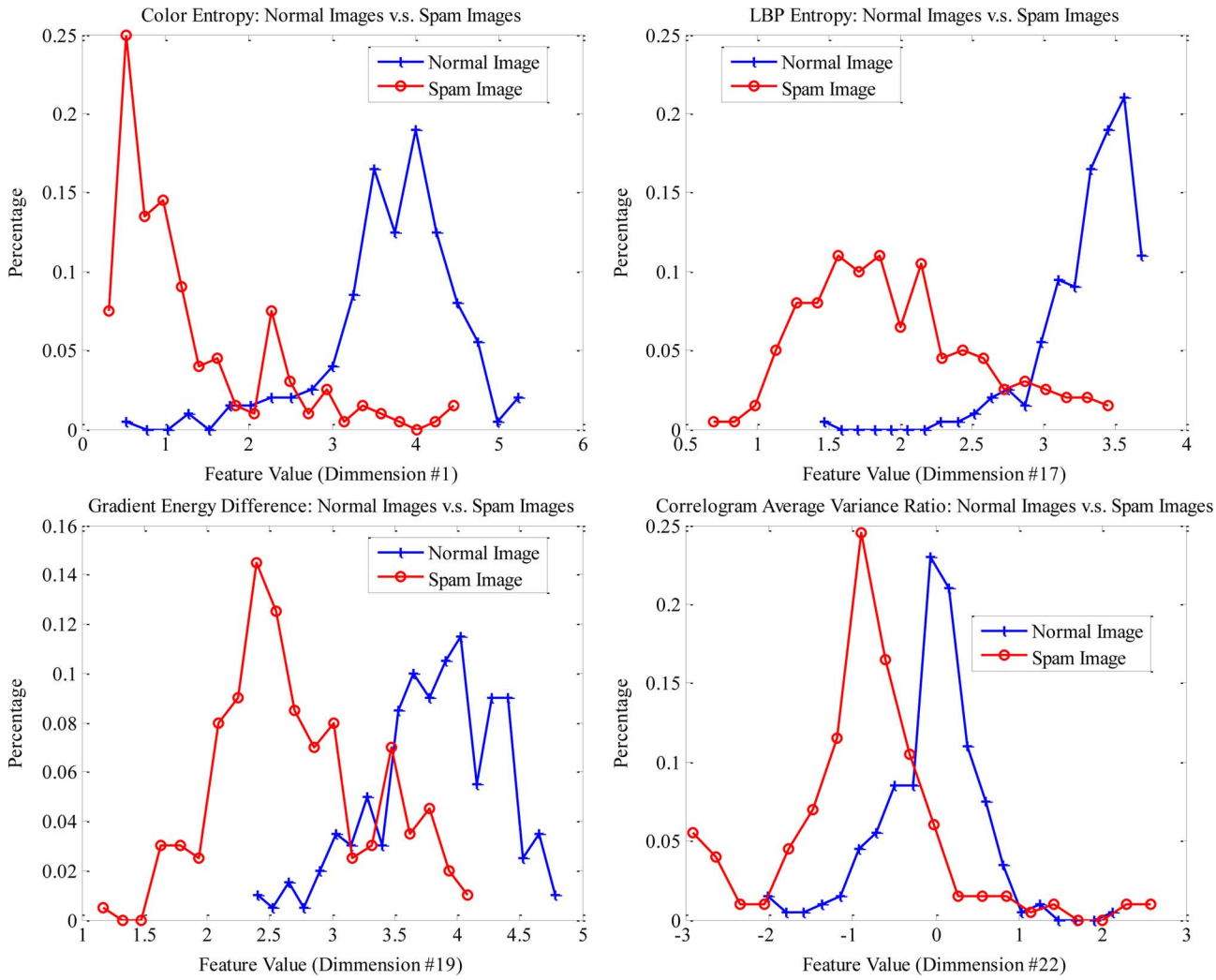


Fig. 5. Feature distributions of normal images and spam images. From left to right, top to bottom, we present the distributions on feature dimension #1, #17, #19, and #22, respectively. All figures present clear separations between normal and spam images.

It is easy to figure out that $\mu\mathrm{MI} \in [0, 1]$, with $\mu\mathrm{MI} = 0$ if the two cluster sets are independent and $\mu\mathrm{MI} = 1$ if the two cluster sets are identical.

2) Client-Side Evaluation Criterion: To evaluate our active learning classifier, we follow the tradition of the literature in evaluating classification algorithms, and hence adopt a set of criteria such as *false positive rate* (FPR), which refers to the percentage of the normal images being erroneously classified as spam images, and *true positive rate* (TPR), which refers to the percentage of spam images that are classified correctly by the classifier.

VI. Experiments

A. Server-Side Evaluation: Cluster Analysis

We compare the proposed similarity measure with two other competitive measures. The first one is the sparsity induced similarity measure without posing the nonnegative constraint, i.e., we simply remove the nonnegative constraints in (2), set $\beta = 0$, and change $\sum_j a_j$ to $\sum_j |a_j|$ in (1). Then the problem becomes a standard Lasso regression problem.[3] This similarity measure is first proposed in [18].[4] The other one is a baseline similarity measure which is induced from the Euclidean distance by applying a Gaussian radial basis function (RBF). For each of the similarity measures, we build the similarity graph matrix and use a spectral cluster algorithm [19] to generate the clustering results.

1) Comparison Results: To evaluate the performance of the proposed nonnegative similarity measure for cluster, we first manually labeled a set of clusters out of the collected 1190 spam images. More specifically, we labeled 37 clusters which covers 756 of the spam images. The number of images in a cluster could be as high as 160, and as low as just 1, as shown in Fig. 6. These 37 clusters of spam images composed the evaluation data set in our experiments. We summarize the cluster performance using three different distance measures in Table I. We name the results of three different similarity measures as NonNegSparse

[3]We solve it with Gaussian–Seidel method using the Matlab code provided at http://people.cs.ubc.ca/schmidtm/Software/lasso.html.

[4]Cheng *et al.* [18] cast it in a slightly different optimization problem, but it should essentially achieve very similar results.

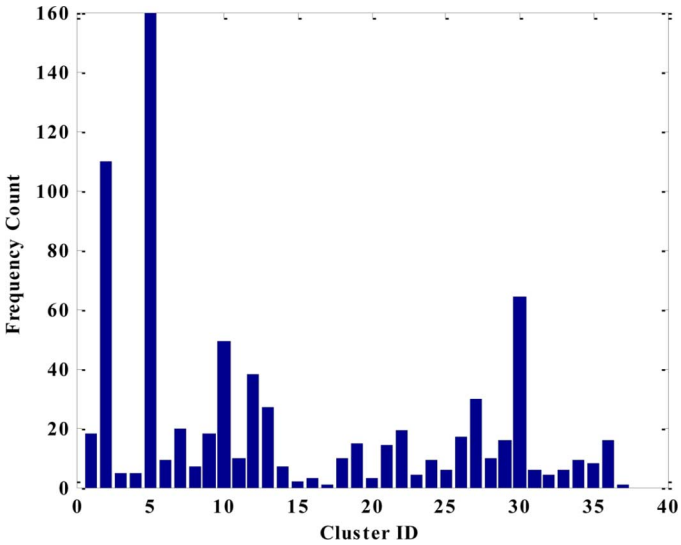


Fig. 6. Number of images in each of the 37 clusters.

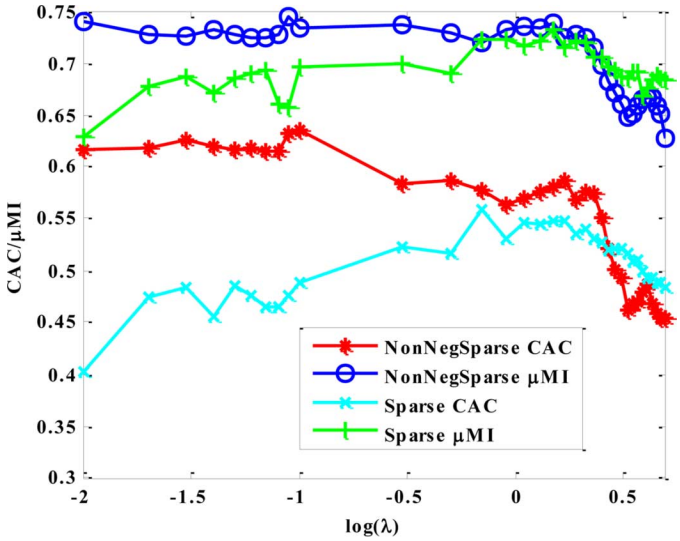


Fig. 7. Changes of cluster performance criteria w.r.t. the λ (in log scale).

TABLE I
CLUSTERING PERFORMANCE OF THREE DIFFERENT SIMILARITY MEASURES

	NonNegSparse	Sparse	Euclidean
CAC	0.635± 0.006	0.559± 0.032	0.485± 0.019
μMI	0.734± 0.005	0.671± 0.006	0.471± 0.025
	$\lambda = 0.1$	$\lambda = 0.7$	–

(our proposed one), Sparse (Cheng *et al.* [18]), and Euclidean (Gaussian RBF baseline). Since the final step of the spectral clustering algorithm [19] is running a k-means, each run of the spectral cluster will result in slightly different clustering results due to different initialization of the k-means iterations. Therefore, we run the spectral clustering 500 times for each case and the results reported in the table are the mean value plus/minus the standard deviation over all the runs.

As we can clearly observe, the proposed nonnegative sparsity induced similarity measure achieves the best clustering performance with CAC $= 0.635$ and μMI $= 0.734$, with a parameter setting $\lambda = 0.1$. This significantly improves the best results achieved by Cheng *et al.* [18], which obtains CAC $= 0.559$ and μMI $= 0.671$ with $\lambda = 0.7$. Nevertheless, both algorithms lead the baseline Gaussian RBF similarity by a significant margin (CAC $= 0.485$ and μMI $= 0.471$), as shown in the table. The standard deviations of the performance quantities of the proposed approach also seem to be smaller than those of the competition methods, which is an indication that the proposed similarity measure is more preferable since the clustering results from it are less sensitive to the initialization of k-means after spectral embedding.

We shall remark here that the weight factor λ in (1) has an impact on both the nonnegative sparsity induced similarity measure and the sparsity induced similarity measure. Therefore, we run the above cluster analysis with different settings of λ for both algorithms, and plot the changes of the cluster performance criteria w.r.t. λ (in log scale) in Fig. 7. It again demonstrates the better performance of the proposed nonnegative sparsity induced similarity measure.

We notice that the two evaluation criteria, CAC and μMI, are not always strictly tied with each other. That is, when CAC achieves the optimal value, the μMI may not achieve the best simultaneously, and vice versa. We regard CAC as a more direct criterion, so we pick up the working parameter λ based on it in Table I.

B. Client-Side Evaluation: Active Learning Classification

In our experiments, we report the recognition accuracy on both the *active learning pool* $\mathcal{X} = \mathcal{X}_L \cup \mathcal{X}_U$, and a *hold-out data-set* $\mathcal{X}_h$. We keep track of the recognition accuracy with the progress of active learning. We also compare with a baseline setting where at each step we randomly choose an image sample from $\mathcal{X}_u$ for the users to label. We call the active learning process to be *active supervision* and the baseline setting to be *random supervision*. We adopt the Gaussian radial basis kernel for both the SVM and the GP classifier.

1) Results Comparison: Since typical users usually deal with hundreds of e-mails in a one-day batch, we randomly extract a subset of 10% images from the whole data corpus as the test subset in each experiment. To test the generalization performance of the classifiers induced from active learning, each time we randomly sample 20% data from the test subset as a hold-out dataset $\mathcal{X}_h$. The rest 80% is adopted as the active learning pool $\mathcal{X}$. We randomly select ten samples from active learning pool to initialize the system. Fig. 8 presents the experimental results averaged over 100 runs. Fig. 8(a) presents the progressive changes of the overall recognition accuracy, false positive and true positive rates on $\mathcal{X}$ with the human adding more and more labels, while (b) shows the results on $\mathcal{X}_h$.

In general, the classifiers induced form active supervision achieve much better results than those from random supervision; in other words, much less labels are needed for active supervision to achieve the same recognition accuracy as random supervision. In particular, the active learning SVM only requires us to label less than 50 images in $\mathcal{X}$ to achieve over 99% recognition accuracy. This is also observed in the holdout dataset where the recognition accuracy quickly approaches to the saturation point

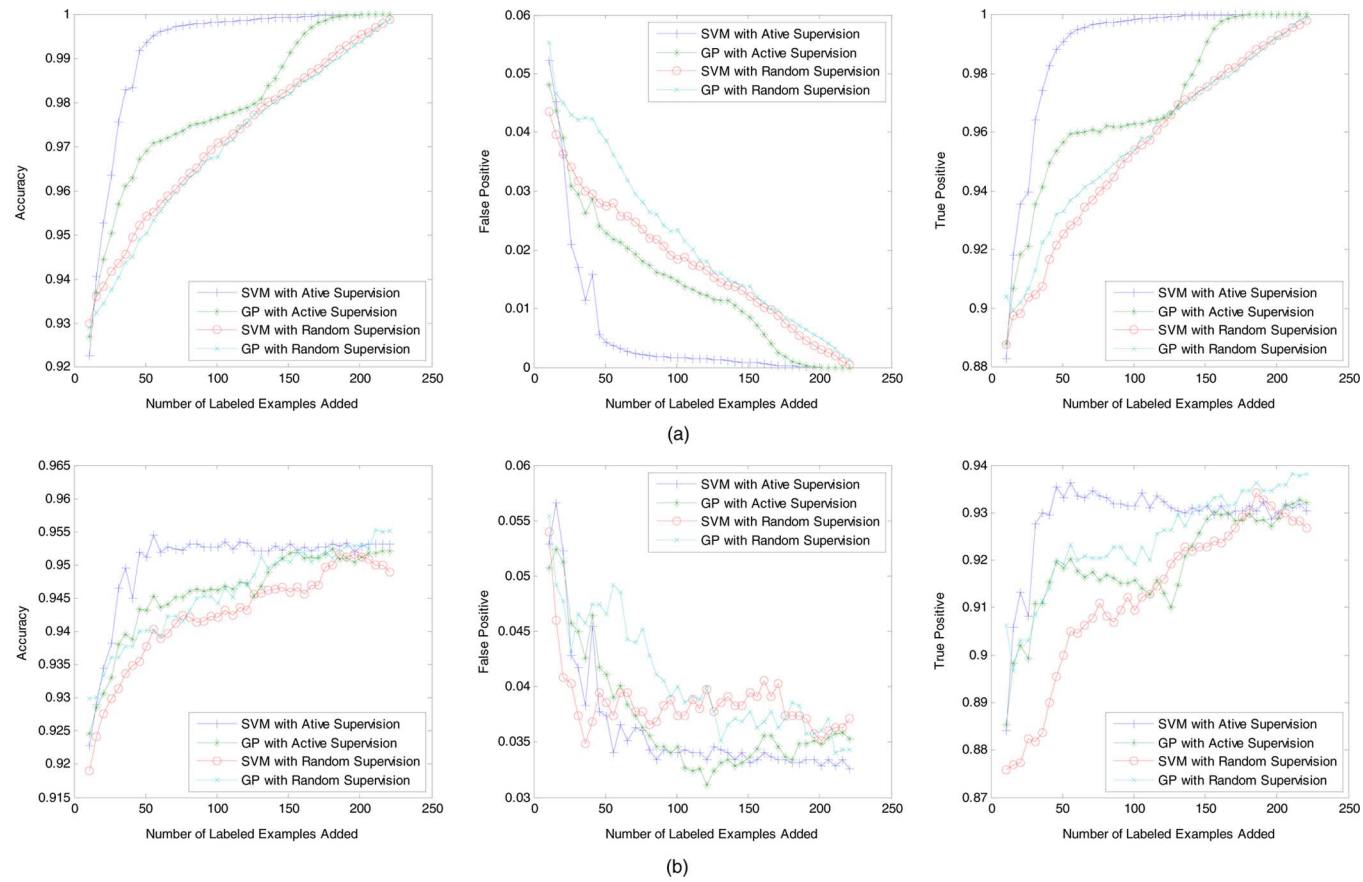


Fig. 8. Changes of the accuracy with the progress of active learning. (a) The progressive changes of the overall recognition accuracy, false positive, and true positive rates on active learning pool X. (b) The progressive changes of the overall recognition accuracy, false positive, and true positive rates on hold-out data pool X_h.

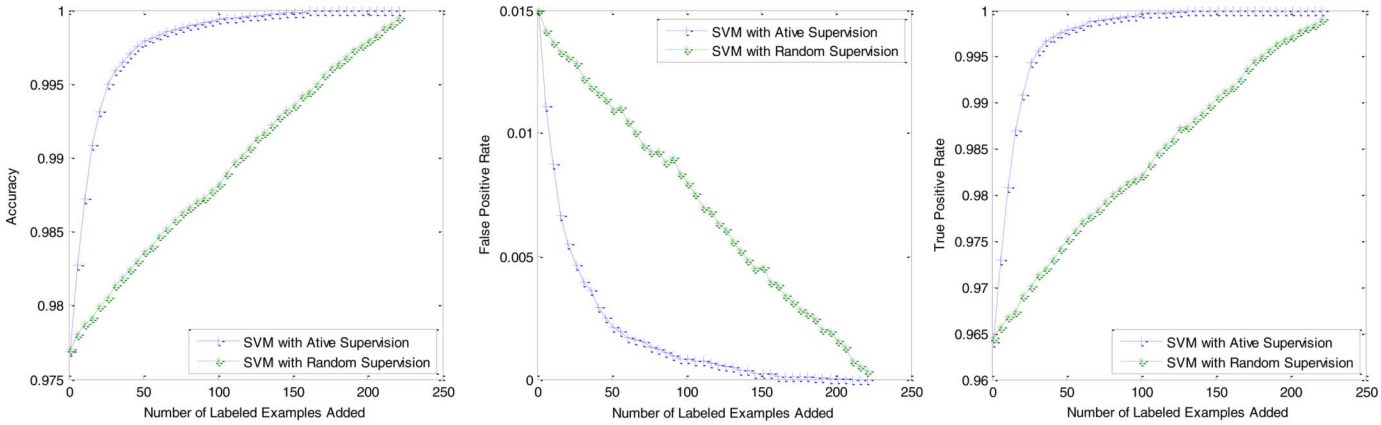


Fig. 9. Recognition accuracy of running active learning SVM on an initialized classifier.

than the algorithms with random supervision. Moreover, with our feature setting and the selected kernel function, the active learning SVM consistently shows better performance than the active learning GP classifier.

The recognition performance on $\mathcal{X}_h$ also shows that the induced classifier generalizes well so that it may be employed for fully automated image spam filtering. But it is preferred to always run in the active learning mode as we can ensure more than 99% accuracy by the end of the learning process. If not considering the initialization process of the system, the amount of labels required to adapt the classifier to the next batch of e-mails is even less. Fig. 9 presents the recognition performance of continuously running the active SVM algorithm on a second subset of data, initialized from the SVM classifier obtained from the first subset. The reported results are also averaged over 100 different runs. As we can clearly observe, with a well-trained initial SVM, the active learning SVM only requires us to label 20 ($<7\%$) images in order to achieve over 99% recognition accuracy. That is to say, our client-side active learning image spam hunter only needs $<7\%$ label data to get the ideal high detection rate. This

ratio may further reduce with the increase of the dataset. In our experiment, on average adapting the SVM classifier in each step of the active learning process is always less than 0.5 s.

In our previous work [7], [8], we have explored the purely supervised classifier [7] such as probabilistic boosting tree [35], as well as the semisupervised learning algorithm [8] for classifying the the image spams. It is beyond the scope of this paper to have an extensive discussion of results from these two previous works, and interested readers may refer to them for detailed discussion. Nevertheless, although an automated system is always our dream goal, we shall argue that none of these classifier could achieve 100% accuracy if it is achievable at all. Hence, the users need to make the final check on the automated system anyway, such as manually checking the spam folder for potential false positives. Therefore, it may be desirable to involve the users from the beginning in the proposed active learning framework.

VII. Conclusion and Future Work

In this paper, we presented a comprehensive solution to image spam filtering, which combines cluster analysis of spam images on the server side and active learning classification on the client side for effectively filtering image spams.

For server-side mitigation, we propose a nonnegative sparse representation induced similarity measure to be used together with spectral clustering algorithm for clustering analysis of spam images. Then relatively larger groups of images are suspected to be spams, which can be further analyzed to identify the spam sources. The spam sources can then be blocked on the server side directly without reaching e-mail users. For those spam images that survived this server-side filtering and reached the client of e-mail users, we propose a prototype active learning spam hunter to enable the users to efficiently and interactively filter out the spam images.

Extensive experimental evaluations of both server-side algorithms and client-side algorithms on a real image spam dataset collected from an e-mail server demonstrated the efficacy of the proposed comprehensive solution. Our future works may include, but not necessarily limited to, 1) further combining our server-side system with IP tracing techniques to identify the source IP or e-mail account of the spammer; 2) exploring embedded UI/UX designs to fit in our client-side active learning spam hunter with mainstream e-mail clients such as Office Outlook; 3) investigating more discriminative image features for dealing with spam images even more effectively.

Acknowledgment

The authors would like to thank Dr. Z. Liu and Dr. M. Yang for helpful discussions.

References

[1] Sophos Plc [Online]. Available: http://www.sophos.com/pressoffice/news/articles/2008/07/dirtydozjul08.html

[2] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A bayesian approach to filtering junk e-mail," in *Proc. AAAI Workshop on Learning for Text Categorization*, Madison, WI, Jul. 1998.

[3] H. Drucker, D. Wu, and V. N. Vapnik, "Support vector machines for spam categorization," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 1048–1054, Sep. 1999.

[4] X. Carreras and J. G. Salgado, "Boosting trees for anti-spam email filtering," in *Proc. 4th Int. Conf. Recent Advances in Natural Language Processing*, Tzigov Chark, BG, 2001, pp. 58–64.

[5] McAfee [Online]. Available: http://www.avertlabs.com/research/blog/?p=170

[6] SpamAssassin [Online]. Available: http://spamassassin.apache.org

[7] Y. Gao, M. Yang, X. Zhao, B. Pardo, Y. Wu, T. Pappas, and A. Choudhary, "Image spam hunter," in *Proc. 33th IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Las Vegas, NV, Apr. 2008.

[8] Y. Gao, M. Yang, and A. Choudhary, "Semi supervised image spam hunter: A regularized discriminant em approach.," in *Proc. 5th Int. Conf. Advanced Data Mining and Applications*, Beijing, China, Aug. 2009, vol. LNCS 5678, pp. 152–164.

[9] M. Dredze, R. Gevaryahu, and A. Elias-Bachrach, "Learning fast classifiers for image spam," in *Proc. 4th Conf. Email and Anti-Spam (CEAS)*, California, Aug. 2007.

[10] Z. Wang, W. Josephson, Q. Lv, M. Charikar, and K. Li, "Filtering image spam with near-duplicate detection," in *Proc. 4th Conf. Email and Anti-Spam (CEAS)*, California, Aug. 2007.

[11] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby, "Selective sampling using the query by committee algorithm," *Machine Learning*, vol. 28, pp. 133–168, 1997.

[12] S. Tong, D. Koller, and P. Kaelbling, "Support vector machine active learning with applications to text classification," *J. Machine Learning Res.*, pp. 999–1006, 2001.

[13] K.-S. Goh, E. Y. Chang, and W.-C. Lai, "Multimodal concept-dependent active learning for image retrieval," in *Proc. 12th Ann. ACM Int. Conf. Multimedia*, New York, 2004, ACM.

[14] N. D. Lawrence, M. Seeger, and R. Herbrich, "Fast sparse Gaussian process methods: The informative vector machine," in *Advances in Neural Information Processing Systems 15*. Cambridge, MA: MIT Press, 2003, pp. 609–616.

[15] D. J. C. MacKay, "Information-based objective functions for active data selection," *Neural Comput.*, vol. 4, pp. 590–604, 1992.

[16] Y. Gao and A. Choudhary, "Active learning image spam hunter," in *Proc. 5th Int. Symp. Visual Computing*, Las Vegas, NV, Nov. 2009.

[17] B. Mehta, S. Nangia, M. Gupta, and W. Nejdl, "Detecting image spam using visual features and near duplicate detection," in *Proc. 17th Int. World Wide Web Conf.*, Beijing, China, Apr. 2008.

[18] H. Cheng, Z. Liu, and J. Yang, "Sparsity induced similarity measure for label propagation," in *Proc. IEEE Int. Conf. Computer Vision*, Kyoto, Japan, Oct. 2009.

[19] Y. Song, W.-Y. Chen, H. Bai, C.-J. Lin, and E. Y. Chang, "Parallel spectral clustering," in *Proc. Eur. Conf. Machine Learning and Knowledge Discovery in Databases*, Antwerp, Belgium, Sep. 2008.

[20] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Royal Statist. Soc.*, ser. B, vol. 58, pp. 267–288, 1994.

[21] A. Neumaier, Minq-General Definite and Bound Constrained Indefinite Quadratic Programming 1998 [Online]. Available: http://www.mat.univie.ac.at/~neum/software/minq

[22] L. Benaroya, L. McDonagh, F. Bimbot, and R. Bribonval, "Non negative sparse representation for wiener based source separation with a single sensor," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Apr. 2003, vol. 6, pp. 613–616.

[23] A. Madevska-Bogdanovaa, D. Nikolikb, and L. Curfsc, "Probabilistic SVM outputs for pattern recognition using analytical geometry," *Neurocomputing*, vol. 62, pp. 293–303, Dec. 2004.

[24] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[25] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA: MIT Press, 2006.

[26] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell, "Active learning with Gaussian processes for object categorization," in *Proc. Eleventh IEEE Int. Conf. Computer Vision*, Rio de Janeiro, Brazil, Oct. 2007.

[27] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.

[28] A. Aizerman, E. M. Braverman, and L. I. Rozoner, "Theoretical foundations of the potential function method in pattern recognition learning," *Autom. Remote Control*, vol. 25, pp. 821–837, 1964.

[29] T.-T. Ng, S.-F. Chang, and M.-P. Tsui, "Lessons learned from online classification of photo-realistic computer graphics and photographs," in *Proc. IEEE Workshop on Signal Processing Applications for Public Security and Forensics (SAFE)*, Washington, DC, Apr. 2007.

[30] T. Mäenpä, "The Local Binary Pattern Approach to Texture Analysis Extensions and Applications," Ph.D. Dissertation, Infotech Oulu, University of Oulu, Oulu, Finland, Aug. 2003.

[31] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, Nov. 1986.
[32] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih, "Image indexing using color correlograms," in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, Los Alamitos, CA, 1997, IEEE Computer Society.
[33] D. Cai, X. He, W. V. Zhang, and J. Han, "Regularized locality preserving indexing via spectral regression," in *Proc. 2007 ACM Int. Conf. Information and Knowledge Management*, Lisboa, Portugal, Nov. 2007.
[34] L. Lovasz and M. D. Plummber, *Matching Theory*. Amsterdam, The Netherlands: Elsevier, 1986.
[35] Z. Tu, "Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering," in *Proc. ICCV*, Beijing, China, Oct. 17–21, 2005, vol. 2, pp. 1589–1596.

Yan Gao received the B.E. degree in electrical engineering and the M.S. degree in system engineering, both from Xian Jiaotong University, China, in 2000 and 2003, respectively, and the Ph.D. degree in electrical engineering and computer science from Northwestern University, in June 2010.

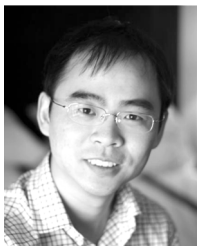
During the summer of 2007 and 2008, she fulfilled two summer research internship in Microsoft Research, Redmond, with the Internet Service Research Center. Her current research interests include data mining and machine learning algorithms and applications, network measurement and security. To date, she has published nearly 20 research papers, and has one U.S. patent pending.

Dr. Gao was the recipient of the prestigious Walter P. Murphy Fellowship, Morrison Fellowship, MEAS Fellowship, all from Northwestern University in 2004, 2006, and 2009, respectively.

Alok Choudhary (S'84–M'85–SM'00–F'05) received the M.S. degree from the University of Massachusetts, Amherst, in 1986, and the Ph.D. degree in electrical and computer engineering from the University of Illinois, Urbana-Champaign, in 1989.

He is the chair and professor of the Electrical Engineering and Computer Science Department, in the McCormick School of Engineering and Applied Science, Northwestern University, Evanston, IL. He is also a Professor at Kellogg School of Management. He is the founder and director of the Center for Ultra-scale Computing and Information Security (CUCIS). He joined Northwestern in 1996. Prior to that he was a faculty member of the ECE Department at Syracuse University. He has published more than 350 papers in various journals and conferences. He has also written a book and several book chapters. He has cofounded two companies. He cofounded Accelchip, Inc., a developer of electronic design automation tools and services. He served as its Vice President for Research and Technology from 2000 to 2002. He also cofounded Nimkathana Inc., a company that provided consulting services in the area of software systems, high-performance computing, cluster systems, data mining, databases, datawarehousing and other related topics. His team's scalable parallel software such as MPI-IO, and Parallel NetCDF are used by thousands of scientists worldwide. The datamining benchmark "NU-minebench" developed by his team is used by thousands of researchers. He has served or serves on the the technical advisory board of several companies. His research interests are in supercomputing and parallel computing, embedded systems, computer architecture, e-commerce and web-based systems, system software and algorithms, data mining, marketing and analytical marketing, customer relationship management, business intelligence, and information security. He has served on the editorial board of several journals, has chaired several international conferences, and has served on more than 75 program committees.

Dr. Choudhary received the National Science Foundation's Young Investigator Award in 1993 (1993–1999). He has also received an IEEE Engineering Foundation award, an Intel research council award (1993–1997, 2003–2005), and an IBM Faculty Development award. In 2006, he received the first award for "Excellence in Research, Teaching and Service" from the McCormick School of Engineering. He is a Fellow of ACM and AAAS. His research has been sponsored by (past and present) DARPA, NSF, NASA, AFOSR, ONR, DOE, Intel, IBM, and TI.



Gang Hua (S'03–M'06) received the B.S. degree in automatic control engineering and the M.S. degree in pattern recognition and intelligence systems from Xi'an Jiaotong University (XJTU) in 1999 and 2002, respectively. He received the Ph.D. degree in electrical and computer engineering from Northwestern University, Evanston, IL, in 2006.

He is a Research Staff Member at IBM Research T. J. Watson Center, Hawthorne, NY. Before that, he was a Senior Researcher at Nokia Research Center, Hollywood, from 2009 to 2010, and a Scientist at Microsoft Live Labs Research from 2006 to 2009. During the summer 2005 and summer 2004, he was a research intern with the Speech Technology Group, Microsoft Research, Redmond, WA, and a research intern with the Honda Research Institute, Mountain View, CA, respectively.

Dr. Hua was enrolled in the Special Class for the Gifted Young of XJTU in 1994. He received the Richter Fellowship and the Walter P. Murphy Fellowship from Northwestern University in 2005 and 2002, respectively. His current research includes human centered visual computing, and large-scale visual data analytics. He is a member of ACM. As of September, 2010, he holds two U.S. patent and has 17 more patents pending.