Blocking Objectionable Web Content by Leveraging Multiple Information Sources

Nitin Agarwal
Comp. Sci. & Eng.
Arizona State University
Tempe, AZ 85287
Nitin.Agarwal.2@asu.edu

Huan Liu
Comp. Sci. & Eng.
Arizona State University
Tempe, AZ 85287
Huan.Liu@asu.edu

Jianping Zhang
AOL, Inc.
44900 Prentice Drive
Dulles, VA 20166
Jianpingz032@aol.com

ABSTRACT

The World Wide Web has now become a humongous archive of various contents. The inordinate amount of information found on the web presents a challenge to deliver right information to the right users. On one hand, the abundant information is freely accessible to all web denizens; on the other hand, much of such information may be irrelevant or even deleterious to some users. For example, some control and filtering mechanisms are desired to prevent inappropriate or offensive materials such as pornographic websites from reaching children. Ways of accessing websites are termed as *Access Scenarios*. An Access Scenario can include using search engines (e.g., image search that has very little textual content), URL redirection to some websites, or directly typing (porn) website URLs. In this paper we propose a framework to analyze a website from several different aspects or information sources, and generate a classification model aiming to accurately classify such content irrespective of access scenarios. Extensive experiments are performed to evaluate the resulting system, which illustrates the promise of the proposed approach.

1. INTRODUCTION

The growth of internet traffic, the lack of central management of web contents, and the need to prevent people, especially children, from seeing offensive or inappropriate materials on the web have intensified the efforts to develop web filters that can effectively block intentional or unintentional accesses to certain objectionable websites such as pornographic ones. Accurate web filters rely on correct recognition of inappropriate web contents. One effective way is *content categorization* [12].

Both manual and automated approaches to web content categorization are adopted in practice. In a manual approach, human analysts tag websites with categories according to their contents. The URLs of these manually categorized websites together with their category tags are stored in a database for future use. To categorize a website, its URL is matched against the pre-categorized URLs in the database. If a match is found, the website is classified as the category of the matched URL. Otherwise, the category of the website is unknown. The main advantage of manual approach is its efficiency in categorization and high precision. However, the size of the web, the sheer number of new websites created on a daily basis, and many dynamically generated websites prevent such a manual approach from achieving a high recall. In addition, manually constructing and maintaining such a URL database is time-consuming, labor-intensive, expensive, and unscalable.

An automated approach applies machine learning techniques to create models of categories from the textual content of a set of pre-categorized training websites. The learned models are then applied to classify websites online. An automated approach complements a manual approach. It is able to assign a category to every website and to handle new and dynamically generated websites. It also is less time consuming and less labor intensive to create, and affordable. However, there are several challenges for an automated approach. One major challenge lies in data collection while training the classifier. More specifically, collecting negative samples is an arduous task. First, we want to avoid selection bias in the sense that negative instances should be carefully selected to represent a gamut of categories other than the positive class (e.g., pornographic websites). Second, the number of positive instances (say, porn websites) is often significantly less than that of negative instances considering all other websites as negative - the so-called imbalance problem with the training data. To overcome the first problem obviously requires more instances, which exacerbates the second problem; to mitigate the second problem is to reduce the number of negative instances, which likely worsens the first problem due to the potential number of categories. Given the fact that there do exist negative instances (albeit they are plethora), it is counter-intuitive to proceed without using the negative instances and to solely work on the positive instances. We propose to collect negative instance based on web taxonomy in order to avoid selection bias and data imbalance. We will discuss about how web taxonomy is used for negative instance collection in Section *3.2.1*.

The second challenge is that text/web categorization data are generally high-dimensional. We cannot consider all the features together because in high-dimensional feature space, classifiers may not perform well due to the curse of dimensionality [1]. We may reduce dimensions by either selecting the most relevant features (i.e., *feature selection* [17; 18]) or mapping high-dimensional space to low-dimensional feature space (i.e., *feature extraction* [16]). We may also perform subspace clustering [21] to find subset of features pertinent to a cluster or to a class in this case. Often these approaches find features that are relevant in the training set. But in this case the test set may not always have the same feature-value distribution. Some websites may be using some set

of adult[1] keywords and others might be using a completely different set. Moreover the websites tend to change the content frequently in order to cheat some (e.g., keyword-based) porn-blockers. Therefore, feature selection/extraction once learned might not always work. Nor is it feasible to learn relevant features every time you are classifying a website, due to the time overhead that will lead to intolerable delays in response to the user. Hence we need a technique that reduces the feature set while keeping in mind the changing relevant set of keywords for these websites. That is why we propose a density-based measure to calculate porn-indexes of websites. We will discuss about this in Section *3.2.4*.

Third challenge in building a content-based classifier is its inability to categorize websites accurately with little textual information. In the web filtering project of an internet company, for example, a combination of both a manual approach and an automated one was applied to identifying objectionable websites. While the combined approach worked well on regular objectionable websites, it failed to identify "non-regular" access to objectionable websites such as Google image searches, access to websites with little or no textual information or URL redirections. We call these as different *Access Scenarios*. By analyzing a set of access attempts by teenagers to pornographic websites, we found that more than half of the access attempts were image and keyword searches and about 3% involve accesses to websites with little text information. It is evident that textual content based filters alone cannot correctly categorize these attempts. A system is needed that can leverage multiple information sources to correctly classify these access attempts irrespective of the access scenarios.

In this paper, we focus on the above mentioned problems and make the following technical contributions.

- Glean data based on a generic taxonomy for negative instance collection to avoid the problems due to selection bias.
- Propose a density-based summarization of the features to reduce high-dimensional feature space and improve accuracy. The resulting summarized features play the role of different information sources.
- Propose a "Generative Model" for density-threshold based classification that automatically learns thresholds from training webpages and fine-tunes them as more data is acquired.
- Propose a "Relaxed Model" approach, wherein additional information about a webpage is useful, but not necessary, for better classification accuracy in which summarized features are used in conjunction with an SVM-based classifier to classify the access attempts. Additional information is useful to automatically handle different access scenarios.
- Consider the actual structure of a webpage rather than uniformly treat it as bag of words, verifying the fact that leveraging structural information is in general more robust than using only the raw text.

[1]'Adult', 'Porn' and 'Objectionable' are used interchangeably in the rest of the paper. They all refer to the same entity.

The rest of the paper is organized as follows. Section 2 discusses the related work to the problem addressed in this paper. In Section 3, we elaborate upon multiple information sources, data collection and its representations. We discuss approaches in Section 4 followed by a thorough evaluation and comparison of proposed approaches with a baseline Support Vector Machine (SVM) model in Section 5. We present the impact of the proposed approach for Web service providers such as AOL, Yahoo, etc. in Section 6. Finally, Section 7 concludes the paper and outlines further challenges in this direction.

2. LITERATURE REVIEW

A large amount of work has been devoted over the last few years on content filtering. It has become a social issue in information systems and is interdisciplinary in nature. Ever since the Communications Decency Act (CDA) was passed in 1995 and the Information Highway Parental Empowerment Group (IHPEG), a coalition of Microsoft Corporation, the Netscape Communications, and Progressive Networks was established to set up the standards for empowering parents to screen inappropriate web content [6], blocking filters, such as CyberPatrol, Internet Filter, NetNanny and SurfWatch have been developed. Content filtering is a multifaceted problem as discussed earlier. We briefly review below some recent work on structured document classification, semi-supervised approach to webpage classification, and one-class classification based techniques.

2.1 Structured Document Classification

Webpage classification is one of the most widely studied problems in web content mining. A lot of text mining approaches have been applied to web mining. However, former typically deals with unstructured documents but the later has the leeway of leveraging structure information. [11] discusses a variety of research done in web content mining. Most of these works treat webpages as a bag of words. Some of them record the meta information including the hyperlink information, URL and position of the words. Some approaches even store the webpage in ontologies. Several machine learning algorithms including modified Naïve Bayes, supervised and unsupervised classification algorithms, rule learning, TFIDF, reinforcement learning, neural networks [25], k-nearest neighbor [12] and association rules have been used for knowledge discovery on web data.

Depending on the application domain, people have exploited structure information differently. In [14], Lin and Ho use the information contained in `TABLE` and `TITLE` tags to find intra-page redundancy by calculating term entropies and come up with informative and redundant content blocks. Approach used by [20] incorporates additional information like location of the web page, type of content that forms majority of the web page, number of paragraphs, images, forms on the page, number of HTML compliance warnings, percentage of readable text and collection of highlights from first ten major elements on the page to automatically generate summary for the visually impaired. Kan [9] advocates "URL-only" technique for classifying webpages with occasional reliance on `TITLE` information. URL for some websites are self-descriptive which can be segmented using information content reduction and title token based finite state transducer. Kan & Thi [10] expanded the work reported in [9] by extracting additional features from URLs. The additional

features are URI components and length features, orthographic features and sequential features. URI components and length features are the URI component such as domain name of a URL token, the length of the URL and the length of a URL component. They use orthographic features of token to correlate unrelated tokens. Sequential orders of URL tokens are expressively extracted as features. Instead of using SVMs, they use maximum entropy algorithm as their learning algorithm.

2.2 Semisupervised Classification

Numerous semisupervised classification algorithms have been applied to the webpage classification problem. This problem domain is favorable for semisupervised machine learning approaches due to various reasons such as sampling bias or prohibitive labeling costs, labeled datasets are often small in size and/or biased. The idea is to use unlabeled data samples which are typically easier to collect at a significantly lower cost. Moreover, unlabeled samples are less likely to be biased [22].

Dempster et al. [4] discusses the use of Expectation Maximization (EM) algorithm to iteratively estimate model parameters and assign soft labels to unlabeled examples by treating the unknown labels as missing data and assuming generative model such as mixture of Gaussians. EM has been widely used in text document classification [19]. Co-training [3] is another popular strategy for learning form unlabeled data if the data can be described in two different sufficient views. Soonthomphisaj et al. [26] proposed a cross-training machine learning algorithm to classify webpages due to the lack of labeled webpages in training corpus. Two simultaneous training algorithms are developed based on Naïve Bayes classifier for heading-based and content-based categorization. Each heading tag (such as `TITLE`, `H1`, `H2`, `H3`, `H4`, `A`, `B`, `U`, `I`) is assigned a weight. They also considered metadata information associated with the page. The transduction approach proposed in [30] assigns labels to an unlabeled dataset by maximizing the classification margins for both labeled as well as unlabeled data. Another semisupervised learning occurs when it is combined with SVMs to form transductive SVM [7]. However, due to convergence to local maxima or violated model assumptions the performance of these techniques could degrade [24].

2.3 One-Class Classification

Typical to porn classification where there is a large number of unlabeled examples, there is another line of research that uses unlabeled data, often referred to as single-class learning or learning from positive and unlabeled data or one-class classification [29]. Here the goal is to predict out-of-sample examples either as belonging to the class or as outliers. A major motivation behind single-class classification is the challenge of collecting unbiased negative data or prohibitive cost of labeling. One approach applies Kernel density estimation to learn the probability density of the labeled data. Another approach uses support vector data description method which learns from the positive examples and artificially generated outliers to separate the positive class from the rest [29]. [15] proposed a partially supervised classification that assumes all the unlabeled data belongs to negative class and then applies EM algorithm to refine the assumption. Another work for classifying user-interesting classes such as "personal homepages" and "call for papers" proposed in [32] uses an SVM-based two-phase classifier to first draw an initial approximation of "strong" negative data from the unlabeled samples by using a weak classifier. Then it iteratively runs an SVM which maximizes the classification margin to progressively improve the approximation of the negative data.

3. INFORMATION SOURCES, DATA AND REPRESENTATIONS

This section presents terms and concepts for better understanding of the approaches we adopt for objectionable webpage classification. First we illustrate the multiple information sources and various advantages of using them, next present the data collection and representation techniques.

3.1 Multiple Information Sources

A web page is different from regular corpora of text documents. A text document can be treated as a bag of words whereas a web page has additional structural information marked within HTML tags. We need to harness this difference for improved classification accuracy [13]. This structural information is also useful in different access scenarios. We divide a webpage into seven different information sources based on the HTML tags. These tags are:

- Webpage URL (URL)
- Anchor and HREF (A)
- Image and ALT (Img)
- TITLE
- METADATA
- BODY
- TABLE

A webpage is parsed and broken into these information sources using regular expressions. Information from these different tags is important in different scenarios. Several webpages contain very little or no textual content, for example, image searches or automatic URL redirection. An instance of image search could look like:

http://images.google.com/images?q=amateur+pussy&hl=en&btnG=Search+Images

In such a case, URL can prove to be helpful in deciding the webpage class. Often webpage URLs are quite descriptive that can be simply viewed as a string of characters. This string can be matched against the words in the profile to find occurrences of adult keywords. These occurrences can be recorded in the form of document-term vector. Data representation is discussed in Section 3.2 in more detail. Other advantage of leveraging URL information is that it avoids fetching the complete webpage and processing its content, which makes it fast. A lot of porn webpages point to other similar webpages through `HREF` tags. These webpages generally have very little textual content. Anchor tags associated with the `HREF` tags, sometimes, contain a small description of the webpages the link is pointing to. In such cases, harnessing `HREF` and Anchor information could be useful. Some

porn webpages have images as links to other similar webpages. In those cases we utilize the URL in `HREF` in exactly the same way as webpage URL is used.
Some porn webpages have a lot of images and almost negligible textual content. In these cases, we utilize information from `ALT` tag. `ALT` tags are associated with `IMG` tags and contains a small description about the image. `IMG` tag has `SRC` attribute which is the URL of the image. This information could be useful in exactly the same way as the webpage URL. Sometimes webpages depend heavily on the use of `TABLE` tags for elegant layout or for summarizing the content. We utilize this information for classifying these webpages. Some webpages mention `META` tag. We specifically use `name='KEYWORDS'` attribute of the `META` tag wherever available. These keywords provide the best description of the webpage, but webpage developers tend to omit this optional tag to escape porn-blockers. Information from `TITLE` and `BODY` tags supplement the above special access scenarios. We propose a model that gives importance to each and every information source.

3.2 Data and Representations

In this section we specifically talk about the challenges in collecting negative instances and using web taxonomies. Then we mention the need for category-profile and its creation. We also talk about two different representation techniques of the data in terms of the profiles.

3.2.1 Negative Instance Collection

The problem of web content categorization is a binary classification problem and involves two categories: a target category and its complementary non-target category. In our case, a target category is one that contains objectionable websites and non-target category refers to the category that includes all other websites which do not contain objectionable material. It can be easily perceived that a very small subset of all the webpages currently on internet belong to objectionable (target) category and rest of the webpages belong to non-objectionable (non-target) category. Unbiased negative data collection in such a domain becomes a major challenge. As discussed in Section 2.3, these issues in unbiased negative data collection forms the motivation of single-class classification, which treats all the unlabeled data as negative instances and then applies EM algorithm to refine the assumption. This repetitive application of EM algorithm degrades the response time of such an approach.
We propose the use of web taxonomy in building the non-target category. Web taxonomies like Google or Yahoo are a rich source of accurately labeled web data. They organize the websites into categories by pre-defined topics in a hierarchical fashion. We use Google taxonomy in our approach. We pick websites for non-target category by selecting websites from categories or sub-categories that do not belong to target category. In Google taxonomy, the objectionable websites are a part of 'Society → Gay, Lesbian and Bisexual', 'Society → Sexuality' and 'Society → Transgendered'. So we collect websites for non-target category from 'Business', 'News', 'Recreation' and 'Sports'. Instances from these categories cover a wide range of negative data. By picking websites from these topics as substitute for non-target category, we avoid the difficulties in collecting unbiased negative data or prohibitive cost to label negative instances from unlabeled webpages, using EM algorithm iteratively.

3.2.2 Profile Creation

A text categorization problem typically represents each document as a vector of terms. These terms are carefully selected using known feature selection algorithms to avoid noisy features and improve accuracy of classification [23][31]. Webpage classification is no different in terms of selecting good features or words to improve classification accuracy. We use a modified information retrieval based technique to rank these features or terms.
We adopted $TFIDF$ in category perspective instead of document perspective. We have $TFICF$, where TF refers to frequency of term, T_k in category C_i and ICF refers to inverse category frequency of the term, T_k. For our case, how much a term is important to a document loses its significance. We are interested in rather which term is more important in recognizing a category. Each category is considered a big document formed by virtually combining all the documents within that category. Hence, we defined our method as,

$$TFICF(T_k, C_i) = TF(T_k, C_i) \times ICF(T_k),$$
$$ICF(T_k) = \log(|C|/CF(T_k))$$

where, $|C|$=Number of categories in the collection,
$CF(T_k)$=Category frequency for term T_k.
We first generate a dictionary of terms for each category using the training webpages. Then $TFICF$ measure is used to rank the terms in the dictionary for each category. Finally, top-k terms are selected. These top-k terms form the profile for that category. Hence, we build a profile for each category. We discuss later in this section how this profile is used to calculate the density vector representation of webpages.
Before applying $TFICF$ all stop-words are removed from dictionary of each category. Stemming [2] further reduces the dimensionality. By stemming we mean that only word roots are kept and the word root covers all the ramifications to that. For example, "sex" represents "sexy", "sexual", "sexuality". Lot of words on these websites are intentionally misspelled to cheat keyword-based porn filters. For instance, "sexy" is sometimes spelled as "sexie" or "sex" is sometimes spelled as "sexxx". All these cases can easily escape a naïve word filter but cannot skirt stemming approach.

3.2.3 Document-Term Based Representation

Once a webpage is broken according to different tags, the problem of webpage classification is transformed to the problem of text classification. Webpage is then typically represented as multiple document-term vectors one for each information source. Words in the profile constitute the terms or features. The values in these vectors are occurrence frequency of those words in the corresponding information source. More formally, consider

- a webpage, ω, with seven specified information sources $\sigma_1, \sigma_2, \sigma_3, ..., \sigma_7$,
- a profile, π that has m terms $\{\tau_1, \tau_2, \tau_3, ..., \tau_m\}$,

then a webpage is represented as seven vectors $\nu_1, \nu_2, \nu_3, ..., \nu_7$ each with $1 \times m$ dimension. Each vector ν_p corresponds to information source σ_p respectively, for $1 \leq p \leq 7$. $\nu_i[j]$ represents the occurrence frequency of term, τ_j in information source, σ_i for webpage, ω. For several webpages, we construct seven document-term matrices for each information source instead of vectors. The document-term matrices

are represented as $\nu_i[k][j]$ which denotes the occurrence frequency of term, τ_j in information source, σ_i for webpage, ω_k. These are $n \times m$ matrices for n webpages.

3.2.4 Density Based Representation

We also represent each webpage in terms of density vectors for each information source. Density vector is denoted as $\rho_i[k]$, signifying the density of profile (π) terms (τ_j) in the block of ω_k marked by σ_i. In the process of matrix creation we record the number of words (excluding the stop words) that appear in the blocks marked by each information source on a webpage in a vector, $\delta_i (1 \leq i \leq 7)$. $\rho_i[k]$ represents the density of profile terms for ω_k in the block marked by σ_i. Density is then given by the following formula:

$$\rho_i[k] = \frac{\sum_{j=1}^{m} \nu_i[k][j]}{\delta_i[k]}$$

For the adult profile, we refer to this density value as the porn-index of the webpage for the corresponding information source. In this section, we discussed how to represent various information sources from a webpage as document-terms and density vectors after building the profiles. This step paves the way for the approaches we present in the next section and could be considered as a data preprocessing step often essential to a data mining application.

4. APPROACHES

In this section, we present approaches based on the data representations discussed in the previous section. SVMs are known to perform well on text categorization [8]. We use SVMs and the document-term matrix representation in the first approach. We call this baseline SVM model, discussed in Section 4.1. Next we propose two novel approaches based on our density vector representation: one is density-based SVM model discussed in Section 4.2; and *the other is an intuitive approach* that uses thresholds for the density values in classifying webpages, called *the density threshold model* elaborated in Section 4.3.

4.1 Baseline SVM Model

Once webpages are converted to document-term matrices webpage classification task is similar to text classification. The ability of Support Vector Machines (SVM) to separate a set of positive data from a set of negative data, that may or may not be linearly separable, with maximum margin makes it a good text classification model [8]. Text classification datasets are usually high-dimensional and sparse. These data points are often not linearly separable and hence difficult for linear classifiers to make predictions accurately. This makes a good application ground for Gaussian kernel based SVM. Although linear kernels are fast but generally gaussian kernels perform better in terms of accuracy [28].

After webpages are decomposed into blocks of information sources marked by different tags and represented in different matrices, SVM models are learned on these document-term matrices. This generates seven different and independent SVM models. Each of these models could be used to predict against test webpages, hence giving seven decision values for a test webpage. We call them sub-decisions. There could be several schemes to combine these sub-decisions and come up with a universal decision. We employ a majority voting scheme with equal weights assigned to each sub-decision: if four out of seven sub-decisions classify a webpage as porn then we declare it as porn.

There could be other schemes to combine the sub-decisions from different classification models. One possible scheme is so called "OR" scheme. In the "OR" scheme, all the classification models work independently to categorize a webpage. As long as one of the classification models classify the webpage as objectionable, it is objectionable. This approach is simple, but the drawback is that it may cause an increased false positive rate. Another possible scheme could be assigning different weights to sub-decisions, but again these weights have to be learned carefully without overfitting the training samples.

We use this model for both binary classification as well as multi-classification task. Since the negative training data is large and more varied than the positive training data, we try to avoid any bias due to the skewed data distribution by considering categories in negative training data. As mentioned earlier we consider four categories in the negative data (i.e., Business, News, Recreation and Sports) and Adult category for the positive data, which makes a total of five categories. An equal number of webpages is used from each category, for multi-classification tasks. Here we do not focus on how correctly the classification is made within the sub-categories of the non-porn (or negative) class, hence classification accuracy is calculated in a different way. We discuss about evaluation parameters in Section 5. For a binary classification task, we consider just two categories porn and non-porn. For the porn category, training and testing data is selected from Adult category. For the non-porn category, data is selected from the mix of Business, News, Recreation and Sports categories. The number of data instances selected for porn and non-porn categories in binary classification case is equal, whereas, the number of instances in non-porn category is four times the number of porn data instances in a multi-classification task.

4.2 Density-Based SVM Model

In the previous baseline SVM model, we considered webpages as document term matrices similar to traditional text categorization approach. Usually text documents have a rich textual content and learning from document-term matrix constructed from these text documents results in impressive accuracy results. However, porn webpages contain a lot of images but very limited text. A document-term matrix constructed in such a scenario could be very sparse which is not good for learning SVM classifier. Also, these document-term matrices are high-dimensional. High-dimensionality and sparsity degrades the classification performance. Using feature reduction/selection techniques before learning a classifier does not quite improve classification accuracy specially in porn webpage domain. The reason is feature selection/reduction finds features relevant to the training data which may not be present in the test data at all. This is often true for porn webpages due to very little textual content available on the webpage. It is possible that the intersection of vocabulary of training webpages with the vocabulary of test webpages be null.

So it is desirable to represent webpages in a way that avoids problems due to sparsity and high-dimensionality while capturing the essence of relevant terms. We calculate density of porn words for each section of webpage marked by corresponding information source. To calculate density, we sum

the number of times, various words from porn-profile occur in that section and divide it by the total number of words in that section. Note that stop-words are excluded from density computation. We explain mathematically, how these density vectors are computed in Section 3.2. In the baseline SVM model, each webpage was represented as a set of seven vectors with m dimensions (terms). Now each m-term vector is transformed into a density value using the formula in Section 3.2. A webpage is now represented as a vector of seven density values, one for each information source instead of a set of seven term-vectors. This transformation reduces seven document-term matrices to a matrix of density vectors. We train SVM classifier on this density matrix.

Unlike the previous model, where we trained seven different and independent SVM classifiers, in this approach we train a single SVM model on the density vectors. As a result instead of obtaining seven sub-decision values and coming up with a universal decision, density based SVM model generates a single decision for each test webpage.

4.3 Density Threshold Model

In this section we present the most intuitive approach to classify webpages based on different information sources. After webpages are decomposed into blocks according to different information sources, we compute density of porn words in these blocks of webpages. To calculate density, we sum the number of times, various words from porn-profile occur in that block and divide it by the total number of words in that block. Note that we exclude stop-words from our density computation. We explain mathematically how these density vectors are computed in Section 3.2. The "first-attempt" type of approach would compute these density scores for each of the information sources from all the webpages for both porn and non-porn data. Then a suitable threshold is learned from training data for each of the information sources for classifying porn and non-porn with minimum false-positives and false-negatives for training data.

There will be seven such threshold values: one for each information source. For a test webpage we again calculate the seven density values. We compare these density values with the thresholds learned from the training set. Based on these comparisons, the test webpage will have seven decision values one for each information source. We call them sub-decisions. Now we combine these sub-decisions to come up with a universal decision. First, we use "OR" approach, where each sub-decision is made independently: the test webpage is porn if any one of the thresholds is exceeded. This approach is simple but may cause an increased false positive rate. This creates another problem of "over-blocking" which in this domain is more expensive. We also use a majority voting approach with equal weights for each sub-decision. Here, if four out of seven sub-decisions classify a webpage as porn, we declare it as porn. The advantage with this scheme is the reduced number of false-positives.

5. EXPERIMENTS

We conducted experiments to evaluate the density threshold model, the baseline SVM model, and the density-based SVM model. The dataset of positive instances is obtained from an internet company. It includes a set of 100 teenagers' anonymous access attempts to porn webpages. These access attempts include image and keyword searches along with other direct porn webpage access attempts. These instances constitute only the positive data. Negative instances are collected using the Google taxonomy. We collected negative instances from four classes: Business, News, Recreation, and Sports. 100 instances were collected from each of these classes. In the experiments we compute classification accuracy and use it as evaluation measure. In calculating classification accuracy, we do not focus on how correctly the test instances among negative classes have been classified. The goal is to classify porn and non-porn instances correctly. We record false positives (FP) - instances that are misclassified as porn, and false negatives (FN) - instances that are misclassified as non-porn between positive and negative classes and define classification accuracy as:

$$Accuracy = 1 - \frac{FP + FN}{TotalInstances}$$

We compute classification accuracy for all the three approaches described in the previous section. In the following, we first present results for the baseline SVM model, then give results for the density-based SVM model, next provide results of the density threshold model, and summarize this section with a comparative study of these approaches.

Classes	Adult	Business	News	Recreation	Sports
Training	90	90	90	90	90
Testing	10	10	10	10	10

Table 1: Number of instances in different categories for experiments in Table 2

	k=500 (2500 terms)	k=1000 (5000 terms)
Anchor (A)	76.2 ± 1.6 %	75.4 ± 1.8 %
Image (IMG)	73.8 ± 0.8 %	71.2 ± 1.2 %
Metadata	49.2 ± 2.6 %	50.6 ± 3.2 %
Body	58.2 ± 1.4 %	56.3 ± 2.5 %
Table	60.4 ± 1.8 %	56.4 ± 2.4 %
Title	59.7 ± 3.5 %	55.7 ± 4.3 %
URL	76.6 ± 3.2 %	72.6 ± 3.8 %
Combined Accuracy (Voting)	79.6 ± 1.4 %	76.2 ± 2.4 %

Table 2: Accuracy results for k=500 and k=1000 terms per profile

5.1 Results for Baseline SVM Model

As discussed in Section 4.1, in the baseline SVM model, webpages are represented as a set of seven document-term matrices. We build a profile for each class, as explained in Section 3.2.2. Each profile contains top-k terms where the terms are selected using a $TFICF$ score, explained in Section 3.2.2. In the first set of experiments, we study two values for k: 500 and 1000 terms per profile, one profile corresponding to one of the 5 classes. This results in two datasets, one with a total of 2500 (5×500) terms and the other with a total of 5000 (5×1000) terms. We use 90 documents from each of the five categories for training the SVM and remaining 10 for testing as depicted in Table 1. We use C-SVC with Radial Basis Function (RBF). RBF is experimentally found to be the best text categorization kernel function [32].

In Table 2 we present accuracy results for different k (500

Classes	Adult	Business+News+Recreation+Sports
Training	90	90
Testing	10	10

Table 3: Number of instances in porn and non-porn categories for experiments in Tables 4, 5, 6, 7 and 8

	Equal porn and non-porn instances
Anchor (A)	79.75 ± 1.75 %
Image (IMG)	80.5 ± 2.0 %
Metadata	73.75 ± 2.75 %
Body	65.25 ± 2.25 %
Table	81.25 ± 1.75 %
Title	75.75 ± 1.75 %
URL	80.25 ± 3.25 %
Combined Accuracy (Voting)	83.25 ± 1.75 %

Table 4: Accuracy results for balanced data distribution

and 1000). Here classification accuracy for each information source is calculated using 10-fold cross validation. Finally, we combine these accuracy values using a majority voting based scheme and report the accuracy values. From the results it is obvious that increasing the number of terms (features) per profile does not necessarily improve the accuracy. The drop in the accuracy value is due to noisy features. Increasing the term size for the profile can result in the inclusion of many irrelevant terms. This increases the number of false positives and false negatives, and hence accuracy decreases [5; 27]. Thus we perform the subsequent experiments using k=500 terms per profile. Another interesting observation from Table 2 is that Anchor and Image have the highest accuracy values among all the information sources, which tells us that porn webpages contain rich information of Anchor and Image which could be utilized to our advantage.

In the previous experiments we had an unbalanced number of positive and negative training and testing instances. The inequality in the number of positive and negative documents introduces skewness in data representation and might also distort accuracy calculation. In the subsequent experiments, we dissolve the notion of five categories, although we still maintain the profiles constructed from these categories. We consider two categories: porn and non-porn, hence reducing the multi-class classification problem to a binary classification task. However, we keep 100 documents from the porn category and 100 documents randomly selected, without replacement, from the negative classes of Business, News, Recreation and Sports. Again 90 documents are kept for training and 10 for testing. Data distribution for this experiment is shown in Table 3. Classification accuracy results for non-skewed data are presented in Table 4. Accuracy values for each information source is calculated using 10-fold cross validation. Again, the majority voting scheme is used to combine the accuracy results. Comparing the results of Table 2 and Table 4, we can observe that learning from a non-skewed data distribution performs better than learning from an unbalanced data distribution.

Since we focus more on correctly predicting the positive instances than on the negative ones, we use a single adult profile to represent both positive and negative instances. We prefer to eliminate the need for maintaining the set of negative features by representing both positive and negative instances in terms of adult profile. The basic intuition behind this representation is: to learn a classifier for a class, focus should be on the features that best represent the instances that belong to the class. Positive instances are least expressed in terms of negative features and best represented by positive features. Positive (negative) features refer to the features or terms used to represent positive (negative) instances. In other words, terms in the adult profile denote the set of positive features, and terms in the rest of the profiles denote the set of negative features.

	Single Adult Profile
Anchor (A)	83.25 ± 1.75 %
Image (IMG)	84.25 ± 0.75 %
Metadata	70.0 ± 2.0 %
Body	63.75 ± 1.25 %
Table	83.25 ± 2.25 %
Title	75.5 ± 2.5 %
URL	78.75 ± 2.75 %
Combined Accuracy (Voting)	84.25 ± 1.75 %

Table 5: Accuracy results for single adult profile

Next we experiment with this intuition in our attempt to eliminate the need for multiple profiles. Data distribution for this experiment is similar to the experiment in Table 4 and is described in Table 3. Both porn and non-porn instances are represented by a single adult profile of 500 terms, i.e., only positive features are kept. We train C-SVC on seven document-term matrices (one for each information source) constructed from these webpages. Classification accuracy values of 10-fold cross validation are computed. Majority voting scheme is used to combine accuracy scores of different information sources. Experiment results are shown in Table 5. Classification accuracy results are almost similar to those of Table 4. Therefore, we need not construct the other four negative class profiles. We just keep the profile for the adult class in the subsequent experiments.

Finally, we perform experiments to validate whether leveraging structural information can improve classification accuracy over the traditional "bag-of-words" approach for text categorization. In the "bag-of-words" approach, webpages are treated as text documents. We represent a webpage as a single document-term matrix as opposed to seven matrices when different information sources are considered. We consider two scenarios for this experiment, one where the data distribution is balanced (since we already know that classification accuracy is poor in the skewed data distribution case) and the other scenario is the one where we consider a single adult profile, which performs best so far. Classification accuracy is 54.75 ± 2.25 % in the non-skewed (balanced) data distribution scenario and 55.25 ± 3.75 % in the single adult profile scenario. Thus, we conclude the structural information is certainly helpful. The reason for poor accuracy is that individual importance is not given to each information source. The "bag-of-words" approach averages out the information from these different sources, resulting in higher misclassifications. Classification accuracy results for

	Non-Skewed Data Distribution	Single Adult Profile
Classification Accuracy	54.75 ± 2.25 %	55.25 ± 3.75 %

Table 6: Accuracy results for "bag-of-words" approach

the "bag-of-words" approach are summarized in Table 6.

5.2 Results for Density-based SVM Model

As discussed in Section 4.2, we use the density measure as defined in Section 3.2 to train an SVM in the Density-based SVM Model. In this approach we do not represent the webpages in seven document-term matrices, but construct one matrix with seven dimensions, each dimension represents the density value for an information source. We use the Adult profile to calculate density values of the webpage both from porn and non-porn category for each information source. We refer to these density values as "porn-indices". Each webpage has seven porn-indices, one for each information source. In the experiments we use the data distribution as in Table 3. C-SVC was used and 10-fold cross validation was performed to calculate classification accuracy. Since we have just one matrix to represent webpages, there is a single accuracy value as opposed to seven values in the baseline SVM Model. Thus we do not need a merging scheme like majority voting to calculate the final accuracy value. We experiment with this approach in three scenarios:

Scenario 1 (URL Only) : In this scenario, we test the approach using only the URL feature. The porn-index of webpage URL is calculated (Section *3.2.4*) and C-SVC is trained on the training webpages. The classification accuracy value is 83.5 ± 1.0 %. This scenario is designed for the cases where only webpage URLs are available. Sometimes porn webpages have very little text and most of the times this scarce text is insufficient in classifying a webpage as porn. Thus we have to rely on webpage URLs for classification. The drawback is, however, URLs often contain much information and could sometimes be misleading, resulting in high false-positives. Advantage of using only URLs is that we avoid the need for downloading the webpage contents and parsing them. This can greatly reduce the response time.

Scenario 2 (Content Only) : In this scenario, we consider the content of the webpage and leave out the webpage URL. This is necessary because sometimes the webpage URL is not at all helpful. The objective is to observe how the model performs when webpage URLs are not available. Porn-indices of webpages are calculated based on Anchor, Image, Body, Metadata, Table, and Title. C-SVC is learned on this six dimensional density matrix using the training webpages. The classification accuracy is 91.5 ± 1.5 %. Considering the content of the webpage certainly gives more information about the webpage. Having more information about the webpage reduces the chances for false-positives and improves accuracy.

Scenario 3 (URL and Content) : In this scenario , we combine both 'URL' and 'Content' information from the previous scenarios to observe if we could further improve classification accuracy. In this setup, we calculate porn-indices of a webpage based on all seven information sources and train C-SVC on training webpages. The classification accuracy is 93.5 ± 1.0 %. It is evident that webpage URLs and contents complement each other to achieve better classification accuracy.

	Classification Accuracy
URL only	83.5 ± 1.0 %
Content only	91.5 ± 1.5 %
Combined Accuracy	93.5 ± 1.0 %

Table 7: Accuracy results for three different scenarios using the density-based SVM model

The results of these scenarios are summarized in Table 7. The results indicate that removing sparsity and high dimensionality can significantly improve the accuracy values as discussed in Section 4.2.

5.3 Results for Density Threshold Model

In this section we present the experimental results for the Density Threshold Model introduced in Section 4.3. This is an intuitive approach, where density thresholds are learned from the training webpages after they are represented in density vectors. We discuss in Section 3.2 about density vector representation of webpages. We use the Adult profile to calculate the density values of webpages. For each information source, the threshold for a density value is computed to minimize the number of false-positives and that of false-negatives in the training instances. This generates seven density threshold values. Similarly, test instances are also represented as density vectors. Based on the density threshold values learned from the training instances, test instances are classified as porn or non-porn. For each test webpage, we obtain seven sub-decisions according to the threshold values. Finally, we use the "OR" technique and majority voting technique to compute a final classification.

We use the data distribution shown in Table 3 in the experiments, using 90 documents for training and 10 documents for testing from each porn and non-porn pair. Experimental results are summarized in Table 8. Results for the "OR" and "majority voting" schemes are also presented. We observe that combined accuracy based on the "OR" scheme is smaller than the accuracy obtained using the majority voting scheme. A closer examination shows that the "OR" scheme generates more false-positives than the majority voting scheme does. Comparing these results with the best results from the baseline SVM model (Table 5), we notice that the density threshold model works as well as the baseline SVM model, although the density threshold model is a linear classifier. This shows that the density representation of webpages (vs. the document-term matrices) indeed helps in reducing sparsity and high-dimensionality, hence improving accuracy.

5.4 Summary

We compare the experimental results for different approaches and present a summarized recommendation for the best model. We divide our experiments into two parts. In the first part, we conducted experiments to find the best values for different parameters involved in the approaches. Experiment results in Table 2 suggest the best value for k (= 500), the

	Density Threshold Model
Anchor (A)	88.5 ± 2.25 %
Image (IMG)	64.25 ± 3.25 %
Metadata	63.75 ± 4.75 %
Body	60.5 ± 3.75 %
Table	74.5 ± 2.75 %
Title	83.25 ± 2.25 %
URL	84.5 ± 3.75 %
Combined Accuracy (OR)	81.5 ± 2.75 %
Combined Accuracy (Voting)	84.0 ± 2.25 %

Table 8: Accuracy results for density threshold model

Approach	Accuracy
Bag-of-Words	55.25 ± 3.75 %
Baseline SVM Model	84.25 ± 1.75 %
Density Threshold Model	84.0 ± 2.25 %
Density SVM Model	93.5 ± 1.0 %

Table 9: Comparison of Accuracy Results for Different Approaches

number of terms per profile. Comparison between Table 4 and Table 2 indicates that learning from an equal number of instances in positive and negative class performs better than otherwise. Table 5 shows that learning from a single adult profile performs better than learning from multiple profiles. We use these parameter values throughout the second part of experiments, where we compared different approaches. We compared accuracy results for our proposed approaches, the density-based SVM model and the density threshold model, with the Baseline SVM model. We also compared these approaches with a naïve webpage classification approach that does not consider the webpage's structure. We called this the "bag-of-words" approach. These results are summarized in Table 9. The results show that the "bag-of-words" approach performs worst. We obtain an evident performance gain by leveraging the multiple information sources observed in the baseline SVM model. Though the density threshold model is a linear classifier, it performs as well as the baseline SVM model due to its density-based representation. The density-based SVM model performs best among all the approaches, reason being the utilization of multiple information sources, the succinct density-based representation, and the superior performance of a SVM classifier.

6. IMPACT AND SIGNIFICANCE

The controlling of children's online activities has become a pressing and important topic in both research and practice due to a sequence of nationwide cases involving adult sexual predators using virtual-communities on the Internet using an unprecedented number of means to prey on innocent and credulous child victims. With the emergence of Web 2.0, the web becomes more open. On one hand, the predators exploit the openness of the web and intensify evil activities with disguises; on the other hand, the young web browsers devour anything of remote interest and are susceptible to falling in numerous hidden and disguised traps. Therefore, it is imperative for Web service providers to better police the Web, and protect our intriguing young Web surfers by orchestrating relentless counterattacks to automatically block the deleterious and preying webpages and to promote education-orientated, mind-opening, conducive websites. This necessitates highly accurate Web filtering services. The work presented in this paper represents one of the efforts of current research and practice. An accurate filter with a learning component can not only block the objectionable Web contents, but also adapt to the dynamically changing Web and learn to improve its performance.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we study the problem of blocking objectional web contents aiming to make the Internet kid-safer. We utilize multiple information sources and show that treating them independently for classification improves accuracy. We propose a robust and compact representation of data in terms of density vectors to eliminate high-dimensionality and data sparsity, and verify experimentally that the density vector representation improves classification accuracy. We propose the use of web taxonomies for negative instance collection to eliminate data imbalance and prohibitive costs for labeling. We propose a density threshold model and a density-based SVM model to classify objectionable content combining the above contributions, and demonstrate a significant improvement over the baseline SVM model. We also study the applicability and impact of our system in an internet company.

The work presented in this paper opens up many opportunities for further research. Cost-sensitive learning could be helpful in porn filtering. For example, we can fine-tune to further reduce "over-blocking" by assigning higher costs to false-positive errors; or higher costs are assigned to false-negative errors in order to realize an "extremely-strict" porn filter. We could learn weights for different information sources based on their contributions in accurate classification to further improve the current system's performance. We could also include link-graphs as one of the information sources. Intuitively, if a webpage points to several porn webpages, then it is possibly a porn webpage. This work could be extended to finding "Similar Webpages" over the Internet that has potential applications in indexing and making targeted Web search more efficient.

8. REFERENCES

[1] N. Agarwal, E. Haque, H. Liu, and L. Parsons. A subspace clustering framework for research group collaboration. *International Journal of Information Technology and Web Engineering*, 1(1):35–58, January-March 2006.

[2] S. Ahmed and F. Mithun. Word stemming to enhance spam filtering. In *In Proceedings of Conference on Email and Anti-Spam (CEAS 2004)*, 2004.

[3] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of COLT'98*, pages 92–100, 1998.

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Roy.Statistical Society (B)*, 39:1–38, 1977.

[5] G. Forman. An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.*, 3:1289–1305, 2003.

[6] S. Y. Ho and S. M. Lui. Exploring the factors affecting internet content filters acceptance. *SIGecom Exch.*, 4(1):29–36, 2003.

[7] T. Joachims. Transductive inference for text classification using support vector machines. In *Proccedings of 16th International Conference on Machine Learning (ICML'00)*, pages 200–209, 1999.

[8] T. Joachims. A statistical learning model of text classification for support vector machines. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 128–136, New York, NY, USA, 2001. ACM Press.

[9] M. Y. Kan. Webpage classification without the web page. In *Proceedings of the 13th International World Wide Web Conference*, pages 262–263, 2004.

[10] M. Y. Kan and H. O. N. Thi. Fast webpage classification using url features. In *Proceedings of the conference on Information and Knowledge Management*, 2005.

[11] R. Kosala and H. Blockeel. Web mining research: A survey. *ACM SIGKDD Explorations Newsletter*, 2(1):1–15, 2000.

[12] O. W. Kwon and J. H. Lee. Text categorization based on k-nearest neighbor approach for web site classification. *Information Processing and Management: an International Journal*, 39(1):25–44, 2003.

[13] C. H. Lee, M. Y. Kan, and S. Lai. Stylistic and lexical co-training for web block classification. In *Proceedings of the 6th annual ACM international workshop on Web information and data management (WIDM 04)*, pages 136–143, 2004.

[14] S. H. Lin and J. M. Ho. Discovering informative content blocks from web documents. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 588–593, 2002.

[15] B. Liu, Y. Dai, X. Li, W. S. Lee, and P. S. Yu. Building text classifiers using positive and unlabeled examples. In *Proceedings of the Third IEEE International Conference on Data Mining (ICDM-03)*, page 179, 2003.

[16] H. Liu and H. Motoda. *Feature Extraction, Construction and Selection: A Data Mining Perspective*. Boston: Kluwer Academic Publishers, 1998. 2nd Printing, 2001.

[17] H. Liu and H. Motoda. *Feature Selection for Knowledge Discovery & Data Mining*. Boston: Kluwer Academic Publishers, 1998.

[18] H. Liu and L. Yu. Toward integrating feature selection algorithms for classification and clustering. *IEEE Trans. on Knowledge and Data Engineering*, 17(4):491–502, 2005.

[19] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134, 2000.

[20] P. Parente. Audio enriched links: web page previews for blind users. In *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS 04)*, pages 2–8, 2004.

[21] L. Parson, E. Haque, and H. Liu. Subspace clustering for high dimensional data - a review. *SIGKDD Explorations*, 6(1):90–105, 2004.

[22] K. Peng, S. Vucetic, B. Han, H. Xie, and Z. Obradovic. Exploiting unlabeled data for improving accuracy of predictive data mining. In *Proceedings of the Third IEEE International Conference on Data Mining (ICDM)*, pages 267–275, 2003.

[23] G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. Technical Report 87-881, Department of Computer Science, Cornell University, 1987.

[24] M. Seeger. Learning with labeled and unlabeled data. Technical report, Institute for Adaptive and Neural Computation, University of Edinburgh, 2001.

[25] A. Selamat and S. Omatu. Web page feature selection and classification using neural networks. *Information SciencesInformatics and Computer Science: An International Journal*, 158(1):69–88, 2004.

[26] N. Soonthomphisaj, P. Chartbanchachai, T. Pratheeptham, and B. Kijsiriku. Web page categorization using hierarchical heading structure. In *Proceedings of the 24th International Conference on Information Technology Interfaces (ITI 02)*, pages 37–42, 2002.

[27] L. Tang and H. Liu. Bias analysis in text classification for highly skewed data. In *ICDM'05*.

[28] D. M. J. Tax and R. P. W. Duin. Support vector domain description. In *Pattern Recognition Letters*, volume 20, pages 1991–1999. 1999.

[29] D. M. J. Tax and R. P. W. Duin. Uniform object generation for optimizing one-class classifiers. *Journal of Machine Learning Research, Special Issue on Kernel Methods*, 2(2):155–173, 2002.

[30] V. N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[31] Y. Yang and J. Pedersen. A comparative study on feature set selection in text categorization. In *Proceedings of the 14th International Conference on Machine Learning*, pages 412– 420, Nashville, TN, 1997. Morgan Kaufmann.

[32] H. Yu, J. Han, and K. C. Chang. *PEBL*: Web page classification without negative examples. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 16(1):70–81, 2004.