Using Support Vector Machines to classify unwanted content on the internet

Vincent Slot, 1691716, v.r.slot@ai.rug.nl,
Marco Wiering[*], Markus Jelsma[†]

Every day it is getting harder to find the correct information on the internet. The amount of web pages has taken on huge proportions, for private use, but also for businesses large and small. For this reason companies might choose to deploy a custom made search engine. In this way they can keep the content on their servers easily accessible for their clients. To keep the search engine database as small as possible we used a machine learning tool based on Support Vector Machines to filter out any unwanted content that is considered to be entered in the database. This filter will increase the efficiency of processing search queries on the servers containing the databases. In this paper the algorithm will be explained and evaluated to decide whether or not it satisfies the demands and standards of today's commercial search engine providers.

1 Introduction

1.1 The Assignment

OpenIndex, a company located in Groningen, provides custom made search engines for other companies, like site search engines. They have been in business since 2010, and connected their first client in 2011. Their clusters download 1-2 GB/s. Most of this data is thrown away, but the data they keep on their servers add up to about 1.5 TB, with information about approximately 70 million hosts.

A search engine generally works as follows. The user types a query in the search bar, the query is sent to the host of the search engine, the servers perform an off-line search and finally return the results. In order for this process to be efficient the database on the servers has to be indexed, and as small, yet complete, as possible. The database on the servers is gathered by a virtual agent called a 'crawler', or 'spider'. This digital entity, as the name suggests, crawls the web and records all the data it finds, before the data is added to the database. A logical consequence of this activity is that you need a lot of server space. Furthermore, there is a lot of useless data. Server space is expensive and the more you use, the less efficient your search engine becomes. Searching more data means taking more time.

It is evident that the amount of information gathered by the crawler must be reduced somehow. The solution to this problem lies in the filtering of the data. To achieve this, we need to be able to make a distinction between wanted and unwanted data. For this project we limited ourself by defining unwanted content as pornography. There are two major reasons for this. Firstly, pornography is relatively easy to classify, and secondly, there is a lot of porn on the internet (or so I've been told).

1.2 Goals

OpenIndex asked us to provide them with a pornography filter that meets a number of requirements.

- The classifier has to be able to classify quickly and reliably.
- The classifier should definitely not filter out any non-pornographic documents.
- The classifier should return a probability of something being pornographic, not just a binary result.
- The classifier has to run on a web server, making it easily accessible for multiple applications within the company.
- The classifier has to run as a plugin for Nutch, the crawler used by OpenIndex.

[*] Rijksuniversiteit Groningen, Department of Artificial Intelligence
[†] OpenIndex, http://www.openindex.io

2 Methods

2.1 Previous research

The concept of content filtering is rather well-documented in the form of porn filtering, but even better in the form of Spam filtering [Jezek and Hynek, 2007]. Spam is a similar problem, and has more or less the same solution. These problems are similar because both need a classification based on nothing but natural language. Websites and email often contain little information besides the main text body about the 'appropriateness', and thus about whether the document is wanted by its user or not.

Looking at several ways porn classifiers have been built, we can see that various methods have been tried and tested. Good results in text-based porn or Spam classification have been achieved with neural networks [Selamat et al., 2011] and Support Vector Machines [Amayri and Bouguila, 2010, Drucker et al., 1999]. In porn classification based on image analysis good results have been booked with Support Vector Machines [Wang et al., 2009].

We will be basing our classifier just on text. In that category Support Vector Machines (SVM) get very good results, so that is the classifier we will be using.

2.2 Support Vector Machines

The information will be categorized by a linear SVM algorithm. SVM classifiers are based on supervised machine learning. The algorithm will calculate the probability that a document is 'porno' and should be labeled accordingly [Hu et al., 2007]. The threshold of the probability for labeling documents as 'porno' can be fine-tuned so that we will get optimal results. It is important that we get the best results in classifying pornography, but it is even more important that we leave the rest of the data intact.

SVM is a popular classifier for problems like text classification for three main reasons [Diao et al., 2012]:

1. They offer relatively good results on unbalanced data sets in high-dimensional problems.

2. There is a theoretical guarantee of finding a global optimal solution.

3. There is a reasonable trade off between time and performance.

The usage of supervised learning algorithms consists of two phases. First the algorithm needs to be trained, then the algorithm can be tested (or deployed in practice). The training phase consists of offering beforehand labeled examples to the algorithm. The flat text in these documents can be seen as numerical feature vectors, where the amount of appearances any word makes in the document is a feature. These multidimensional feature vectors can be separated by a hyperplane such that the maximum number of training samples fall into the right category. Training the algorithm means finding the hyperplane that has the maximum margin on both sides. Instead of using all the training samples to decide this boundary on, SVM decides it by using just the input vectors that lie closest to the boundary. Hence, an SVM-model is just a set of support vectors, instead of all the input vectors.

Once the classifier has been trained we will proceed to the next phase: testing. In this phase the algorithm will be tested on its performance. The data used in this phase also will need to be labeled in advance as pornography or regular. The algorithm then will classify the data, and will check whether the web page was correctly classified or misclassified. Based on these results we will get a percentage of correctly classified pornographic documents, a percentage of correctly classified regular documents, and an overall score of the correctly classified documents.

2.3 The Algorithms

The algorithms used with the classifiers were based on previously constructed email classifiers [Znaor and Elzinga, 2011] using Java libSVM libraries [Chang and Lin, 2011]. The algorithm we used for the measuring performances of the classifier for these experiments is slightly different from the algorithm that OpenIndex will use. In this section we will briefly illustrate the differences between the two.

2.3.1 The experimental algorithm

The experimental algorithm uses either cross validation on a sizable data set to randomly create test sets and training sets, or loads both sets separately from a file system. The classifier both trains and tests on labeled data, meaning that we will be able to assess its performance. This is necessary to optimize the algorithm and achieve the best performance.

This algorithm is able to create models and save them after the training phase, or skip the training phase by loading an already existing model.

2.3.2 The applied algorithm

The algorithm was stripped down to be applied on Nutch, the crawler. Once initialized this algorithm will load a model. There is no data set needed. A single document will be passed on to the classifier by Nutch and the algorithm will return a probability of the document being pornography.

The second part of the software we supplied OpenIndex with consists of a stripped down algorithm to build SVM models from data sets. This enables them to create additional models for different languages if they need to.

2.4 The Data

We used three ways to gather the data set needed for the classifier. In this section all three methods will be briefly explained.

2.4.1 Manual classification of random documents

The first approach to gather data was to retrieve a huge chunk of random websites from the OpenIndex database and manually classify the documents as porn or regular. The data this yielded was varying in subject and intensity. This is a very good thing for a machine learning algorithm like SVM. The more varied the data is, the better the classifier will perform on real test samples on the internet. The downside to this is that it takes enormous amounts of searching for a relatively small data set. Between 6% and 7% of the pages we searched through was porn, meaning a sizable data set is nearly impossible to find this way within reasonable amounts of time.

2.4.2 Crawling domains

Another option for gathering data was to send out a crawler to gather a chunk of information from a single pornographic domain. This type of data was of somewhat lesser quality. Using this data to train an algorithm would mean that we would be training it *s*pecifically for the websites we crawled, instead of for pornography in general. Another problem was that this data contained no 'borderline'-cases: sites that lie somewhere between regular sites and porn sites (think of dating sites or escort services). This is the type of training sample that is most important to SVM classifiers, since the borderline cases will end up being support vectors, and eventually will decide the location of the separation hyperplane.

2.4.3 Manual classification of suspicious documents

The third method of gathering data could only be used in the final stages of the process. In this case we would use a classifier already trained on data gathered using the first two methods. We let the classifier calculate the probabilities of the documents being porn on a huge amount of random data. The higher rated websites could automatically be included in a new training set, the websites the classifier had doubts about were checked and classified manually.

This approach combined the strong points of the first two methods nicely. The obvious pornographic content could automatically be harvested. The only thing that had to be checked manually was the somewhat more ambiguous content. This method was the one we used to build the models for OpenIndex.

2.5 The Experiments

We measured the performance of the SVM classifier on the specific issue of porn-filtering in a number of different experiments. We used these results to optimize the final algorithm. First off we looked at how much influence the size of the training set had. Next we ran some tests to look at the influence of the C parameter. Furthermore, we looked at two types of input: binary or counted. Finally we compared the SVM classifier to the Naive Bayes classifier, which will be discussed in the corresponding section.

3 Results

The results presented in this section were gathered in experiments performed on qualitatively and quantitatively different data sets. This means that the absolute results of performance can vary between experiments. The experiments show different trends and effects of different variations in parameters, not absolute measures of performance. Out on the internet the true performance will probably be different.

3.1 The influence of the size of the training set

The first experiments we ran concerned the size of the training set. How would the performance of the classifier change if we changed the size of the training set? To measure this we used a relatively small data set of 1067 documents. By changing the cross validation fraction (`cvf`) we can clearly see the influence of the size of the training set.

We performed 5 iterations (it) for each test to guarantee that we would have enough samples to rule out randomness. In this graph we show results for classification based on the rule: *if* $P(porn) > P(regular)$ *then classify as pornography.*

The first thing we notice about the graph (Figure 1) is that the size of the training set is clearly important for the results. As the cross validation fraction increases, the performance of the classifier also increases. This is important to remember: we need enough data to train the classifier.

The next thing we notice about this graph is that the classification of regular web pages is nearly 100%, while, with not too much training data, the correct classification of pornographic documents is quite poor (70%-80%). This implies that the threshold, θ, for classifying a document as pornography should not be 0.5 (this value can be deduced from the decision criterion $P(porn) > P(regular)$ and the fact $P(porn) + P(regular) = 1$). It might be interesting to know if there is an optimal θ, and if so, at what value.

3.2 The influence of decision threshold θ

For the next experiment we used a larger data set of 1540 samples. Again we ran 5 iterations to ensure a significant outcome. In this experiment we varied θ to see if we could find an optimal decision criterion based on the probabilities. The standard decision criterion $P(porn) > P(regular)$ has proven to be suboptimal for the SVM classifier on this specific problem in the previous experiment. The best value for the threshold would be if the classifier showed good performance on the pornographic content, without compromising on the classification of the regular content.

Figure 2 shows the trade-off between correct pornographic classification and correct regular classification, depending on the threshold. Even though overall classification seems to be remarkably stable throughout the whole experiment, it is clear that there is an optimum at about 0.25. At this θ-value we get a large portion of correct classification within the pornographic data without compromising a lot on regular classification.

3.3 C parameter

Generally the C parameter is considered to be one of the most important parameters for SVM classifiers. C tells something about the amount of generalization the algorithm will perform. Formally C determines the trade-off between the margin between the separation plane and the support vectors, and the complexity of the model [Hsu et al., 2009]. So a high value for C might result in over training on the specific samples in the training set, while a low value might result in over generalization. This parameter usually can take on any value in an enormous range.

C	Correctly classified
0.0001	91.236%
0.01	94.382%
0.1	93.895%
1	93.972%
10	92.734%
100	92.697%
10000	90.487%

Table 1: Performance of the classifier depending on the C value (n = 1335, cvf = 0.6, it = 5)

The results in Table 1 are slightly lower than the results in Figure 2. The reason for this is that we used a lower cross validation fraction and a smaller dataset to give a better view on the C parameter. The influence of the C parameter appears to be quite limited for this specific problem. The best performance is reached at 0.01, but up to 1 the differences are not too big.

3.4 Type of input

The input vectors used to train and test the algorithm can be of two different types. Either all the words are counted and added to the vector, or, no matter how many times a word appears in a document, the vector just contains binary information about whether a word appears at all. We tested on a large and representative dataset of 1334 documents, with 80% cross validation and ran 10 iterations for both types of input.

Type of input	Correctly classified porn	Correctly classified regular	Correctly classified overall
Counted input	95.011%	99.317%	98.315%
Binary input	93.390%	99.273%	97.902%

Table 2: Performance of the classifier depending on the type of input vector

As we see in Table 2 the differences are not very relevant. The histogram input performs slightly better, but not by much.

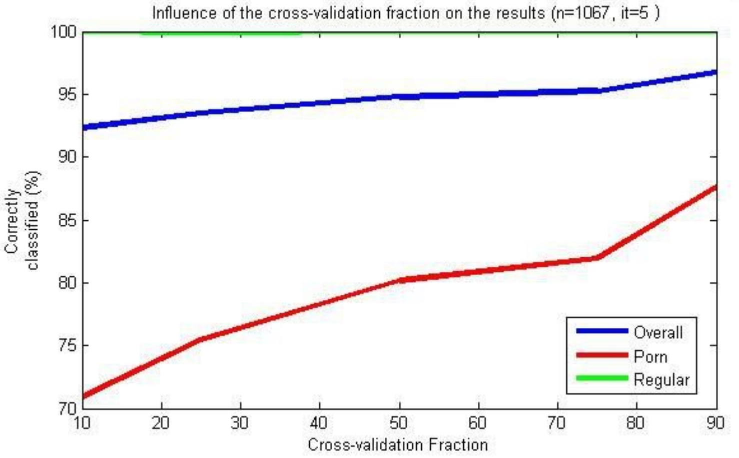

Figure 1: The effect of varying the cross validation fraction, and thus the size of the training set, on the relative amount of correct classifications of pornographic content, regular content and overall content.

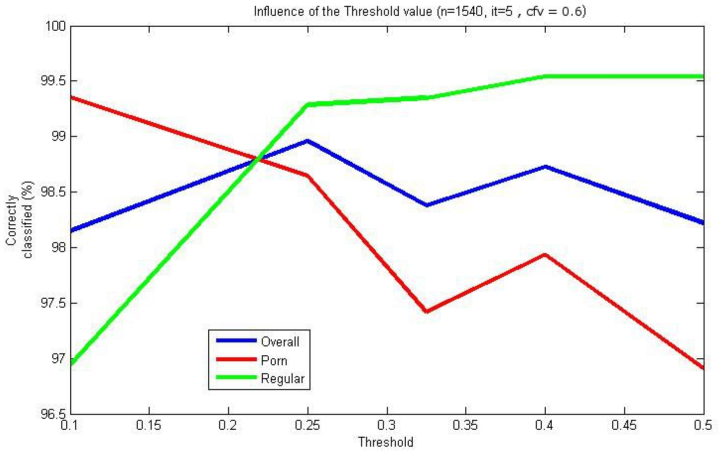

Figure 2: The effect of θ on the relative amount of correct classifications of pornographic content, regular content and overall content.

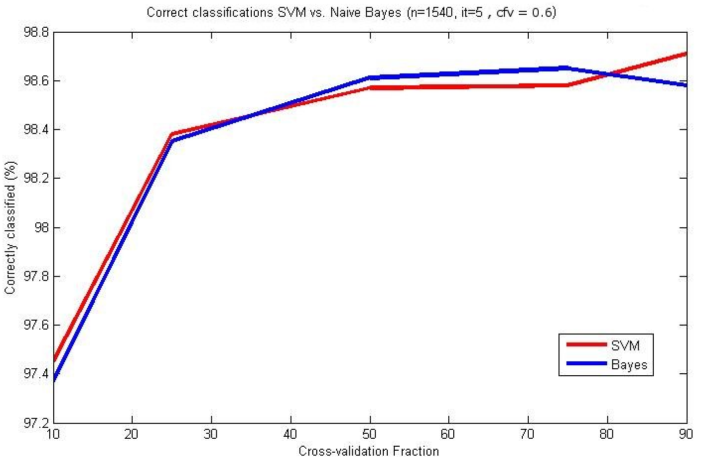

Figure 3: The overall performance of both the SVM classifier and the Naive Bayes classifier with varying cross validation fractions.

3.5 Comparison to Naive Bayes

Another popular way of classifying data (a very common application is filtering Spam) is the Naive Bayes classifier, which is based on Bayes' theorem [Sahami et al., 1998]. In this section we will compare the SVM classifier to the Naive Bayes classifier.

It is interesting to see how strikingly similar the two classifiers perform (Figure 3). Both classifiers appear to be very dependent of the amount of training data, but in the end both can reach very acceptable levels of correct classification.

4 Discussion

4.1 Size of the training set

The results obviously imply that for an optimal performance we need large training sets. The classifier trained on the training set we used in the first experiment (Figure 1) reached acceptable levels of performance only when the cross validation fraction was very high. For the rest of the experiments we raised the amount of samples in the data set to 1540, 770 in both categories, porn and regular. For the models for OpenIndex we need as much data as possible.

The more information the model contains, the better the classifier will perform, without losing much speed. The SVM classifier bases its separation hyperplane purely on a limited number of support vectors, the feature vectors of the samples that lie closest to it. For this reason an increase of the data set with a large number of documents will only increase the size of the models moderately.

4.2 Threshold of the decision criterion

The results of the threshold experiment seem to indicate an optimal threshold around $\theta = 0.25$. This value gives the best results in the classification of pornographic documents without losing performance of classification of regular pages. Although this value holds for the randomly determined test set, based partly on the previously discussed not-so-diverse crawled web pages, this value will undoubtedly be different outside the training set (i.e. on the internet). However, it is evident that this threshold can be optimized in such a way that the porn classification rate is optimized while the regular classification rate is being preserved. In the end it's up to OpenIndex how 'safe' they want the classifier to operate. With a threshold of 0.9 there will be almost no false positives, but some pornographic content might slip past the filter. When the threshold is set to very low values all pornographic content will be filtered out, but regular content might also be classified as porn. It will depend on the practical application of the filter how much risk they can take allowing pornography to slip through the filter.

4.3 Comparison to Naive Bayes

The practical consequences of the remarkable results of the comparison to the Naive Bayes classifier will be minimal. Since both the SVM classifier and the Naive Bayes classifier show good results in filtering pornography apparently both algorithms are suitable algorithms to use on this specific problem.

4.4 Analyzing the misclassifications

The misclassifications of the algorithm can be roughly divided into three main categories. In this section the three categories will be explained while looking at three typical examples for that type of misclassification:

4.4.1 Adult websites using regular words

– *Example:*

```
http://www.dubaisexyescorts.com/
Dubai Sexy Escorts - +971 555 35
38 71 - Dubai Sex Companions -
Dubai Sexy Night Girls - Female
Escorts in Dubai Feature Escorts
Sponser Listing Welcome To Dubai
Sexy Escorts Dubai History Dubai
is the capital of, and the second
largest city in the United Arab
Emirates. Dubai lies on a
T-shaped island jutting into the
Persian Gulf from the central
western coast. The city proper,
making up an area of 67,340 km2
(26,000 sq mi), had an estimated
population of 860,000 in 2008.
Parts of Dubai were settled in...
```

This page was manually labeled as porn, although the classifier did not classify it as such. The text on the page seems rather innocent. Only at the bottom of the page inappropriate words are used. Opening this page in an internet browser however, will prove that this website is definitely adult-themed. The most effectual way to improve classification of this type of page is to use image analysis. This is however not possible using a crawler.

Image analysis would take too much time and would require too much data to be downloaded.

4.4.2 Adult websites using mainly names

– *Example:*

```
http://www.xnxx.com/pornstars
PORNSTARLIST AaliyahJolie (24)
AaralynnBarra (13) AbbeyBrooks
(19) AddisonRose (28)
AdeleStevens (84) AdinaJewel
(21) AdrianaMalkova (42)
AdrianaRusso (88) AdrianaSage
(96) AdriannaDeville (12)
AdriannaFaust (23)
AdriannaNicole (22)
AdriannaZarcova (14)
AdrienneShand (16)
AfroditeNight (17) Afrodithe
(17) AimeeSweet (72)
AimeeTaylor (15) AimeeTyler
(10) Aisha(15) AkiraLane (11)
AlanaEvans (80) AlanaPlay (11)
AlauraEden (196) AlayahSashu...
```

This page was also labeled as porn, but the classifier did not classify it as such. This type of page lists people with a certain profession and seems to mention almost no regular words. This is very hard to classify for an algorithm, since there is no way the algorithm could know what profession these people exercise. A person would easily recognize this page as porn, but the classifier doesn't generally know words such as PORNSTARLIST. Even if there would be obvious 'tells', the classifier will not base its classification on just a single or a few words. If it would do that it could easily classify a lot of false positives. Sex education websites are good examples of sites that might contain seemingly inappropriate words, but are in fact not pornographic. If the classifier would base page classification on a single instance of an obscene word, these educational websites would be classified as porn as well.

4.4.3 Questionably pornographic websites

– *Example:*

```
http://www.eastangliaswingers
.co.uk/ East Anglia Swingers -
Home of Swinging in East Anglia
Contact Us East Anglia Swingers
- Helping UK Swingers make Local
Adult Contacts If youwant to meet
others who swing in East Anglia
and across the UK, just search
below and register for free now.
JOIN NOW FOR FREE Looking for
Swingers in Norwich, Cambridge,
Peterborough, Ipswich and
Colchester both local to you and
across the UK. Register For Free
and search for either Single
Swingers or Swinging Couples in
a town or city near you. The
East Anglia Swingers network
contains 1000's of sexy adult
photo ad's...
```

This page was also labeled as porn, but the classifier did not classify it as such. This third category of misclassifications is perhaps the most interesting. Some web sites are even for humans hard to classify. People could discuss about these web sites where the content itself seems quite innocent but the theme of the website (in this particular example sex-dates) would definitely qualify as being adult-themed. Obviously the classifier disagreed with our decision to label this website as porn.

4.5 Duration of classification

Apart from the classifier having to produce a high rate of correct classifications, another requirement it needs to fulfill is that running the SVM cannot take too long. It should not be the bottleneck in the process of crawling data. Although the exact time it takes to classify a single document on the servers is hard to find out, the implementation of the Nutch-plugin has proven in practice to work quick enough. Crawling the internet will not be slowed down by the porn classification plugin.

4.6 Other findings after implementation

After implementation and evaluation of the classifier we noticed a minor weakness. Despite high accuracy it seems that as of yet there are still some misclassifications which cannot be fixed by tweaking the classification threshold. It seems that the only solution to improve the classifier performance would be to improve and expand the model that is currently used for classification. We will have to wait and see how much improvement we can achieve by gathering more representative data to create a better training set.

4.7 Future work

Improvement of the classifier, as discussed above, might be possible through raising the quality and the quantity of the training data to improve the model. The more training samples used to build the model, the better performance we will observe.

Another possible improvement to the classification of web pages, also mentioned above, would be to analyze more than just the flat text on the page. Images and videos contain a lot of information about the nature of a page, but are a lot harder to analyze. It takes more complex algorithms to extract useful features from images and videos than from flat text. These algorithms would take more time to classify crawled data, and would slow down the whole process of crawling the internet.

Image analysis combined with text analysis might give even better results. The quickness of text analysis could be combined with high performance rates of image analysis by first looking at the text, and if there is still doubt about the page, analyze the images. In this way only some bordeline cases would need image analysis.

References

[Amayri and Bouguila, 2010] Amayri, O. and Bouguila, N. (2010). A study of spam filtering using support vector machines. *Artificial Intelligence Review*, 34(1):73–108.

[Chang and Lin, 2011] Chang, C.-C. and Lin, C.-J. (2011). Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at http://www.csie.ntu.edu.tw/˜cjlin/libsvm.

[Diao et al., 2012] Diao, L., Yang, C., and Wang, H. (2012). Training SVM email classifiers using very large imbalanced dataset. *Journal of Experimental & Theoretical Artificial Intelligence*, 24(2):193–210.

[Drucker et al., 1999] Drucker, H., Wu, D., and Vapnik, V. (1999). Support vector machines for spam categorization. *Neural Networks, IEEE Transactions on*, 10(5):1048 –1054.

[Hsu et al., 2009] Hsu, C.-C., Han, M.-F., Chang, S.-H., and Chung, H.-Y. (2009). Fuzzy support vector machines with the uncertainty of parameter C. *Expert Systems with Applications*, 36(3, Part 2):6654 – 6658.

[Hu et al., 2007] Hu, W., Wu, O., Chen, Z., Fu, Z., and Maybank, S. (2007). Recognition of pornographic web pages by classifying texts and images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(6):1019–1034.

[Jezek and Hynek, 2007] Jezek, K. and Hynek, J. (2007). The fight against spam - a machine learning approach. In Chan, L. and Martens, B., editors, *ELPUB*, pages 381–392.

[Sahami et al., 1998] Sahami, M., Dumais, S., Heckerman, D., and Horvitz, E. (1998). A bayesian approach to filtering junk E-mail. In *Learning for Text Categorization: Papers from the 1998 Workshop*, Madison, Wisconsin. AAAI Technical Report WS-98-05.

[Selamat et al., 2011] Selamat, A., Zhi Sam, L., Maarof, M. A., and Shamsuddin, S. M. (2011). Improved web page identification method using neural networks. *International Journal of Computational Intelligence & Applications*, 10(1):87 – 114.

[Wang et al., 2009] Wang, Y., Huang, Q., and Gao, W. (2009). Pornographic image detection based on multilevel representation. *International Journal of Pattern Recognition & Artificial Intelligence*, 23(8):1633 – 1655.

[Znaor and Elzinga, 2011] Znaor, A. and Elzinga, L. (2011). Email classifier using machine learning algorithms.