Selecting Negative Examples for Training an SVM Classifier

Robert Toth

Master Thesis
Fall 2011

Advisors: Anders Ardö & Fredrik Andersson

Lund University
Department of Electrical and Information Technology

Printed in Sweden
E-huset, Lund, 2011

Abstract

In this thesis, five different methods for selecting negative examples for training an SVM have been developed and evaluated. The aim of the methods is to produce examples of the complement to web pages from a category of interest – the positive examples. The methods were tested with three different positive categories. The first four methods are attempts to artificially create negative examples while the fifth simply takes random web pages as negative directly. The fifth method produced a classifier with surprisingly high performance, measured in precision and recall. Classifiers trained with the artificial examples, however, performed worse.

Acknowledgements

I would like to thank my two advisors Anders Ardö, Electrical and Information Technology at Lund University, and Fredrik Andersson, at Entireweb, for their never ending patience and support. They have both helped me with ideas, encouragement, explanations, feedback and general good practice for working with larger projects.

My wonderful spouse, Linnéa Möll-Nielsen, also deserves special thanks. Not only did she suffer through my many ramblings about classifiers, but also pushed me to keep working when motivation was scarce. Her support and expectations have kept me going.

Table of Contents

1 Introduction **1**
1.1 Problem description . . . 2
1.2 Thesis organization . . . 3

2 Theory **5**
2.1 Support Vector Machine . . . 5
2.2 Document representation . . . 7
2.3 Performance measure . . . 9
2.4 Related work . . . 9

3 Experimental setup **11**
3.1 The categories . . . 11
3.2 Manual classification . . . 12
3.3 Global document frequency . . . 14
3.4 Spam reduction . . . 15
3.5 Weights . . . 15
3.6 Stop words and stemming . . . 16
3.7 Performance measurement . . . 16

4 Method **19**
4.1 Building the corpus . . . 20
4.2 Early testing . . . 21
4.3 Tests . . . 21
4.4 Model selection . . . 24
4.5 Implementation . . . 24

5 Results **27**
5.1 Category characteristics . . . 27
5.2 Artificial samples . . . 28
5.3 Real samples . . . 30

6 Summary and discussion **31**
6.1 Summary . . . 31
6.2 Index pages . . . 31

6.3 A note on vocabulary size 32
6.4 Using artificial weights 32
6.5 Using existing pages as negative 32
6.6 Training vs. prediction accuracy 33
6.7 Conclusions 33

7 Future work **35**

References **37**

A Weight distributions **39**

B Vocabulary/Vector overlap diagrams **55**

List of Figures

1.1 Different types and degrees of negative examples along with an attempted separating rule . . . 2
1.2 This illustrates the benefit of using examples from as many negative classes as possible . . . 3

2.1 (A) Data points from two different categories. (B) Two of the many possible separating lines. (C) Maximum margin separating line. . . . 6
2.2 Example of data that is not linearly separable but can still be used to train a classifier using error tolerance . . . 7

4.1 Overview of the implementation architecture . . . 19
4.2 Example of a URL listing on DMOZ along with sub-categories. Screenshot from http://www.dmoz.org/Kids_and_Teens/School_Time . . 26

A.1 Christianity with no TF ceiling . . . 39
A.2 Christianity with a TF ceiling of 15 . . . 40
A.3 Christianity with a TF ceiling of 10 . . . 41
A.4 Christianity with a TF ceiling of 5 . . . 42
A.5 Cooking with no TF ceiling . . . 43
A.6 Cooking with a TF ceiling of 15 . . . 44
A.7 Cooking with a TF ceiling of 10 . . . 45
A.8 Cooking with a TF ceiling of 5 . . . 46
A.9 Martial-arts with no TF ceiling . . . 47
A.10 Martial-arts with a TF ceiling of 15 . . . 48
A.11 Martial-arts with a TF ceiling of 10 . . . 49
A.12 Martial-arts with a TF ceiling of 5 . . . 50
A.13 Random pages with no TF ceiling . . . 51
A.14 Random pages with a TF ceiling of 15 . . . 52
A.15 Random pages with a TF ceiling of 10 . . . 53
A.16 Random pages with a TF ceiling of 5 . . . 54

B.1 The fracation of positive vectors from Christianity that overlap with the vocabulary . . . 55

B.2 The fracation of negative vectors (from the random pages collection) that overlap with the vocabulary from Christianity 56
B.3 The fracation of negative vectors (from the random pages collection) that overlap with the vocabulary from Cooking 56
B.4 The fracation of negative vectors (from the random pages collection) that overlap with the vocabulary from Martial-arts 57

List of Tables

5.1 Results from training and prediction test, using uniformly distributed random weights 28
5.2 Results from training and prediction test, using weights from the weight distribution of random pages 28
5.3 Results from training and prediction test, using inverted weights from the positive examples 29
5.4 Results from training and prediction test, using inverted weights from the positive examples and extended vocabulary 29
5.5 Results from trial and error test, using uniformly distributed random weights 29
5.6 Results from trial and error test, using weights from the weight distribution of random pages 29
5.7 Results from trial and error test, using inverted weights from the positive examples 29
5.8 Results from trial and error test, using inverted weights from the positive examples and extended vocabulary 30
5.9 Results from extra test along with the cost parameters that gave highest training accuracy. The method for choosing error cost and training is the same as for the previous experiments. 30

Chapter 1

Introduction

In a library, literature is ordered by appropriate categories. Books rest on shelves labeled with the name of the category, and on the shelves of the same category the books are usually ordered by author name. A visitor can easily find a specific book and even literature from a specific category with the help of said labels. Such convenience is possible because someone has taken the trouble to label each book with the correct category and placing them on their respective shelves.

In many ways the World Wide Web is like a library. One difference is that there is so much information on the Web that there are not enough librarians on earth to label all of it during one life time. Even if there were, more information is being added constantly and at an ever increasing rate, making it something of a Sisyphean task. However, if shelves can be replaced by computers then so can the librarians that label the information.

Today, it is possible to train a computer to recognize the differences between texts belonging to different categories. In other words; given a collection of unlabeled documents, the computer is able to sort them into categories much like a librarian would sort books into shelves. The Support Vector Machine (SVM) is one method for training computers to accomplish this. It is a learning algorithm that learns by example. When presented with example data from every category that it should learn, it can create a general rule which can be applied to any unlabeled text – thereby labeling it. There are other methods as well but this thesis only deals with SVM.

The categorization rule created by SVM can often accomplish categorization of text task rather well if the categories are well defined. The complement of a category, however, is difficult to define in any other way than indirectly – it is more difficult to say what it is than what it is not. Since the SVM needs examples of everything it is to learn it needs examples of texts that do not belong to the category of interest. The problem is: what are good examples of not belonging to the category of interest? Such examples – so called negative examples – must be as different as possible from the positive examples. The problem is further complicated by the possibility of there being many different kinds of negative examples where two, or more, can be equally different from the positive but also from each other.

Figure 1.1 illustrates such a scenario with four different classes of negative examples. It is therefore also of importance to provide the SVM with negative examples from every class so as to cover the entire negative space. Figure 1.1 also contains an example of a categorization rule, also known as a separating rule, in the form of a line that (rather poorly) separates the positive examples (dark circle) from the negative ones (light circles). Another level of complication, which is also depicted in Figure 1.1, is the somewhat fuzzy boundaries between classes — e.g. it is not always clear if a text is about Christianity or cooking, it may be about both or more or less about one than the other.

In this thesis, a few methods of selecting negative examples for training an SVM are developed and evaluated.

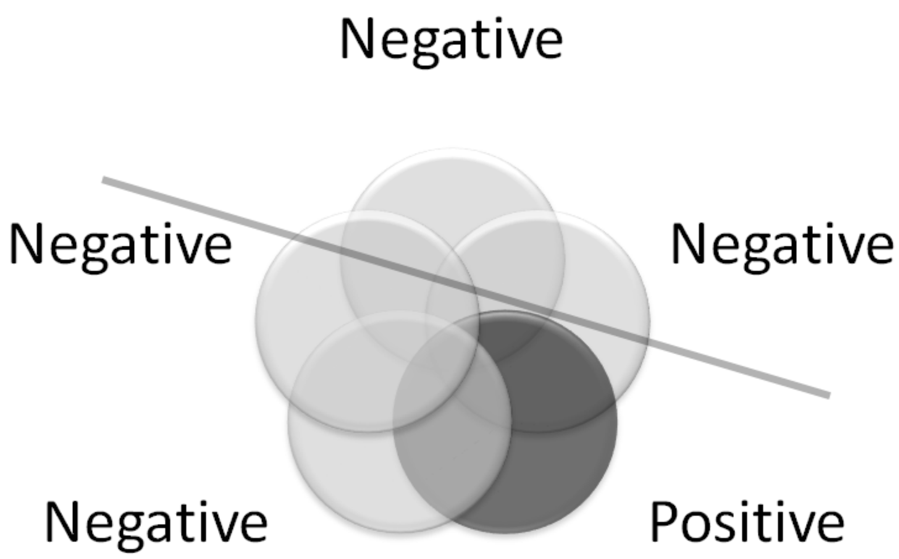


Figure 1.1: Different types and degrees of negative examples along with an attempted separating rule

1.1 Problem description

The setting for the thesis is that there exists a category of interest along with a collection of example web pages that have been labeled as belonging to that category. Furthermore an SVM classifier is to be trained, using the aforementioned example web pages, which will later be used to distinguish between unlabeled web pages that belong to the category of interest and any other web pages.

The intention of this thesis is to evaluate different ways of selecting the negative examples that are also needed for the training. As discussed in the previous section, negative examples must be composed of a good enough representation of the web, excluding pages from the category of interest, to produce a high performance classifier. The importance of this can be seen in Figure 1.2 where the right side line separates more negative examples from positive ones because the class of the top most circle is known. This also needs to be done in a time and effort efficient way so as to allow for use in a commercial setting.

Many previous papers have dealt with training SVMs for text classification using different standard data sets. However, such sets tend to consist of very homogeneous text as opposed to random pages on the web. In this thesis only web pages will be used and the data sets will be built from scratch.

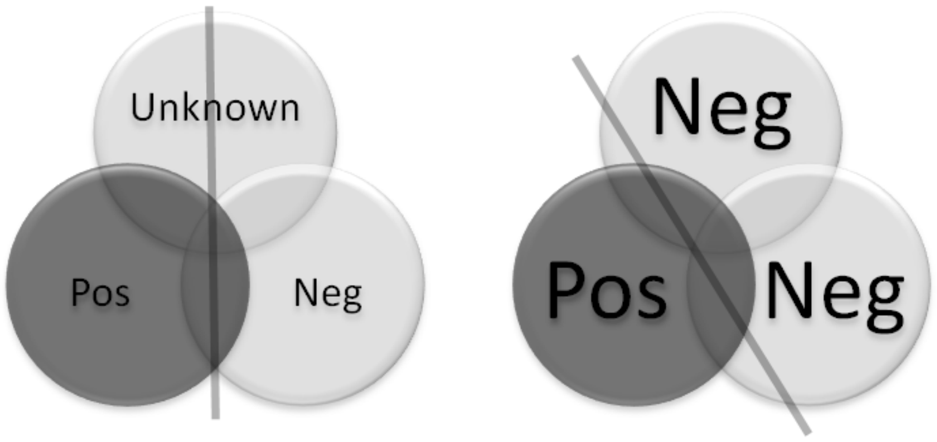


Figure 1.2: This illustrates the benefit of using examples from as many negative classes as possible

1.2 Thesis organization

Chapter 2 presents the theory that much of the work in this thesis is based on. First the idea behind SVM and text categorization in general is explained followed by brief introductions to the different information retrieval specific concepts that are also required in order to understand the results.

Chapter 3 describes the specifics about the experiments and the design choices. It is also meant to facilitate a continuation of the work presented in the report.

The proposed methods for selecting negative examples are described in chapter 4 along with other methods used during preprocessing and analysis of results. A brief description of the developed code is also presented as an orientation.

Chapter 5 presents the results which are discussed and commented in chapter 6 and the report ends with suggestions for future work in chapter 7.

Chapter 2

Theory

This chapter contains short introductions to the concepts and theories required to understand the results and the paper in general. First the SVM is explained in a conceptual fashion with the purpose of communicating the idea behind it and to give a feel for how it works. Formal descriptions and mathematics will be avoided in order to keep the focus on understanding the essence. There are already many mathematical explanations in the literature [1] [2][3] for those interested.

Thereafter the concept of representing documents as vectors is explained along with why it is needed and followed by a short explanation of the way that performance is measured. The chapter ends with a brief look at what other work has been done in related areas of SVM research.

2.1 Support Vector Machine

The support vector machine (SVM) is a supervised[1] machine learning algorithm, used to classify data. Given some training data that has been manually classified, the algorithm can produce a general rule that is applicable on previously unknown data of the same type[2].

A more specific application is the categorization of text which is also the focus of this thesis. By providing the learning algorithm with pages of text belonging to different categories it can learn to distinguish between them by inferring a rule. The rule can then be applied to any text from those categories and the classifier will try to predict which category the text belongs to. This thesis will only deal with the two-category setting though; one category of pages that are of interest, called positive, and the complement to that category, called negative[3].

The theory behind SVM has its roots in statistical learning theory and was first developed by Vladimir N. Vapnik et al. [1, p. 35] [2, p. 421]. However, the theory in itself is very general — SVM can be used on any type of data, be it images or text. It was Joachims who first did a serious analysis of the text classification task using SVM [1]. He also implemented the algorithm in the form of SVMlight [4].

[1]It cannot learn on its own but must be supplied with examples.

[2]If the generated rule will be applied to images for instance, then the training must also be done with images.

[3]The complement consists of any other category that is not the category of interest.

A simple way of illustrating how SVM works is with a 2D example. Suppose there are some data points and every point belongs to one of two categories. When plotted in a diagram, points belonging to the same category tend to lie closer to each other than to points from the other category, as illustrated in Figure 2.1(A) . One way of creating a general rule for determining what category a point belongs to, is to draw a line between the groups of points belonging to the different categories. When a new point is added, its category can be determined simply by looking at which side of the line it appears on.

However, there may be many possible lines that separate the two groups of points. This is of importance because the angle at which the line is drawn will determine on which side of the line new points will appear — and consequently which category they will belong to. Two extremes are shown in Figure 2.1(B) where the grey points can belong to either category depending on the line chosen. There are, however, many more possible lines that could be drawn.

SVM not only finds such a separating line but also the, so-called, maximum-margin line. So, essentially the problem is an optimization problem where the size of the margin is maximized with the constraint that positive data points must lay on one side of the line and negative on the other. An example of this can be seen in Figure 2.1(C). The two outer lines, running through some of the points, shows the margin between the separating line and the points, and the points that lie on those lines are called support vectors.

The white and black points in Figure 2.1 are the training data — the separating line is created from them. The circles come from unclassified data and are classified by the separating line.

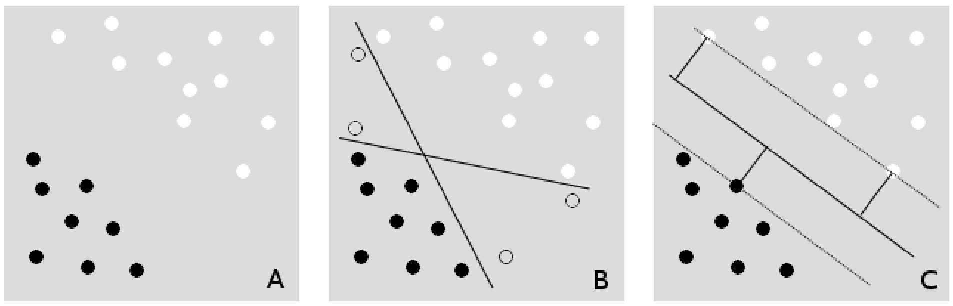


Figure 2.1: (A) Data points from two different categories. (B) Two of the many possible separating lines. (C) Maximum margin separating line.

Two dimensions are often not enough though. As will be explained in section 2.2, the text categorization task often requires tens of thousands of dimensions. It is hard to imagine a line is such a space, but in three dimensions the line becomes a separating plane and in higher dimensions the plane is called a hyper plane.

Many times the data is not linearly separable at all. In other words, there is no line or hyper plane that can separate all the data points in one category from all the points in the other. However, it is not necessary to find such a hyper plane to create a classifier that is good enough, that is to say a classifier that makes correct predictions most of the time. With the introduction of a, so-called, slack variable, some level of errors during training are permitted. The trainer used in this thesis takes a parameter (C) which is the error cost and determines how much error should be permitted. Figure 2.2 illustrates a scenario where the classifier would perform well enough despite the four renegade data points.

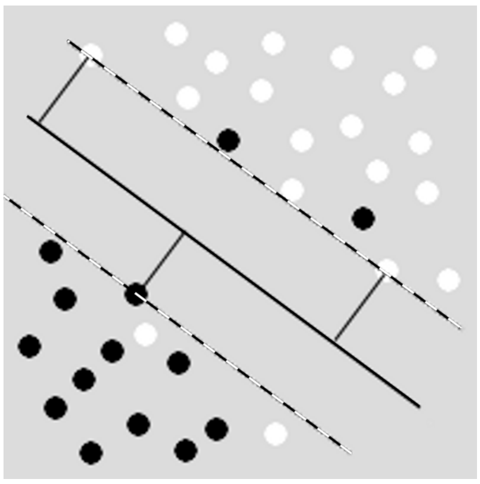


Figure 2.2: Example of data that is not linearly separable but can still be used to train a classifier using error tolerance

For some applications a hyper plane might not be an optimal — or even possible — separator. It is possible to use other forms of separation such as radial, polynomial, etc. But in this thesis, only the linear mode of classification will be used as it is known to work well with text classification [3]. Therefore, if not explicitly stated otherwise, this report will refer to the linear form throughout. For information about the other forms of separation, please refer to other literature on the subject.

2.2 Document representation

There are many ways of representing information. The written text is, perhaps, the most common and widely used one. Unfortunately, it is not suitable for training the SVM. As the name suggest, the algorithm is designed to work with vectors. To this end a document of text needs to be represented in vector form where the elements of the vector are some values that represent each unique feature of the text.

One possible type of feature could be the letters of the text. Another, perhaps more natural, is to use words as features. In this thesis words are used as features but whatever the choice, the features make a feature space where each feature corresponds to one dimension. In such a space, every text becomes a point, defined by its features in the vector representation of the text.

Because words have been chosen as features, the total number of features is the total number of unique words in the entire collection of pages — usually training is done with many example pages that make a collection. Therefore, it is not at all uncommon to have feature spaces with tens of thousands of dimensions. [2, p. 2]

The elements of the vectors are called weights and, just like with features, they can be defined in many different ways. But generally speaking, the weights should reflect the features importance in characterizing the text. Words that occur often in a document are probably, in some way, defining for that document. However, if those same words occur in all documents — and consequently in all categories — they contribute little or nothing to the categorization task. Such statistics of the words are usually described by document frequency (DF) and term frequency (TF), where DF is the number of documents a word occurs in, and TF is the number of times the same word occurs in a single document. Since a large DF makes a word a bad indicator[4], it is common to use the inverse (IDF) instead for measuring relevance. In information retrieval weights are often calculated using Equation 2.1. The equation is commonly used in information retreival for calculating weights and takes into account both the benefit of a high TF and the drawback of a high DF [1, p. 21]. All normalization has been abstracted in order to convey the principal.

$$w = \frac{TF}{DF} = TF \cdot IDF \tag{2.1}$$

Words like "to", "the" and "or" for instance do not contribute much when trying to determine what the text is about. This is because they are found in just about any text. But if a text contains many instances of the words "binary", "CPU" and "memory" then one can be fairly certain that the text is about computers, in one way or another, since they usually only occur in texts about computers. The first group of words should therefore receive lower weights and the latter group higher weights.

Actually, the first group of words mentioned above contributes so little information that they are generally not even included in the feature set. Words like that are called stop-words and are often filtered out of a document before turning it into a weight vector. It not only reduces the length of each document but also the number of dimensions in the feature space.

Another method for reducing feature count is by stemming words. All words are transformed into the stem of the word because apple and apples contribute the same amount of information about the category. They do not need to be treated separately. However, stemming is not used in this thesis.

When constructing the weight-vector of a document the order in which the words appear is not important — only the statistics of the words are. The document is therefore first converted into a, so-called, word-bag which is, in practice, usually just an array of all the words and their TF values.

Statistical analyses of words have provided some interesting results. Zipfs law is one such result which describes the distribution of words. More specifically; it

[4]The presence of a good indicator makes it more probable that a text belongs to the indicated category. Bad indicators do not contribute much to the identification of the category that the text belongs to.

states that [5, p. 146]

> *... the i-th most frequent word is 1/iθ times that of the most frequent word.*

The parameter θ depends on the characteristics of the text but can be set to 1 for simplicity, sacrificing some accuracy. The behaviour of word distributions is still clear, even though the simpler formula is used; the second most frequent word occurs half the times of the most frequent one, the third most frequent occurs one third the times of the most frequent one, etc.

2.3 Performance measure

One common way of measuring performance of information retrieval systems and classifiers is precision and recall [1] [5].A good example that most people can relate to is a search engine. When a search is done for pages about cars then recall essentially describes how many of the relevant pages were really returned by the search engine. If there are 100 pages about cars but the search engine only returns 70 of those, then the recall is 0.7. Note that the search engine might have returned 200 pages in total, out of which 130 are about something else, but recall ignores that.

Precision on the other hand measures how many of the pages that were returned actually are about cars. So this is the other side of the performance, the one ignored by recall. If, as before, 200 pages are returned but only 70 of those are about cars then the precision is 0.35.

This thesis does not deal with search engines however, but instead with the classification of text. The difference in precision and recall is that instead of pages returned, the measure is of pages correctly classified. See Equation 3.3 for the definition.

2.4 Related work

The main inspiration and reference used in this thesis was Thorsten Joachims well written and comprehensive book about using SVM for text classification [1]. It is not only a must-have reference for anyone working with SVMs, it is also quite enjoyable reading for those interested.

Some of the ideas behind the methods developed were inspired by Sassano's paper on using virtual examples based on existing ones to extend the manually categorized set of training examples — the set is called "annotated corpus" in the paper [6]. Sassano motivates his work with the fact that "... corpus annotation is labor intensive and very expensive." Most of the examples used in this thesis are entirely virtual — or artificial as they are called in this report — but the idea is in part an extension of Sassano's.

Another interesting paper presents a method for training an SVM with few positive examples and a large number of unlabeled examples. Yu Hwanjo et al. present a way of iteratively training an SVM by starting with positive examples and only the "strong negative" [7] examples. The first classification on the rest

— the ones that are neither positive nor strong negative — will yield a few more negative examples which are then added to the initial set of only strong negatives. This is iterated until there are no more unlabeled samples. As interesting as these results are, they were not quite applicable to the problem of this thesis since the problem would merely have shifted from selecting negative examples to selecting unlabeled examples.

Chapter 3

Experimental setup

There are many parameters and methods when working with SVM and information retrieval in general. This chapter documents exactly which numbers, methods, configurations and other data that were used in the implementations and testing.

First there is a brief motivation of the choice of categories for the positive examples followed by an explanation of how manual categorization was done and why it is needed. The rest of the chapter presents some technical details regarding document and feature representations and calculations.

3.1 The categories

The positive categories that are used to test the classifier must be chosen so as to offer a fair challenge. If the category is highly specific e.g. medical reports — where each text contains mostly highly specialized word — it will be too easy to distinguish between the positive and the negative — negative being other text. At the same time the category cannot be too general, because that would make it very difficult for the classifier to separate from other categories — perhaps even for a human. The focus must be on the classifier, a good classifier should be able to perform well and a bad one should produce bad results.

Another important aspect to consider when choosing positive categories is that manually confirmed examples are needed when testing. Either there should be a pre-categorized collection available or there must be someone who is confident enough in the subject to be able to manually select examples that, with some level of certainty, indeed do belong to the category. It is difficult to define what a good category is, but there are some characteristics that are desirable:

1. There should be terms that are so unique to the category, that the appearance of such a term in a text is a strong indication that the text belongs to the category in question.
2. There should be terms that are so distinct from anything belonging to the category that the appearance of such a term in a text is a strong indication that the text does not belong to the category in question.
3. There should be ambiguous terms that do not contribute much or any information about what category a text, containing those terms, should belong to.

3.1.1 Religion

This was an early candidate, but religion as a category is much too general. It was therefore narrowed down to Christianity. The choice of religion is motivated by its rather general nature. Religion often deals with many aspects of life and philosophy. Christianity in particular is often associated with communities and community events. As such, there are many very specific words that immediately draw the mind to Christianity, e.g. bible, Christ, Judas etc. but there are many other words as well because of how deeply integrated in society religion is.

3.1.2 Martial-arts

Martial-art is another category that contains a high degree of philosophy which in turn deals with many areas of life. This gives the category a certain level of generality but, as with Christianity, there are many very specific terms as well despite the fact that all forms of martial-arts are included and not any one specific. The reason for this is that there is relatively little information on the web about a single martial-art style but on the other hand, there are large overlaps in specialized words between different styles.

3.1.3 Cooking

The final positive category is cooking. It has the interesting quality that recipes make up a large part of the data available on the subject so the structure is easily recognizable and there are a few words that are specific to the category — such as recipe and cooking themselves. However, the actual recipes contain animals, vegetables, fruits and many other things that, taken out of context, do not necessarily belong to the category of cooking.

3.2 Manual classification

When testing the classifier, it is important to know exactly what kind of data that is used in the tests. The test pages must therefore be manually classified by a human. Pages in the positive set must be verified so that they are, indeed, positive. The same goes for the negative set, of course.

All three categories are rather general, therefore it is not always entirely clear whether a page should be considered as belonging to the category or not. The general philosophy during manual classification was that only pages with content clearly belonging to the category in question was taken as positive. The same was applied for negative pages. The difference being that a page was considered negative if it was clear that the content had nothing to do with the category in question.

Furthermore, a certain degree of relevance was also considered for the positive pages. Some may not contain any significant quantities of information, yet the little there is may still be important for someone looking for pages from that category. One example is the registration page for the site Crisco[1] . It is made abundantly

[1]http://crisco.com/MyCrisco

clear that registration will give access to something the site calls "personalized recipe box" and a grocery list. By just reading that page, it is obvious that it is relevant in the context of cooking. That page was taken as positive. There were several other registration pages that did not indicate what the visitor was registering for though. Such pages usually only contain a registration form and a send button.

A total of 100 pages each were manually classified for every positive category and 209 pages as negative. Since only one collection of manually classified negative examples are used, the collection is larger by the assumption that such an arrangement would give more accurate test results. The odd number comes from an attempt at selecting examples in proportion to the number of pages contained in the category they are chosen from. More on this in the section about negative examples.

3.2.1 Christianity

Many religious sites are community sites. They often offer a wide variety of activities for their communities. Some of the pages from a site belonging to the Christianity category therefore contain pages that strictly taken would be considered of another category. One example, found during manual classification, was a Capoeira class, offered at the Centre at St. Paul's. The page was not considered to be positive for the Christianity category as it had nothing to do with Christianity, other than belonging to a Christian site.

3.2.2 Martial-arts

Again, the aim was to consider a page positive if it, by itself, could be perceived as being about martial-arts. This is also a broad category, perhaps not easily defined in a clear and consistent way. One topic that was avoided was general work-out, where it could not be ruled out that the page was about any other sport or physical training.

3.2.3 Cooking

One example of pages that were not considered to be positive, were those about restaurants. Obviously there is cooking involved, but in those pages it is only in an implicit sense. Others not included, but suggested by DMOZ, were drink mixing, baking, nutrition guides, kitchen appliances etc. It might be hard to single out the pages about just cooking, as most of the features seem to be shared among pages from the topics mentioned. Another issue was that there is no clear definition of cooking. Perhaps some may argue that baking should have been included, but in this thesis the decision was made to narrow the category down to what may be considered breakfast, dinner, supper and the likes.

3.2.4 Negative examples

Choosing negative examples to test the classifier against presents the same difficulty as the main problem of this thesis; the examples should span the entire negative space so that the results are generally applicable. For the purpose of selecting manually classified pages for testing — in a reasonable amount of time and effort — a few assumptions are made regarding the composition of pages on the web.

Firstly, the web is assumed to be divisible into categories. More specifically, it is assumed that there are N number of categories, and that the entire web, U, consist of all the pages, p, from all the categories as illustrated in Figure 1.1.

Secondly, it is assumed that DMOZ represents a fair estimate of the types of categories that occur on the web, and also of the distribution of the number of pages per category.

Obviously, the more negative examples are used in the testing, the better the coverage will be. Time must still be taken into consideration though, which is why the number of manually classified negative examples is double that of the positive, namely approximately two hundred. They are randomly picked from the largest (highest level) categories on DMOZ in numbers representing the category's share of the total number of pages listed on DMOZ.

There is only one negative collection that is used for testing all three positive categories. Because of this it is important to be certain that each page does not belong to any of the positive categories.

In total, there are 209 manually classified negative pages.

3.3 Global document frequency

Aside from the IDF table for each collection another one was used for calculating the weights. The global IDF is an estimate of the document frequencies on the web and was provided by Entireweb[2]. It is essentially an index dump[3] and therefore contains all sorts of words of which many are just gibberish. To only include sensible words the list was filtered using a regular expression that only allows words containing the English letters a-z.

For each collection, the document frequencies are calculated using Equation 3.1, which is the same equation that was used by Entireweb when constructing the global IDF table for the sake of consistency. The variable N is the total number of documents in the collection, and N_i is the number of documents containing feature i

$$IDF_i = \frac{\log \frac{N}{N_i}}{\log N} \tag{3.1}$$

Since most of the methods for selecting negative examples, that were used in this thesis, use artificial negative examples the vocabulary is only based on the words that are present in the positive collections. Each category is treated separately, meaning that each test is based on a unique vocabulary defined by the positive pages in the collection from that category. The DF table[4] is a collection-

[2]http://www.entireweb.com

[3]A collection of indexed words in Entirewebs search engine

[4]A table of document frequencies compiled from all the pages in a collection for use in the calculations.

wide measure and therefore contains all unique words in the collection which means that it defines the vocabulary for that collection as well.

3.4 Spam reduction

When dealing with web pages in an un-controlled environment like the web, one is bound to come across web pages of all sorts of meaningless or ill willed content. Spam is perhaps the most common irks of the web and pages with a lot of spam can cause trouble in training the SVM by, for instance, elevating the importance of certain words beyond any reasonable levels. Spam in this context refers to the unintelligible repetition of words that are present in some text for various reasons.

To counter any unfair weighing caused by spam, an upper bound is imposed on the number of occurrences per word and document that will be taken into account when calculating TF values. Intuitively speaking, a word should not have to appear too often in an average length text which is why a TF ceiling should not affect the categorization task. There is, of course, the question of what "too often" is.

The TF-ceiling was chosen to be 15. As the distribution graphs in Appendix A indicate, a ceiling of 5 is too low because the largest possible weight is smaller than it is when not using a ceiling — indicating that data is lost. Therefore the ceiling has to be larger than 5. It is also evident by the same graphs, that the ceiling has a much greater effect on the set of random pages. While, intuitively, 10 would have been the better choice — at least as far as the amount of meaning the repetition of a word contributes — given the impact that TF ceilings have on the random set, 15 is closer to the uncut distribution while still eliminating the most severe cases of spam.

Also of note is the fact that the ceiling was chosen to be a constant value for all pages of all lengths. If a TF ceiling of 15 is used then if a word occurs 15 times in a page of 100 words, it would be considered spam. On the other hand, 14 occurrences of a single word in a text that is 40 words long would not be considered spam, even though it is highly likely that it is. So the smaller the page, the more spam can pass through the filter. However, one might reason that smaller pages are less prone to contain spam as the spam itself increases the size of the page.

Term frequencies are also normalized on page size, with respect to word count, so that longer texts do not automatically attain higher TF values. This is done by dividing the term count by the number of terms in the page but only after the TF ceiling cut-off, meaning that page sizes are never greater than fifteen times the number of unique words in the text.

3.5 Weights

Single words are used as features for all weight calculations. Joachims offers an explanation to why words are a good choice [1, p. 14]

> *The reason why words provide a reasonable granularity for representing documents might be found in the evolution of language. Words*

are the elements of the language where syntax and semantics meet. They are the basic building blocks that carry their own meaning. The vocabulary of a language is under constant development. Intuitively speaking, the composition and use of words is permanently optimized so that they encode an optimum of information.

Weights for each term t in document d are all calculated using Equation 3.2. DF and DF_g are the document frequency of the term in the collection and the global document frequency — based on a very large sample from the web — respectively.

$$w_{(t,d)} = \frac{DF}{DF_g} \cdot \frac{TF_d}{DF_g} = \frac{IDF_g^2 \cdot TF_d}{IDF} \tag{3.2}$$

The factor $\frac{TF_d}{DF_g}$ in the equation is the standard TFIDF value [1] and $\frac{DF}{DF_g}$ is a scaling factor that takes into account a features relative importance to the collection as compared to the entire web. Since both TF and DF are decimal values between 0 and 1 the weight given by Equation 3.2 can be larger than 1. Therefore, scaling is performed to put all weights in the interval 0-1 as well. Weights are rounded to a precision of 10 digits after the decimal. The mode used was the default mode for PHP function round — which is "PHP_ROUND_HALF_UP".

3.6 Stop words and stemming

The stop-words list that was used in this thesis is an aggregate of Armand Brahajs list [8] and words from the global IDF list with values less than 0.3 — such words appear in so large a fraction of all the web pages that they do not contribute much to the classification task.

No stemming was performed because the values in the global IDF table came from un-stemmed words and there was no practical way of reconstructing the table. This led to larger word bags and DF tables. But apart from increasing processing time, it is assumed that it had no significant impact on the results in this thesis.

3.7 Performance measurement

To put a value on the performance of the classification rule from the different approaches, precision and recall is used. There are many other ways of measuring performance of a classification rule [1, p. 27], though these are the more widely used in information retrieval and text categorization.

It is common to combine precision and recall into a single value to facilitate comparison of results. One is the F1 measure [9] and another is the PRBEP [1]. In this thesis both precision and recall is used and left as two distinct values. The reason for this is that they contain all the necessary information about the performance of the classifier; also the data loss that is associated with any merging of the two values is avoided.

Precision and recall will be calculated using the definitions in [1] which are:

$$Recall = \frac{f_{++}}{f_{++} + f_{-+}}, Precision = \frac{f_{++}}{f_{++} + f_{+-}} \tag{3.3}$$

In the equation above; f++ is the number of positive pages correctly predicted, f+- is the number of positive examples predicted to be negative and f-+ is the number of negative examples predicted to be positive. The results will only show how many positive and negative pages respectively were correctly predicted. In the case where not all pages are predicted correctly, the remaining pages are interpreted as having been predicted to belong to the other category. E.g. if 98 out of 100 pages are correctly predicted as positive then it is assumed that 2 of those positive pages were predicted to be negative.

Five-fold cross validation is used during training to determine performance of the trained classifier. There is built in functionality for this in LIBLINEAR [10]. The training data is divided into five sets and the classifier is trained five times using four sets — different groups of four sets each time — and then the trained classifier is tested against the set that was not used in training.

Chapter 4

Method

This chapter begins with describing how the web pages were selected and collected. Section 4.2 contains a brief note of an early experiment. Next follows an explanation of the different methods for selecting negative examples that have been developed and evaluated in the scope of this thesis. Section 4.4 comments the need for error tolerance and the method by which a good value for it can be found. Lastly the code that was developed and implemented during the work with this thesis is commented.

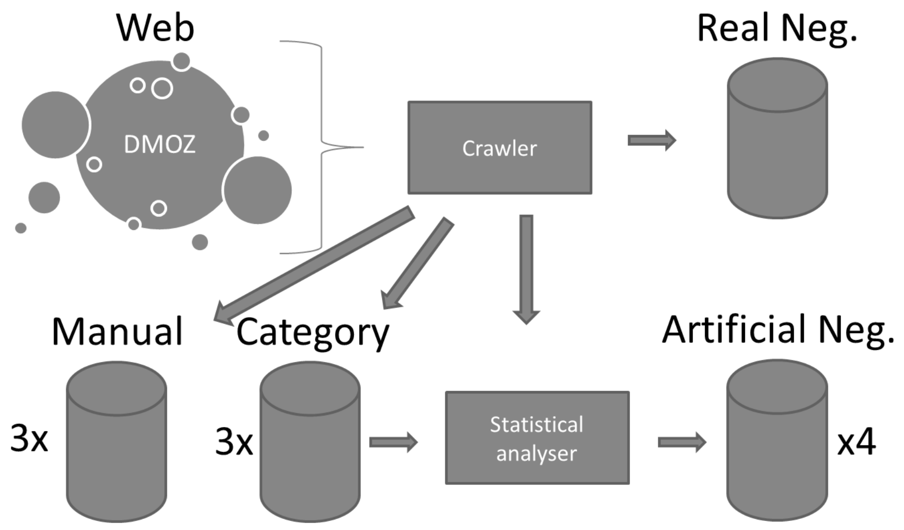


Figure 4.1: Overview of the implementation architecture

The above figure attempts to give an overview of how the practical parts of this project are setup. A crawler collects all required data from the Internet — more specifically the DMOZ site — and the statistical analyser generates the statics that are used for creating the artificial negative examples. The data is ordered into four different collection types:

- Real negative is the collection of 5000 random pages that are used directly as negative in one of the methods.
- Manual represents the collections of manually categorized webpages, 100 from each positive category, that are used in the evaluation of the classifier.
- Category represent the collections of positive examples (5000 from each positive category) used for training the classifier.
- Arificial negative represents the collections of artificially created negative examples (5000 for each method) used for training the classifier

4.1 Building the corpus

In the beginning of the project, some of the standard data sets were considered for use as positive examples. One advantage is that they are already classified and verified to what they are said to be. Another is that results can be compared with others that have done research using the same data sets. However, the problem specification did not allow the use of the available data sets as such data is too homogenous.

The well-known Reuters-21578[1] corpus only contains articles written by people who are used to writing whereas this thesis is concerned with just about any page that can be found on the web.

WebKB[2] is another data set that is often used. It is a collection of web pages from computer science departments from four universities. Despite the fact that they are web pages, they are still confined to a very specific and homogeneous region of the web.

Lacking a suitable corpus[3] one was created from scratch. Fortunately though, there are directories of manually labelled URLs on the web. One such, and the one used in this thesis, is the Open Directory Project at www.dmoz.org. The choice was motivated by several factors; it is open source nature, being well established and kept relatively up to date, easily parseable and available RDFs and recommendations.

There are 16 main categories, each of which contains a multitude of more specific categories. For this thesis, three categories were chosen as sources for positive examples and the pages were downloaded from the corresponding category on dmoz.org.

Thorough evaluation of the classifier requires positive examples from more than one category because different categories have different characteristics and may vary in how challenging they are to classify. Practical restrictions required limiting the number of categories however. The categories that were chosen are: Christianity, Martial-arts and Cooking — each of which are thought to offer different challenges. A more in-depth discussion of the choices can be found in (section 3.1).

[1] http://www.daviddlewis.com/resources/testcollections/reuters21578/readme.txt
[2] http://www.cs.cmu.edu/afs/cs/project/theo-20/www/data/
[3] Collection of training examples.

Each category corpus contains 5000 random pages from the URLs listed on the DMOZ site. Initially, all pages were crawled and the URLs were saved to a file. Every URL in the list that pointed to a start page, e.g. www.examen.com, was also crawled in order to extract all the internal URLs from that page[4]. Those URLs were then added to the complete list of URLs from which the 5000 pages were then drawn at random.

The reason for not including start pages is that they often do not contain much information that can be used in classification. They are often merely a list of links to the other pages on the site. This is purely based on observation from the initial, manual, search for candidate categories. There were, however, enough instances that displayed said characteristics to justify the decision — a rough estimate is that about 30% of the start pages were unusable.

4.2 Early testing

Initially there was hope of finding some way of characterizing categories. This was during the analysis of different categories before choosing the three to use in the tests. Since weights describe the importance of words for the categorization task, they seemed to be the best candidate for the characterization of categories. Therefore, the distributions of all weights in each collection of web pages from the three categories were compared in search of something distinguishable. After a few tests the idea was abandoned as no conclusive or even suggestive results were attained.

4.3 Tests

Fairly early in the project the idea of using artificial pages emerged. It came partly from the seemingly impossible task of compressing all the complement[5] categories on the web to a manageable number of pages for training the SVM. The challenge is to select enough such pages to cover the entire space — other than the positive category — without having to select all of them.

A statistical approach seemed like the most intuitive choice since statistics is all about capturing the characteristics of the many in as little and compressed data as possible — like one or two parameters for a distribution function. But finding examples that have the statistical properties needed would be as hard as finding examples that, together, encompass all the possible negative text on the web. One way, the reasoning went, was to create them from scratch. The decision to use artificial examples was further motivated and inspired by Sassano's paper [6].

Since the pages are converted to word bags before being presented to the learning algorithm, all semantic meaning is lost anyway. Therefore, when creating artificial examples there is no need to consider how meaningful the text will be

[4]URLs linking to other sites were ignored as the content of such sites may be outside of the category

[5]The complement refers to all categories other than the positive one.

and, consequently, there is no need to create a readable page — only a word bag representation of the, would be, artificial page.

Originally four methods were developed, each centered on the creation of artificial pages to represent the complement to the positive category. Initial results from tests using those methods prompted the addition of another method based on using randomly selected existing pages directly as negative examples according to a crude estimate of the number of categories on the web, and the number of pages per category. All weights were calculated using Equation 3.2 — see section 3.5.

The rest of this section contains a short description of each method for selecting negative examples that were tested in this project.

4.3.1 Random artificial weights

This method is based on the theory that the weight distribution of random pages from the web are more spread out than the weights of pages taken from the same category. Since the vocabulary is based on the positive collection alone, weights from positive pages should tend to have a higher number of higher values while weights from random pages should be more evenly[6] spread out. Positive pages will likely have a higher number of category specific features but also an average amount of common feature while random pages will only have the common features as a source for higher weight values.

By creating weight vectors with random weights drawn from a uniform distribution, those weights will also be more spread out, and will therefore capture the characteristic of random pages from the Web — albeit in a rather crude way.

The distribution of words in text can be approximated using Zipf's law [5, p. 145-147] and is clearly not uniform. Therefore, if nothing else, the artificial examples based on a uniform distribution will be very different from real pages coming from a specific category. Granted it is not the words that will be uniformly distributed but their weights. However, tests conducted on the collections used in this project suggest that weights will also follow Zipf's law — see Appendix A — Weight distributions.

Weights were generated using the php function mt_rand(). Each weight was assigned a pseudo random feature number that was selected from an array containing all the feature numbers in the vocabulary. The array was shuffled, using the php function shuffle, before creating each artificial page — thus using new, random features for each page. Of course, the feature vector also had to be sorted by feature as required by LIBLINEAR [10] — the implementation of the SVM-algorithm used in this thesis. Page sizes[7] were also chosen at random using mt_rand(). The values were drawn from a page length distribution, generated from all the positive example pages from all three positive collections — pages from the three positive categories — and all the pages from the random category as well.

[6]Relative the positive pages at least, not necessarily very even at all.

[7]Or equivalently: word bag sizes.

4.3.2 Artificial weights based on inverse distribution

With this method, the negative examples are built using weights drawn from the inverse of the positive weight distribution — which is obtained by simply subtracting each original weight from 1. Although it might be hard to interpret what it means to invert a weight distribution, intuitively it fits well with the notion of complement.

As explained above, the weights are drawn from the inverted weight distribution of the positive category. The aim is to do this randomly. However, a few restrictions apply. The document frequency (DF) is randomly drawn from a distribution created by taking all the document frequencies from all the four collections — the three positive ones and the random one. The DF dictates how many of the artificial vectors that are going to contain a certain feature which is also drawn randomly from a list of all features in the feature set. That is, after all, the definition of document frequency; the number of documents a certain feature appears in. The two remaining unknowns are the weight and the term frequency (TF). Since the whole point of this method is to use weights from a pre-constructed weight distribution (the inverted weights), they are drawn randomly from that distribution. The last remaining unknown is the TF which can be solved for, using Equation 3.2.

4.3.3 Artificial weights based on inverse distribution with extended feature space

This method is, in principal, the same as in 4.3.2. The only difference is that the feature space has been slightly extended so that the more common words on the Web as a whole is taken into account, even though they may not be very common — or even present — within the collection from the positive category. A maximum of 100,000 words with the lowest global IDF value are added to the feature set — common words have lower IDF values. Stop-words are not present in the global IDF table though. No check is done to see whether the word already exists in the feature set or not as the exact amount of additional words is not important and it involved too much computation.

4.3.4 Artificial weights based on statistics of real text

Again, weights are drawn randomly but this time from a weight distribution based on real examples of text. The distribution of 5000 random pages from the DMOZ website is calculated in the same way as with the three positive collections. The weights for the artificial vectors are then simply drawn from the distribution of the random pages collection. The rest of the implementation is identical to 4.3.1

4.3.5 Weights based on random existing pages

This method relies on the existing pages on the web in building the negative collection. It is assumed that the URLs on DMOZ make up a good enough representation of the content on the web as a whole[8]. Assuming they are, random pages from DMOZ should be representative of random pages form the web. Therefore, this method uses 5000 such pages as negative examples directly without any manipulation. The weight vectors are created in the same way as the positive vectors from the positive categories.

4.4 Model selection

Given the complexity of natural language, in the form of text, and the inherent ambiguity in most (if not all) categories we can come up with; it would be naÃŕve to assume that a web page with more than a few lines of text could ever be confined to a single category. Of course, it would be possible to construct a category that encompasses all text. Such a category would, however, not be very useful in the context of this thesis. As long as one considers more than one category — such that any text can be ascribed to at least one of them — web pages are bound to exist that cannot be uniquely categorized.

The reasoning above suggests that web pages, when ordered into categories, are not linearly separable. That is, there is no hyper plane that separates all pages into clearly defined and distinct categories. There will always be web-pages that mostly belong to Christianity, but also have a touch of cooking to them.

Teaching the SVM algorithm is essentially equivalent to solving an optimization problem where the distances from the web pages to the hyper plane are minimized. However, if it is not possible to place the hyper plane in such a way that all web pages reside on their respective side of the hyper plane then there is no solution to the problem. This is where the training error cost comes into the picture.

The introduction of the error cost parameter allows the SVM to make errors. It will be ok to ignore a few web pages that happen to fall on the wrong side of the hyper plane. The amount of errors, and consequently their magnitude, is determined by the value of the cost parameter. In this way, the optimization is both on the distance of the support vectors to the hyper plane, and the number of errors. Too high a tolerance will result in a classifier that predicts all pages to be positive while too low a tolerance will predict many positive pages as negative.

The best value for the error cost depends on the application and must therefore be empirically determined for each classifier. The way this is done in this thesis is by using a so-called grid search — as described by Hsu, Chih-Wei et al. [3] — where the best value is the one that gives the highest training accuracy.

4.5 Implementation

This section gives a quick overview of the actual implementation of the tests and the pre-processing of data. There are three major classes with distinct roles in different stages of development.

[8]Considering only "clean" categories, excluding ones like pornography which surely would shift the distribution of categories.

4.5.1 Setup

The programming language chosen for this thesis was PHP. It is easy to work with, especially for crawling because web pages can be handled like any other stream. Speed was never an issue either and familiarity with the language saved a lot of time as little or no learning was required.

Since the text categorization task is well suited for linear classification [1] [3], the LIBLINEAR [10] classifier is used. It is an implementation of SVM specifically designed and optimized for linear classification that is said to be faster than using a general SVM implementation.

Some of the graphs were generated directly in PHP with the help of the JpGraph library [11] while others were created with MatLab.

4.5.2 Crawler

One of the first practical tasks during the project was the collection of training and testing data. Since the thesis only deals with web pages, the work involved a lot of web interaction. There are crawlers available to use, but the decision was made to create one specific for this task.

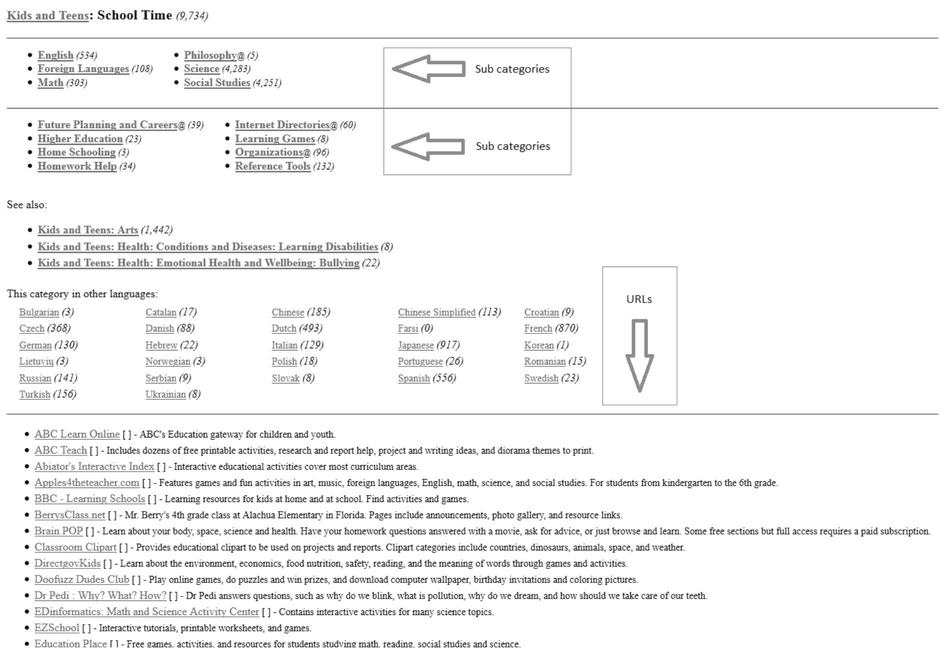
One of the reasons for this decision was that many aspects of the crawler needed to be tailored for this project. Learning how an existing crawler works and what it does and then modifying it to suit this project would have taken as long as developing one from scratch, it was thought.

The task was much more daunting than first estimated. Crawling is far from simple as it needs to be able to handle many different situations. Adaptive browsers allow for sloppy HTML coding which makes generalization of crawlers difficult to implement. HTML is not very strict to begin with. Then there are relative and absolute URLs, both of which must be handled by the crawler. Some tags, like scripts, need to be ignored to reduce noise levels in the texts — otherwise they would contribute with nonsensical programming data to the rest of the text on the page.

Each category on dmoz.org contains a list of sub-categories and a list of URLs. The URLs reside in an un-ordered list with the class attribute set to directory-url while the list containing the sub-categories has a class attribute with the name directory dir-col. The crawler looks for those class names and writes each URL to a file and each sub-category is appended to the end of the list of categories to be crawled — the first of which is the main category e.g. cooking.

4.5.3 SVM

This class contains all the methods necessary for creating training and testing files for the SVM. It also handles scaling and inverting of weights. Many of the functions also rely on the Page class that handles the "word-bag" representation of web pages.



Kids and Teens: **School Time** *(9,734)*

- **English** *(534)*
- **Foreign Languages** *(108)*
- **Math** *(303)*
- **Philosophy@** *(5)*
- **Science** *(4,283)*
- **Social Studies** *(4,251)*

- **Future Planning and Careers@** *(39)*
- **Higher Education** *(23)*
- **Home Schooling** *(3)*
- **Homework Help** *(34)*
- **Internet Directories@** *(60)*
- **Learning Games** *(8)*
- **Organizations@** *(96)*
- **Reference Tools** *(132)*

See also:

- **Kids and Teens: Arts** *(1,442)*
- **Kids and Teens: Health: Conditions and Diseases: Learning Disabilities** *(8)*
- **Kids and Teens: Health: Emotional Health and Wellbeing: Bullying** *(22)*

This category in other languages:

Bulgarian *(3)*	Catalan *(17)*	Chinese *(185)*	Chinese Simplified *(113)*	Croatian *(9)*
Czech *(368)*	Danish *(88)*	Dutch *(493)*	Farsi *(0)*	French *(870)*
German *(130)*	Hebrew *(22)*	Italian *(129)*	Japanese *(917)*	Korean *(1)*
Lietuvių *(3)*	Norwegian *(3)*	Polish *(18)*	Portuguese *(26)*	Romanian *(15)*
Russian *(141)*	Serbian *(9)*	Slovak *(8)*	Spanish *(556)*	Swedish *(23)*
Turkish *(156)*	Ukrainian *(8)*			

- ABC Learn Online [ ] - ABC's Education gateway for children and youth.
- ABC Teach [ ] - Includes dozens of free printable activities, research and report help, project and writing ideas, and diorama themes to print.
- Abiator's Interactive Index [ ] - Interactive educational activities cover most curriculum areas.
- Apples4theteacher.com [ ] - Features games and fun activities in art, music, foreign languages, English, math, science, and social studies. For students from kindergarten to the 6th grade.
- BBC - Learning Schools [ ] - Learning resources for kids at home and at school. Find activities and games.
- BerrysClass.net [ ] - Mr. Berry's 4th grade class at Alachua Elementary in Florida. Pages include announcements, photo gallery, and resource links.
- Brain POP [ ] - Learn about your body, space, science and health. Have your homework questions answered with a movie, ask for advice, or just browse and learn. Some free sections but full access requires a paid subscription.
- Classroom Clipart [ ] - Provides educational clipart to be used on projects and reports. Clipart categories include countries, dinosaurs, animals, space, and weather.
- DirectgovKids [ ] - Learn about the environment, economics, food nutrition, safety, reading, and the meaning of words through games and activities.
- Doofuzz Dudes Club [ ] - Play online games, do puzzles and win prizes, and download computer wallpaper, birthday invitations and coloring pictures.
- Dr Pedi : Why? What? How? [ ] - Dr Pedi answers questions, such as why do we blink, what is pollution, why do we dream, and how should we take care of our teeth.
- EDinformatics: Math and Science Activity Center [ ] - Contains interactive activities for many science topics.
- EZSchool [ ] - Interactive tutorials, printable worksheets, and games.
- Education Place [ ] - Free games, activities, and resources for students studying math, reading, social studies and science.

Figure 4.2: Example of a URL listing on DMOZ along with sub-categories. Screenshot from http://www.dmoz.org/Kids_and_Teens/School_Time

4.5.4 Analyser

The Analyser handles all statistical analysis of the data and can also build MatLab.m files for generating graphs like the ones in Appendix A.

Chapter 5

Results

In the very beginning of the thesis, an effort was made to try and find a way of characterizing categories based on their weight distributions. The idea was abandoned when the task proved too time consuming and would contribute little to the thesis. This chapter opens with the few results obtained from these investigations.

Initially four categorization rules were tested on three different categories. The results from the experiments were all very similar and showed a consistent low performance for all combinations of methods and categories — in that almost all pages, including the negative, were classified as positive.

However, further experimentation with the initial four methods yielded far better results. It seems that the grid search method was not enough to produce good classifiers. Instead, error cost values had to be found by trial and error. The results from the first four tests are presented in section 5.2.

The disappointing test results prompted the introduction of another rule where real, random pages were used as negative examples. The performance of the extra rule was a significant improvement over the others. The last section of this chapter is devoted to the results from this test.

To test the SVM, the code and the procedures used, a control classification was done using Christianity as the positive category and cooking as the negative. Weights were calculated using the feature set from Christianity only, and testing was done on the 100 manually classified positive web pages from both categories — where the ones from cooking were instead used as negative samples.

Fivefold cross validation, again, gave a high training accuracy of approximately 98% and the classification accuracy was 100% for both positive and negative documents — as expected. This suggests that the rest of the results are well founded.

5.1 Category characteristics

Initial tests show that, although there are slight differences in the center of gravity, larger collections of pages do indeed have similar weight distributions regardless of belonging to a specific category or being randomly chosen from the web. They also seem to follow Zipf's approximation rather well.

The maximum weight, as calculated with Equation 3.2, seems to differ greatly between the categories. No conclusive results were obtained, however, on whether or not a category can be defined by its weight distribution — as was initially the hope. Some efforts were made to also find a standard distribution that could be matched to the weight distributions but none was found and the work was abandoned before anything presentably was produced. Had there been though, it could have been used for generating random weights and perhaps used to characterize categories — like a fingerprint maybe.

5.2 Artificial samples

The grid search [3] method resulted in four methods having a training accuracy close to 100% for fivefold cross validation, and a recall of 1.0 (100 out of 100 positive webpages correctly classified). Precision, however, suffered greatly from the poor predictive accuracy on the negative samples as shown in tables Tables 5.1- 5.4. The columns "P/R" represent the precision and recall measurements. The error costs (C) are shown as powers of two because that is the way they are defined using the grid-search method in [3]. The two columns "Positive" and "Negative" refer to the number of correctly predicted pages and nSV is the number of support vectors from the training.

A classifier that behaves in a way as indicated by Tables 5.1- 5.4 may indicate that the error cost was too high during training since almost all pages are predicted to be positive. However, some experimentation with trial and error testing showed that the classifiers could indeed be trained to produce much better predictions simply by using a different value for the cost parameter. This means that a high training accuracy does not necessarily produce a classifier with high predictive accuracy — at least not for the methods tested here. The results from the trial and error testing are shown in Tables Table 5.5- 5.8.

Category	*P/R*	*Positive*	*Negative*	*C*	*Train. acc*	*nSV*
Christianity	0.329/1.00	100/100	5/209	$2^{-4.75}$	99.52%	6595
Cooking	0.333/1.00	100/100	9/209	2^{-3}	99.68%	5361
Martial-arts	0.328/1.00	100/100	4/209	$2^{-6.25}$	99.49%	7501

Table 5.1: Results from training and prediction test, using uniformly distributed random weights

Category	*P/R*	*Positive*	*Negative*	*C*	*Train. acc*	*nSV*
Christianity	0.328/1.00	100/100	4/209	$2_{7.5}$	99.64%	4717
Cooking	0.341/1.00	100/100	16/209	2_{8}	99.82%	4698
Martial-arts	0.330/1.00	100/100	6/209	$2_{4.5}$	99.80%	5395

Table 5.2: Results from training and prediction test, using weights from the weight distribution of random pages

Category	*P/R*	*Positive*	*Negative*	*C*	*Train. acc*	*nSV*
Christianity	0.324/1.00	100/100	0/209	2^{-6}	99.98%	5000
Cooking	0.324/1.00	100/100	0/209	2^{-6}	99.98%	4893
Martial-arts	0.324/1.00	100/100	0/209	2^{-8}	100%	5445

Table 5.3: Results from training and prediction test, using inverted weights from the positive examples

Category	*P/R*	*Positive*	*Negative*	*C*	*Train. acc*	*nSV*
Christianity	0.324/1.00	100/100	0/209	2^{-6}	99.98%	4938
Cooking	0.324/1.00	100/100	0/209	$2^{-6.5}$	99.98%	5453
Martial-arts	0.324/1.00	100/100	0/209	$2^{-8.25}$	100%	5341

Table 5.4: Results from training and prediction test, using inverted weights from the positive examples and extended vocabulary

Category	*P/R*	*Positive*	*Negative*	*C*	*Train. acc*	*nSV*
Christianity	0.877/1.00	100/100	195/209	$2^{-11.9}$	92.61%	9663
Cooking	0.935/1.00	100/100	202/209	$2^{-9.2}$	97.05%	8753
Martial-arts	0.552/1.00	100/100	128/209	2^{-10}	97.94%	9140

Table 5.5: Results from trial and error test, using uniformly distributed random weights

Category	*P/R*	*Positive*	*Negative*	*C*	*Train. acc*	*nSV*
Christianity	0.342/1.00	100/100	17/209	2_{13}	99.42%	3156
Cooking	0.346/1.00	100/100	20/209	2_{10}	99.78%	3991
Martial-arts	0.381/1.00	100/100	47/209	2_{25}	99.06%	2579

Table 5.6: Results from trial and error test, using weights from the weight distribution of random pages

Category	*P/R*	*Positive*	*Negative*	*C*	*Train. acc*	*nSV*
Christianity	0.719/1.00	100/100	170/209	$2^{-15.5}$	93.67%	10000
Cooking	0.541/1.00	100/100	124/209	2^{-12}	99.29%	7869
Cooking	0.952/0.990	99/100	204/209	2^{-16}	95.66%	10000
Martial-arts	0.360/1.00	100/100	31/209	2^{-13}	99.41%	9085

Table 5.7: Results from trial and error test, using inverted weights from the positive examples

Category	*P/R*	*Positive*	*Negative*	*C*	*Train. acc*	*nSV*
Christianity	0.820/1.00	100/100	187/209	$2^{-14.7}$	92.87%	9966
Cooking	0.943/1.00	100/100	203/209	$2^{-13.5}$	97.13%	9396
Martial-arts	0.680/1.00	100/100	162/209	$2^{-14.5}$	96.71%	9919

Table 5.8: Results from trial and error test, using inverted weights from the positive examples and extended vocabulary

5.3 Real samples

The extra method, using random pages directly as negative, resulted in a rule with a bit higher performance than the artificial ones. This was done prior to the realization that the original methods — using artificial negative samples — could be improved using error costs other than the ones which resulted in the highest training accuracy. The results are shown in Table 5.9.

Category	*P/R*	*Positive*	*Negative*	*C*	*Train. acc*	*nSV*
Christianity	0.941/0.960	96/100	203/209	93.35%	2^{7}	6215
Cooking	0.971/0.990	99/100	206/209	97.84%	$2^{12.5}$	2099
Martial-arts	0.962/1.00	100/100	205/209	97.03%	$2^{8.5}$	3512

Table 5.9: Results from extra test along with the cost parameters that gave highest training accuracy. The method for choosing error cost and training is the same as for the previous experiments.

The much improved results gave reason for concerns about the vector lengths of the pages to be predicted. As only the vocabulary generated by the positive training samples are used, there is a risk that too few of the words in that vocabulary also occur in pages that do not belong to the positive category. Such a scenario would result in empty vectors for the negative examples and non-empty for the positive ones. If that is the case, classification would be a trivial task as something is easily distinguishable from nothing. However, Figure B.2 (Appendix B — Vocabulary/Vector overlap diagrams) shows that approximately 80% of all vectors from the negative samples from Christianity share 50% or more of their features with the vocabulary. The same examination of the positive samples shows, as expected, that every vector has a 100% overlap with the vocabulary. The other categories show similar behavior.

Chapter 6

Summary and discussion

The chapter begins with a quick summary of the most important findings of this project which is followed by notes and comments regarding different aspects of the project. It ends with some conclusions based on the observations throughout the project.

6.1 Summary

The main focus has been on testing artificially created negative examples for training SVM for use in classification of web pages where the positive category is known and there are labelled positive examples. Four methods were developed and tested and neither performed at an acceptable level based on the optimal training performance but needed to be trained using a trial and error approach.

Another method for selecting negative examples was tested that used random pages from the web directly as negative examples. Better results coupled with less work marks this method as the one recommended by this thesis for use in commercial applications — at least if one of the methods must be chosen.

Code has been written for crawling and downloading web pages as text files, converting them to weight vectors and generating negative examples using all of the methods described. There is also some code for analysing statistical properties of the data produced.

6.2 Index pages

The positive set was built using the categorization on DMOZ. Many of the links harvested points to the index page of a larger site. One concern is that most web sites include a number of standard pages, such as a page with contact information, a page about the authors and the site itself, etc. Compared to the number of pages actually about the subject, the standard ones may constitute a significant portion of all pages on that site. This has not been investigated, but poor results might indicate it.

6.3 A note on vocabulary size

In hindsight, it would probably have been better to use the extended vocabulary in all methods. This is suggested by the results from the inverse methods, where using the extended vocabulary performs significantly better in two out of three tests (cooking and martial-arts).

The attempt of characterizing categories might also have benefitted from an extended vocabulary because the idea was that there should have been a clearer difference between the weight distributions of the collections. Such characterization may only be relevant in a relative context where only a defined set of categories are used, as opposed to a general characterization.

6.4 Using artificial weights

As there is a trade-off between precision and recall and the relative importance between them vary with the application, what is considered high performance depends on the application. In the tests conducted with the artificial weights, recall was given highest priority. The error costs in the tables are those that gave the highest possible precision while still maintaining a perfect recall. In some cases, it would perhaps have been better to compromise a little, sacrificing 1-2% from recall in order to gain 10-20% in precision. The inverse method for cooking is an example of this, which is also why there are two entries in Table 5.7.

The method based on the weight distribution of random pages seems to produce worse results in all cases. For Christianity, increasing the cost parameter initially gave higher precision but any higher than the value in Table 5.6 would only give the same result. For martial-arts, precision improved very slowly with a growing error cost while number of support vectors decreased quite fast as well.

Also worth noting is the relatively high values of the cost parameter that is needed for the random pages distribution to manage even just a few correct negative predictions. The results from the trial and error also show a radical difference in the number of support vectors compared to the other methods. Some of the methods even produced 10,000 support vectors, which means that all examples ended up being support vectors. It is usually an indication of over fitting, meaning that it does not generalize well to other data. Indeed, while the inverse method performed well on cooking and Christianity, it did not do well on martial-arts.

In general; all methods based on artificial examples, except the one based on weights from random pages, needed extremely small values of the cost parameter. It means that high error tolerance was required which in turn suggests that artificial examples and real examples are not linearly separable. This might be due to the fact that the artificial negative examples are so spread out in the feature space that they cover the area occupied by the positive examples as well.

6.5 Using existing pages as negative

It was not the main focus of this thesis, yet it is very interesting how well the classifier, that was trained using just random pages from the DMOZ site, performed.

Considering the relative ease with which the pages were obtained, it is a very viable candidate for use in commercial applications. Furthermore, it is not unlikely that the same set of negative examples could be used with any positive category seeing as how few positive pages would appear among those random negative.

6.6 Training vs. prediction accuracy

As observed in the results in section 5.2; the cost parameter that gives the highest training accuracy is not necessarily the one that yields the best classification rule — at least not for the methods involving artificially generated weight vectors. The results also show that when using existing pages as negative examples, the penalty value attained with the grid search method indeed produces the best — or at least one of the best — classification rules.

With the artificial pages, the classifier was trained using examples of something that it will never encounter when used for prediction. The artificial pages represent only a statistical model of real pages. Therefore, the high training accuracy perhaps only reflects the classifiers performance in an environment where the positive and negative pages are as different as a real intelligible text is from a statistically representative mass of words. Perhaps it is no wonder then that a classifier that has been trained using artificial pages — and have attained high training performance — performs poorly when faced with real world text.

Another possibility is, of course, that the statistical properties of the artificial texts do not reflect the properties of real text. It is surprising though that the artificial weights drawn from the distribution based on real text performs so poorly, while the uniformly distributed weights show, relatively, much better results.

6.7 Conclusions

Creating artificial negative weight vectors was an interesting and educational experience but it is questionable whether the methods presented in this report can be used in practice. The uniform and the inverse extended methods gave best results with the uniform being the easiest to implement. However, if further test show consistent high performance of classifiers trained with random pages as negative, then there really is no reason to even consider artificial generation — at least not based on any results presented in this thesis.

Chapter 7

Future work

The most interesting potential lies with the method that uses random pages directly as negative. A few things that might be worth investigating are:

- Whether or not the same negative examples can be used for many different positive categories. Perhaps even if those negative examples contain a few positive ones, eliminating any need for human inspection.
- Conduct more tests with more positive categories.
- Other ways of obtaining a fair spread of different categories. Perhaps just entirely random pages from the web or from some directory listing like DMOZ.

Artificial examples might be appropriate with more heterogeneous texts. It would be interesting to see how well the methods based on artificial examples would fair against other methods by testing them on the standard data sets mentioned in the report. It is also possible that the loss of meaning with artificial pages has a negative effect on the training despite the fact that meaning is lost any way due to working with word bags. Testing the same methods with n-grams[1] could yield better results if that is the case. Another aspect that was touched upon is the statistical properties of pages when grouped into categories. A few questions that could yield potentially valuable results are:

- Is there any connection between category and weight distribution for pages from that category?
- Can weight distributions be mapped onto standard distributions and thereby characterizing categories by parameters to distribution functions?
- How do TF-ceilings affect weight distributions or texts in general?

Finally, even though it has been said that the linear kernel is well suited for text classification perhaps when artificial examples are involved the radial kernel, or maybe some other one, could improve predictive accuracy.

[1]Instead of using single words, two or more (n) words are used as entities. This results in a larger vocabulary but takes into account relationships between words.

References

[1] **Joachims, Thorsten**. *Learning to Classify Test Using Support Vector Machines: Methods, Theory and Algorithms.* Kluwer Academic Publishers, 2002.

[2] **Vladimir, Vapnik Naumovich**. *Statistical Learning Theory.* John Wiley & Sons, 1998.

[3] **Hsu, Chih-Wei, Chang, Chih-Chung and Lin, Chih-Jen**. *A Practical Guide to Support Vector Classification.* Taipei, Taiwan. Department of Computer Science, National Taiwan University, April 15, 2010.

[4] **Joachims, Thorsten**. *SVMlight Support Vector Machine.* University of Dortmund, Informatik, AI-Unit, August 14, 2008. `http://svmlight.joachims.org/`

[5] **Baeza-Yates, Ricardo and Ribeiro-Neto, Berthier**. *Modern Information Retrievel.* ACM press, 1999.

[6] **Sassano, Manabu.** *Virtual examples for text classification with Support Vector Machines.* Stroudsburg, Association for Computational Linguistics, 2003. pp. 208-215.

[7] **Yu, Hwanjo, Zhai, ChengXiang and Han, Jiawei.** *Text classification from positive and unlabeled documents.* New York : ACM, 2003. CIKM '03 Proceedings of the twelfth international conference on Information and knowledge management. pp. 232 - 239.

[8] **Brahaj, Armand**. *List Of English Stop Words.* April 14, 2009. `http://armandbrahaj.blog.al/2009/04/14/list-of-english-stop-words/`.

[9] **Yu, Hwanjo, Zhai, ChengXiang and Han, Hiawei**. *Text classification from positive and unlabeled documents.* New York, NY, USA : ACM, 2003. pp. 232 - 239.

[10] **Wang, Xiang-Rui; Chang, Cho-Jui; Fan, Rong-En; Yuan, Guo-Xun; Yu, Hsiang-Fu; Huang, Fang-Lan; Lin, Chih-Jen**. *LIBLINEAR – A Library for Large Linear Classification.* version 1.7, Machine Learning Group at National Taiwan University, April 1, 2011. `http://www.csie.ntu.edu.tw/~cjlin/liblinear/`.

[11] **Asial Corporation**. *JpGraph.* Asial Corporation. `http://jpgraph.net/`.

Appendix A

Weight distributions

The following distribution graphs show that the weight distributions seem to follow Zipfs law [5, p. 145] and that the weight distribution is effected to different degrees by the TF-ceiling, depending on the collection. Christianity is least disturbed by the TF constraint.

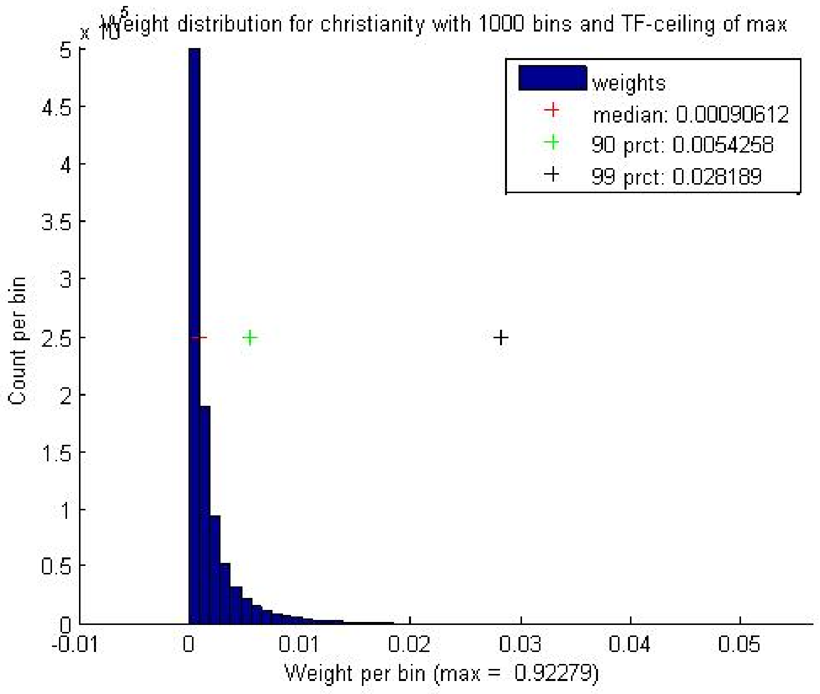


Figure A.1: Christianity with no TF ceiling

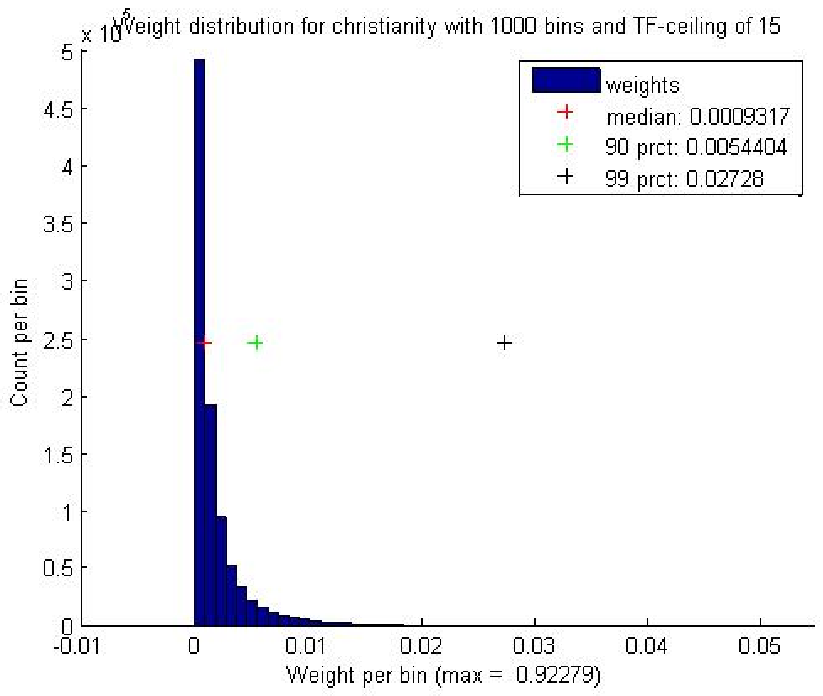


Figure A.2: Christianity with a TF ceiling of 15

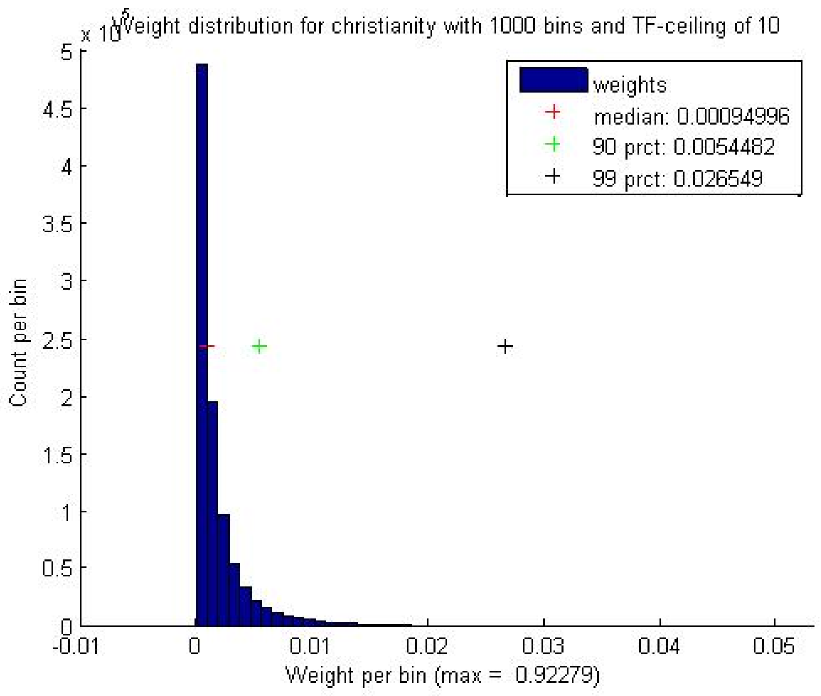


Figure A.3: Christianity with a TF ceiling of 10

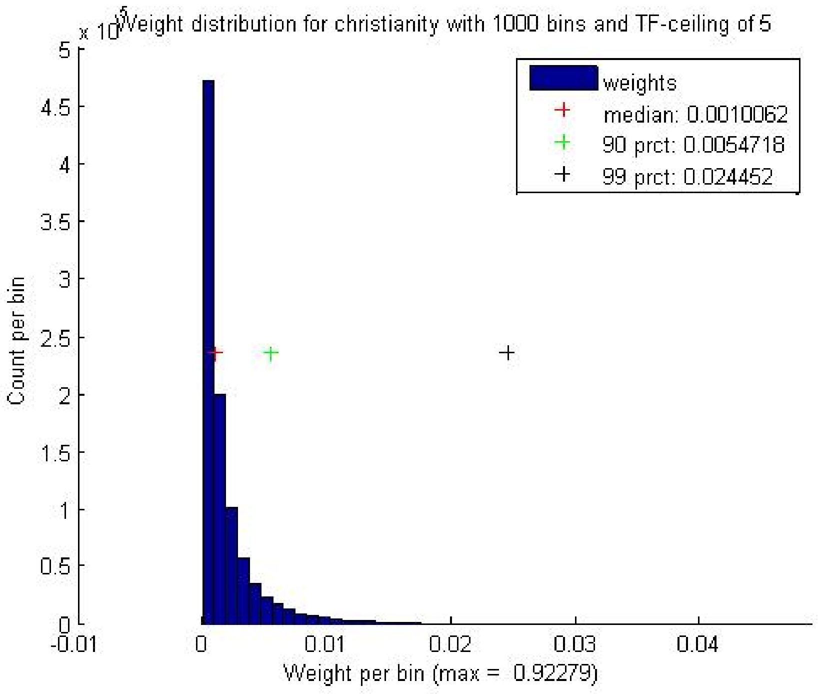


Figure A.4: Christianity with a TF ceiling of 5

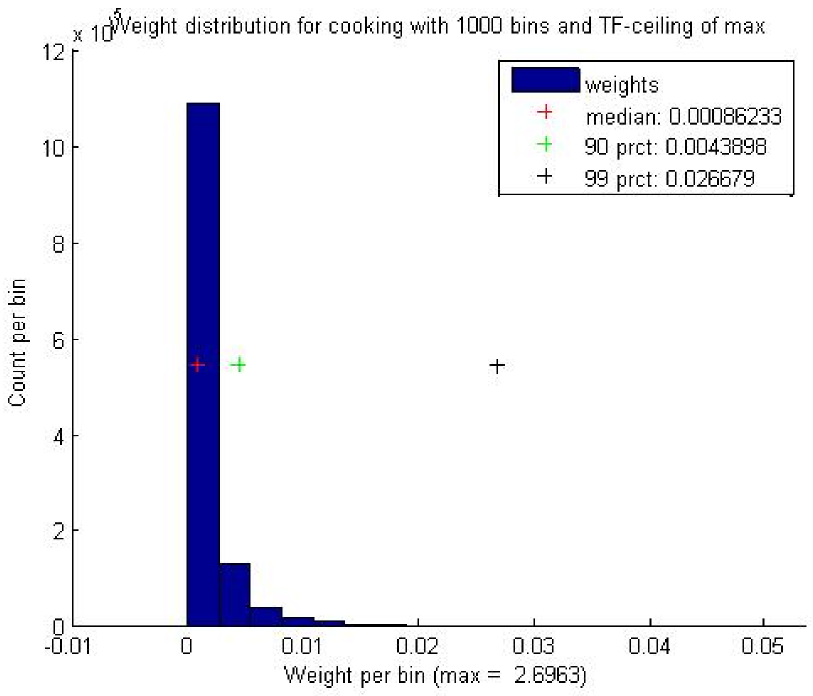


Figure A.5: Cooking with no TF ceiling

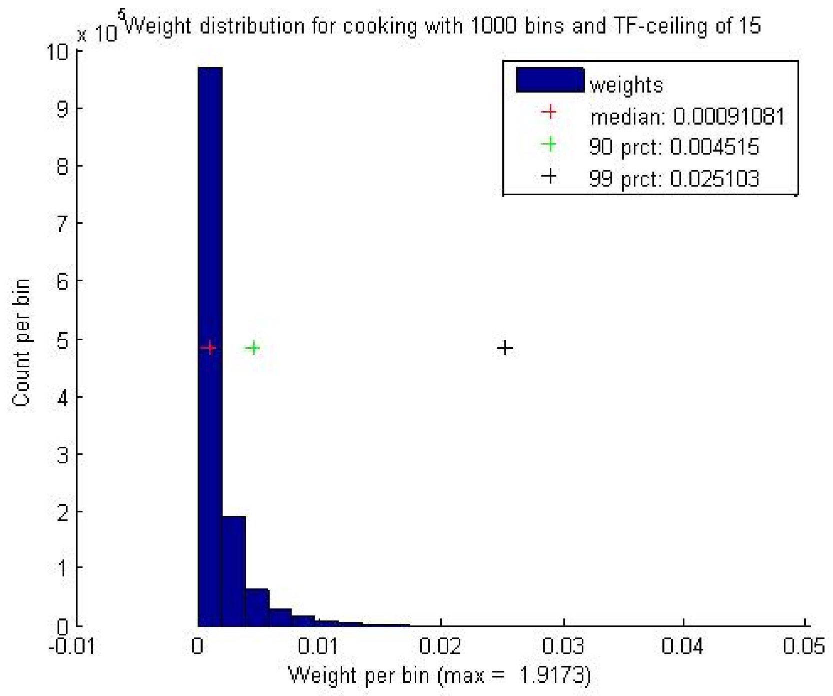


Figure A.6: Cooking with a TF ceiling of 15

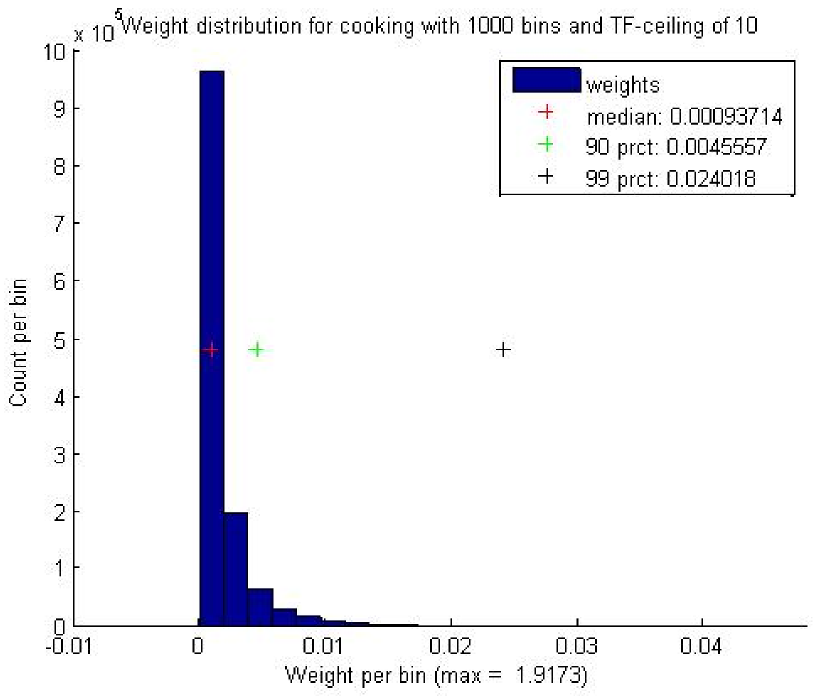


Figure A.7: Cooking with a TF ceiling of 10

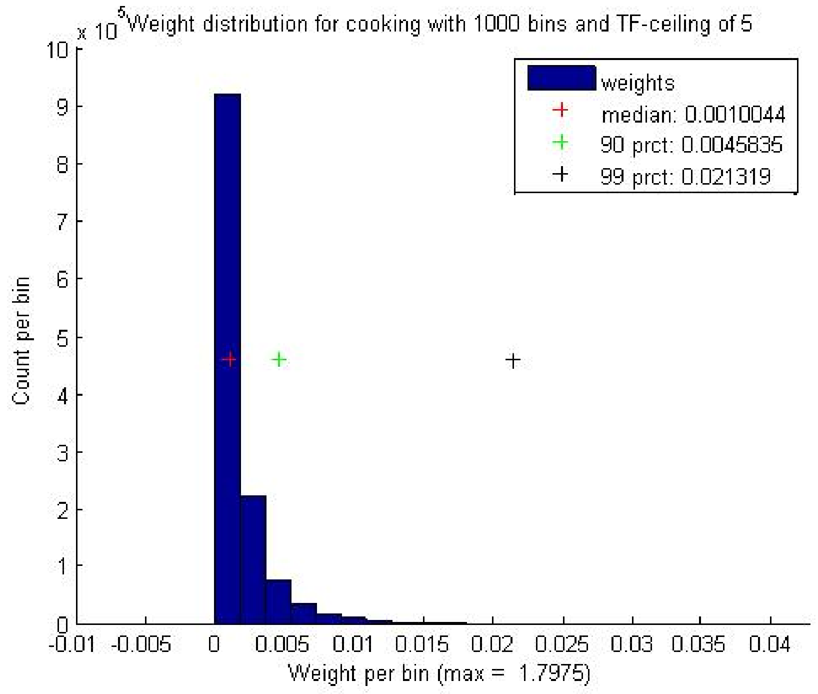


Figure A.8: Cooking with a TF ceiling of 5

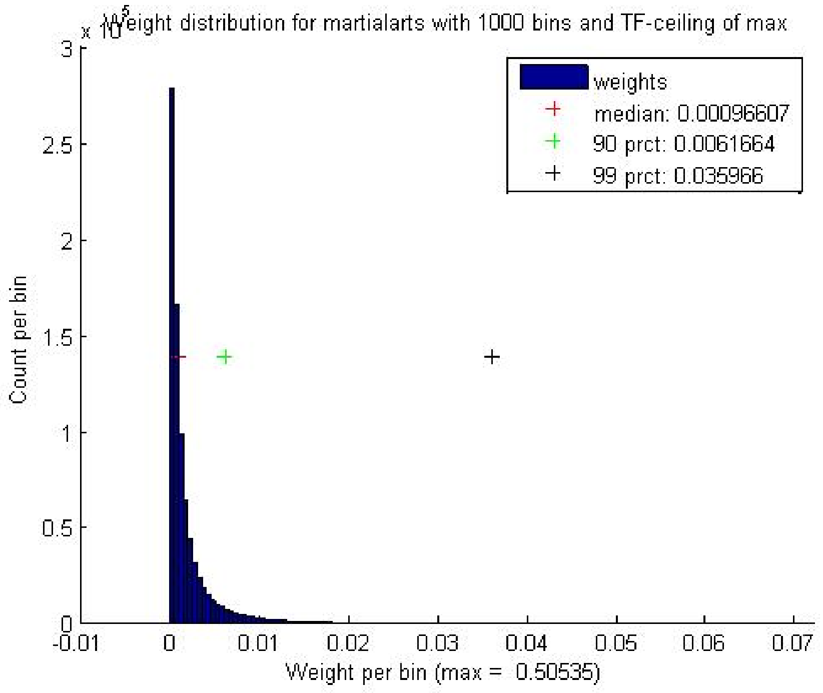


Figure A.9: Martial-arts with no TF ceiling

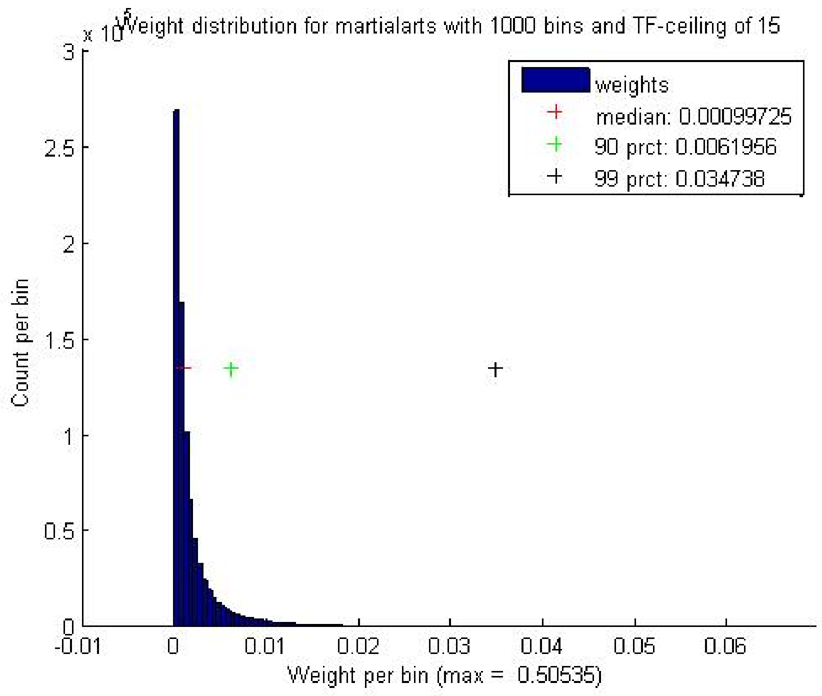


Figure A.10: Martial-arts with a TF ceiling of 15

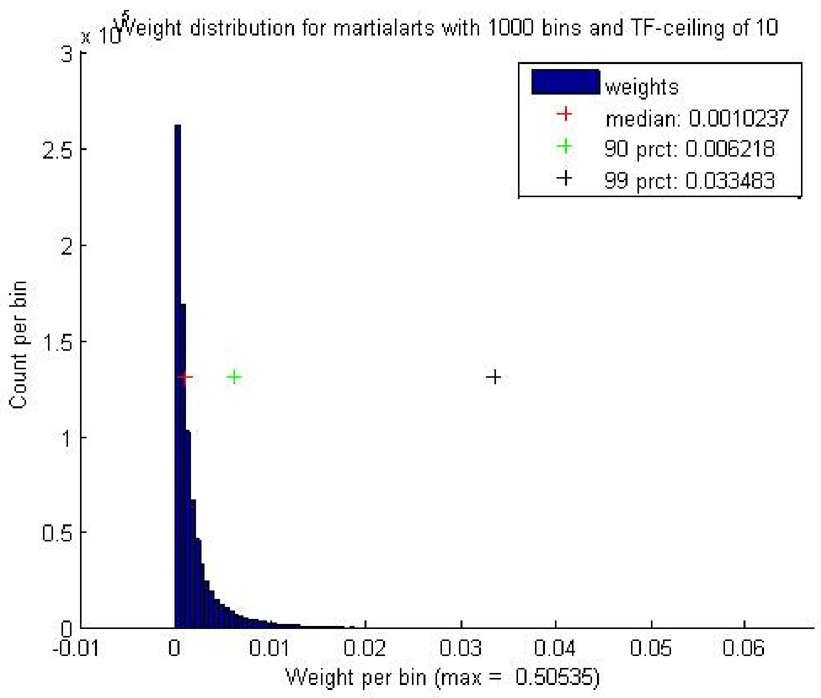


Figure A.11: Martial-arts with a TF ceiling of 10

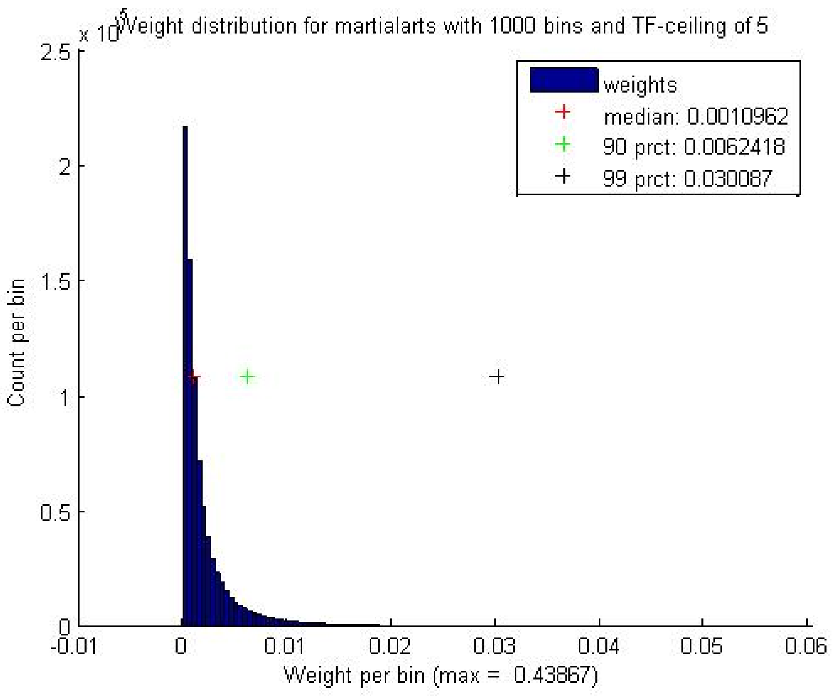


Figure A.12: Martial-arts with a TF ceiling of 5

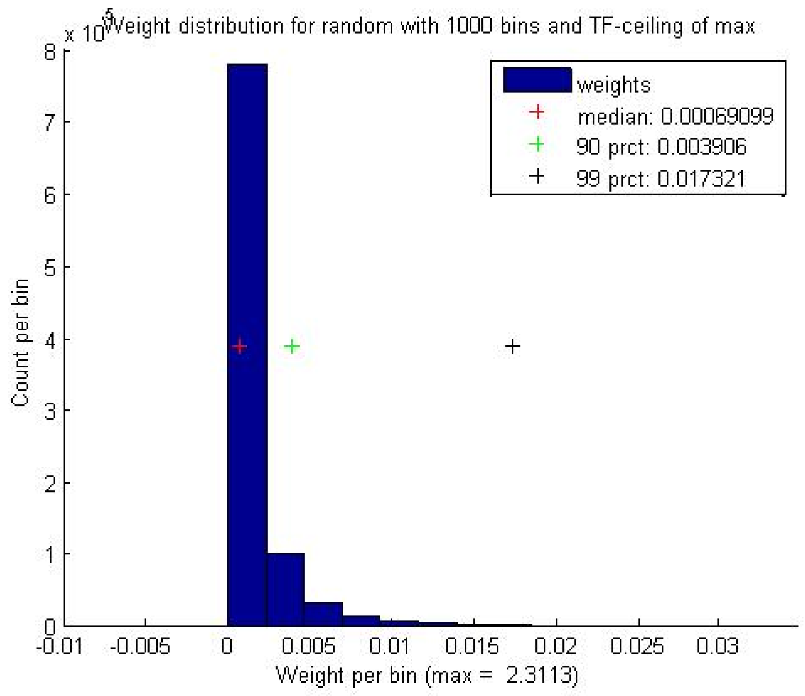


Figure A.13: Random pages with no TF ceiling

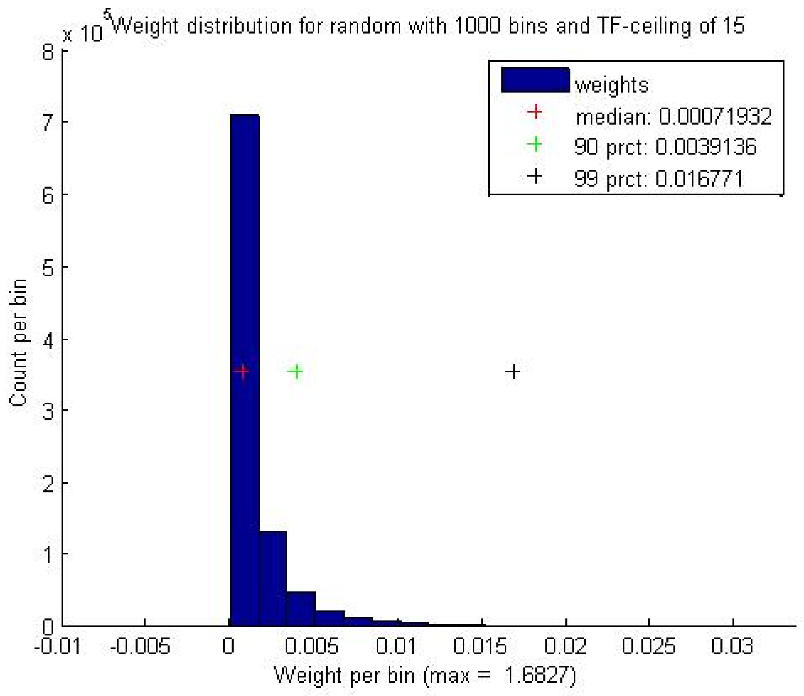


Figure A.14: Random pages with a TF ceiling of 15

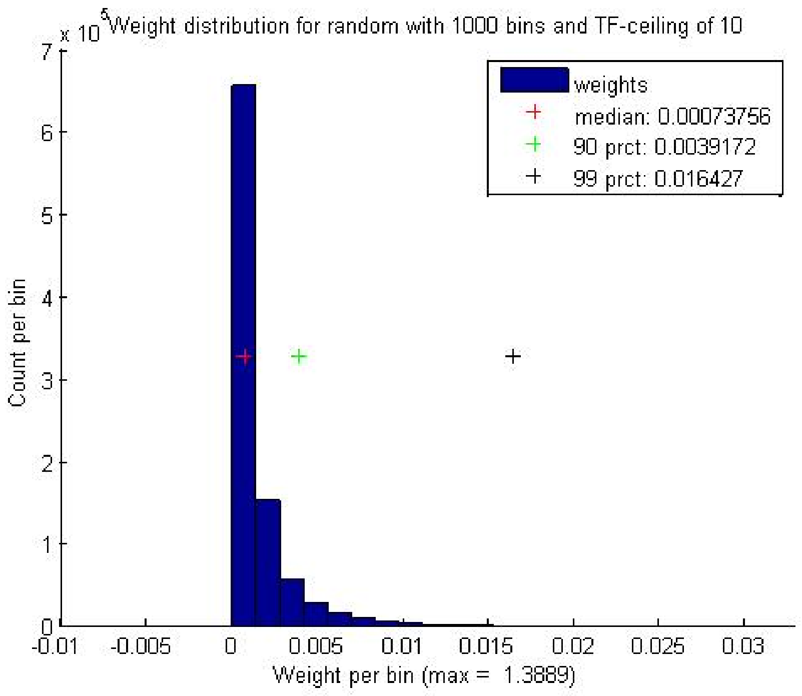


Figure A.15: Random pages with a TF ceiling of 10

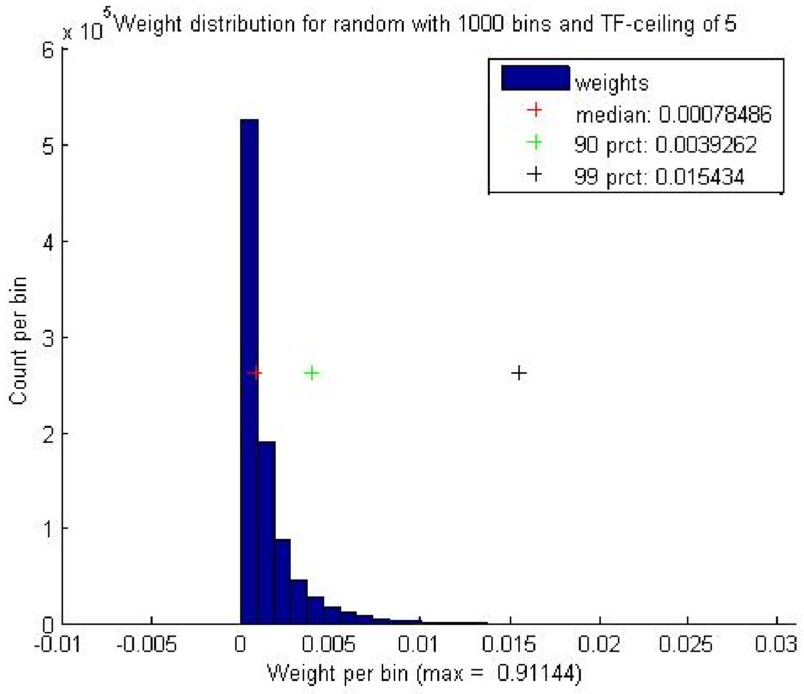


Figure A.16: Random pages with a TF ceiling of 5

Appendix B

Vocabulary/Vector overlap diagrams

The diagrams show the overlap, in per cent, between the weight vectors and the vocabulary when using random web pages directly as negative. Figure B.1 shows 100% overlap for all positive vectors from Christianity. This is expected since the diagram only shows the positive vectors, and the vocabulary is based only on the words found in the positive pages. This is the case for all three positive categories so those diagrams are omittes. Figures B.2- B.4 show the overlap between the vocabulary and the negative vectors from the random pages collection.

Clearly, the high performance of the classifier that was trained with random pages used directly as negative examples cannot be attributed the existence of zero-length weight vectors since there are none — or negligibly few any way.

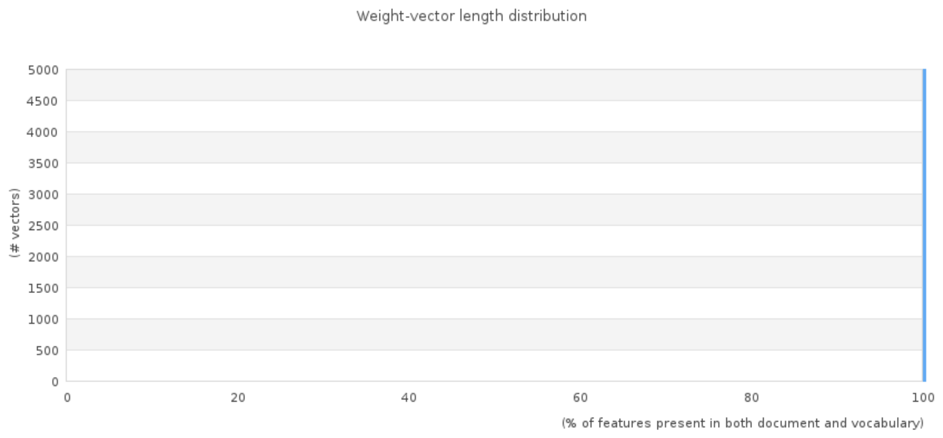


Figure B.1: The fracation of positive vectors from Christianity that overlap with the vocabulary

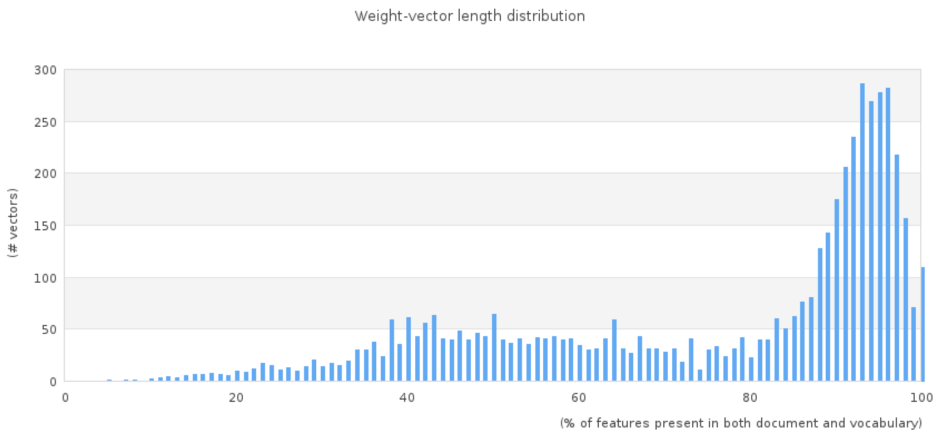


Figure B.2: The fracation of negative vectors (from the random pages collection) that overlap with the vocabulary from Christianity

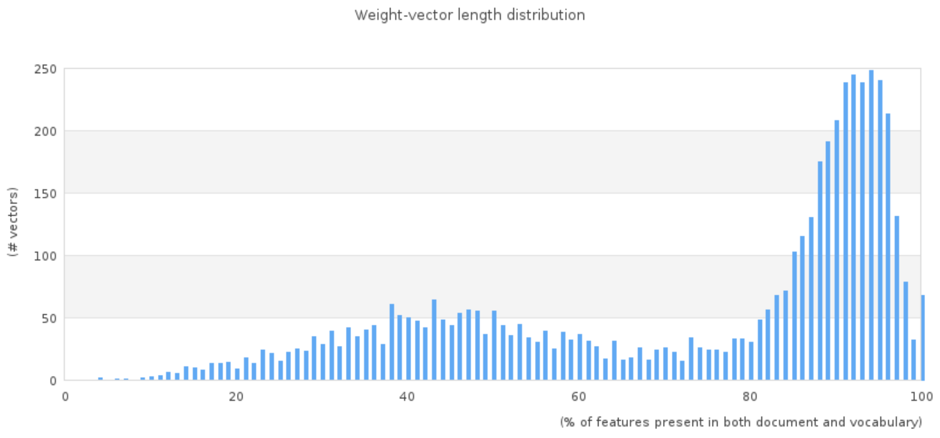


Figure B.3: The fracation of negative vectors (from the random pages collection) that overlap with the vocabulary from Cooking

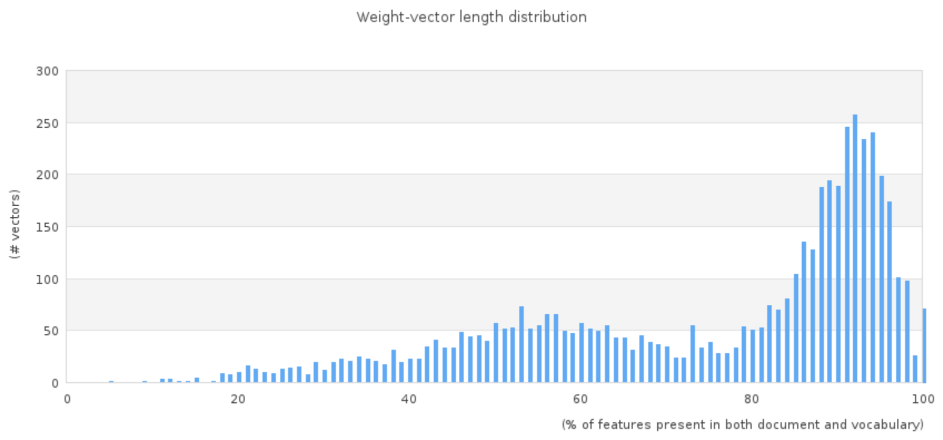


Figure B.4: The fracation of negative vectors (from the random pages collection) that overlap with the vocabulary from Martial-arts