Sirindhorn International Institute of Technology
School of Information, Computer, and Communication Technology

Statistical/Rule-Based Hybrid Method For DOM-Based Auxiliary Web Image Filtering

A Thesis in
Information, Computer, and Communication Technology
by
Tewson Seeoun

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Bachelor of Engineering

March 2010

Abstract

The rapid growth of the internet has significantly increased the number of multimedia contents available on web pages, and the most common form of them, besides text, is image. Images are used in websites for different purposes, e.g., to convey information and support the text, to draw attention, or to decorate the web pages. These purposes can specify the importance of web images in their applications. In applications that information is more important than visual joy, e.g., web page printing, web content indexing, or mobile web browsing, only informative images are needed. In this work we present a real-time method to filter out auxiliary web images. Images are classified according to their visual and contextual features. The visual features taken into account are based on how the image is displayed, e.g., position, size, etc. These features can be extracted in real-time. The contextual features mainly are distinct features for only images on the web, i.e., how they are used, or not, to link to other web pages. Unlike previous attempts on web image classification, we do not rely only on the HTML source code. Our method relies on the information rendered by a web browser to cope with modern website design manner that separates the design from the content in HTML files. An implementation is done as an extension for a web browser. It includes a support vector machine classification model trained from a data set of more than 1,700 web images. The model is evaluated by 10-fold cross-validation method to have an accuracy of 96.59%.

Table of Contents

List of Tables iv

List of Figures v

Acknowledgments vi

Chapter 1. Introduction 1
- 1.1 Structure of a Web Page 1
- 1.2 The Roles of Images 3
- 1.3 Removing Auxiliary Images 3
  - 1.3.1 Web Page Printing 3
  - 1.3.2 Seach Engine Indexing 4
  - 1.3.3 Mobile Web Browsing 4
- 1.4 Related Work 4

Chapter 2. Methodology 6
- 2.1 Features for Image Representation 6
- 2.2 Feature Extraction 7
- 2.3 Classification 8
  - 2.3.1 Support Vector Machines (SVMs) 8
  - 2.3.2 LibSVM 10
  - 2.3.3 Grid Search Technique 12
  - 2.3.4 Data Collection and Preprocessing 12
- 2.4 Implementation 12
  - 2.4.1 Server-Based Approach 13
  - 2.4.2 Client-Based Approach 13
- 2.5 Optimization 14

Chapter 3. Results and Discussion 15

Chapter 4. Conclusion 17

References 18

Appendix A. Software Download 19

Appendix B. Publications 20

List of Tables

3.1 Confusion Matrix of Average Prediction Results 15

List of Figures

1.1 Example of Document Object Model Rendered by Chrome Browser . . . 2
1.2 Comparison of an actual web page and a desirable printed page. 4
1.3 A page of the New York Times from Opera Mini web browser; advertisements can be noticed at the top of the page. 5

2.1 Possible solutions for data classification. Source: Christopher Manning, *Slides for Introduction to Information Retrieval: Lecture 14* (Stanford University, 2009), 5. 8
2.2 Example of nonlinear SVMs. Source: Christopher Manning, *An Introduction to Information Retrieval* (Cambridge University Press, 2009), 331. 11
2.3 Demonstration of firefox extension 13
2.4 Distribution of image dimensions. 14

3.1 Classification results from an article page and a tabular page 16

Acknowledgments

I would like to thank to my project advisor, Assist. Prof. Dr. Toshiaki Kondo for his time and guidance. His teaching in Digital Image Processing and Computer Vision classes inspired me to work on a pattern recognition topic. His kind comments on reports and presentations are valuable.

Besides, I would like to thank to Dr. Choochart Haruechaiyasak who has been my advisor since the internship and throughout the project. Without him and his guidance, this work would never be done and I would never get to Beijing.

Finally, The financial support from Young Scientist and Technologist Programme is gratefully acknowledged.

Chapter 1

Introduction

Internet has become the main source for information consumption. Information can be found through many interfaces, for examples, web pages, RSS feeds, emails, video clips, or podcasts. However, the most common method is to read from web pages. A large amount of multimedia content helps making information consumption easier and faster, but sometimes it can be unnecessary. This work proposes a way to classify unwanted images as one format of multimedia content by first describing in this chapter the way a web page is contructed, how that can be useful for such classification, and in which applications that some images should be removed.

1.1 Structure of a Web Page

A web page is composed of three main components, a HyperText Markup Language (HTML) document, a Cascading Style Sheets (CSS) document, and a JavaScript (JS) document. For modern website design, the HTML document defines the structure of information in the web page, e.g., the order of appearance, or parent-child relationships. The following HTML example code represents a simple web page consisting of a paragraph, an ordered list, and an image, respectively.

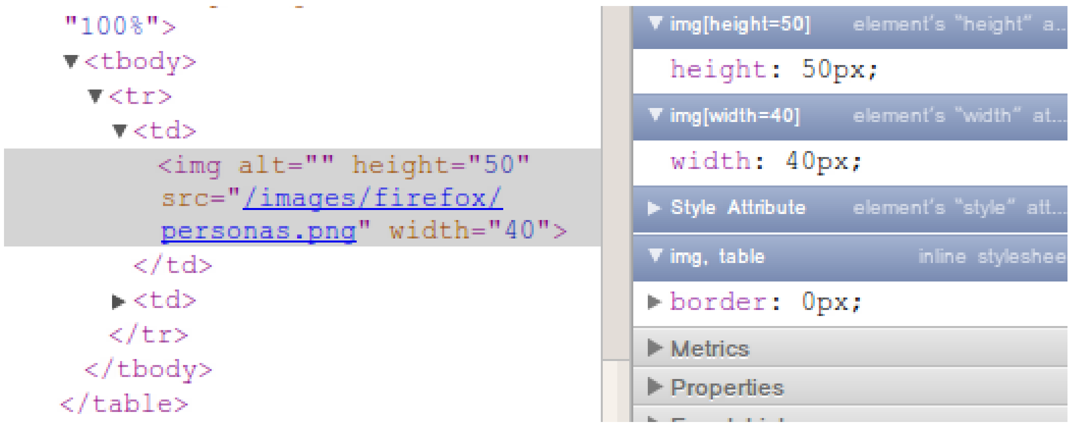
```
<html>
    <body>
        <p>
            This is a paragraph.
        </p>
        <ol>
            <li>Item 1</li>
            <li>Item 2</li>
        </ol>
        <img src=``IMAGE_PATH'' id=``img01''/>
    </body>
</html>
```

The CSS document, on the other hand, defines the appearance of each element in the web page, e.g., position, size, or color. The following CSS example code decorates the above HTML structure.

```
body {
    margin: 0px;
    padding: 0px;
}
p {
    color: green;
    font-size: 12pt;
}
ol {
    font: italic;
}
ol li {
    padding-left: 20px;
}
img {
    width: 50%;
}
```

JavaScript is then optionally used to control both structure and appearance of elements. The following JavaScript example code shows the image's path in a new dialogue box.

```
x = document.getElementById(``img01'');
alert(x.src);
```



```
"100%">
▼<tbody>
  ▼<tr>
    ▼<td>
      <img alt="" height="50"
      src="/images/firefox/
      personas.png" width="40">
    </td>
    ▶<td>
  </tr>
</tbody>
</table>
```

```
▼ img[height=50]    element's "height" a...
height: 50px;
▼ img[width=40]     element's "width" at...
width: 40px;
▶ Style Attribute   element's "style" att...
▼ img, table        inline styleshee
▶ border: 0px;
▶ Metrics
▶ Properties
```

Fig. 1.1: Example of Document Object Model Rendered by Chrome Browser

These documents are rendered by a web browser into a tree-like data structure called Document Object Model (DOM). As shown in Figure 1.1, The DOM consists of elements which possess different properties and behaviors. Among these elements, an image is usually represented by an HTML tag <IMG>. Thus, we can use JavaScript to extract information from web images.

1.2 The Roles of Images

Images in a web page can be used in different ways. Normally they are displayed, alone or with a piece of text, as a container of certain information, or as an excerpt of a larger collection of information, such as an image gallery or a video. For these uses, an image is considered to be 'informative', or to contain a piece of information that is important to its page.

On the other hand, images are also used for the sake of aesthetic. Website designers use buttons and banners instead of text to draw better attention and to create faster understanding. They also dress the page up with images as borders and frames. All these functions are important for normal uses of websites, but some of them can be unnecessary in other applications.

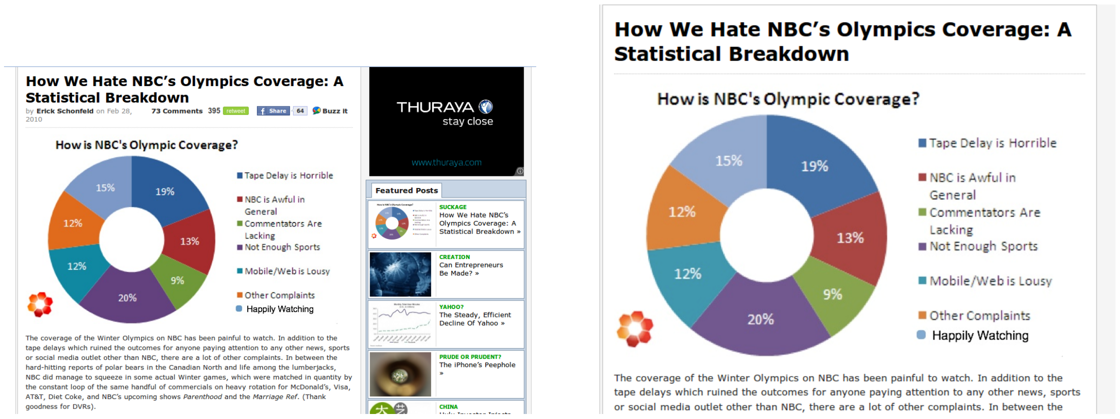
1.3 Removing Auxiliary Images

Considering about information-centric applications with a limited amount of resource, dealing with images that do not contain useful information is not desired. There are at least three applications as following that can benefit from ignoring auxiliary images.

1.3.1 Web Page Printing

Despite the popularity of handheld displays, reading from paper is still the most comfortable way. However, a web page full of advertisements and buttons can waste a significant amount of ink when printed.

Figure 1.2 shows the ideal concept of a printed web page that contains only important information. There have been several companies offering such web page optimization service lately, such as PrintFriendly [3] (starting in May, 2009) which also lets users to convert an optimized page into a PDF file, and Hewlett-Packard [1] which provides an add-on for a web browser for manual cropping and printing of a web page.



How We Hate NBC's Olympics Coverage: A Statistical Breakdown
by Erick Schonfeld on Feb 28, 2010 73 Comments 395 retweet Share 64 Buzz It

The coverage of the Winter Olympics on NBC has been painful to watch. In addition to the tape delays which ruined the outcomes for anyone paying attention to any other news, sports or social media outlet other than NBC, there are a lot of other complaints. In between the hard-hitting reports of polar bears in the Canadian North and life among the lumberjacks, NBC did manage to squeeze in some actual Winter games, which were matched in quantity by the constant loop of the same handful of commercials on heavy rotation for McDonald's, Visa, AT&T, Diet Coke, and NBC's upcoming shows *Parenthood* and the *Marriage Ref*. (Thank goodness for DVRs).

SUCKAGE
How We Hate NBC's Olympics Coverage: A Statistical Breakdown »

CREATION
Can Entrepreneurs Be Made? »

YAHOO?
The Steady, Efficient Decline Of Yahoo »

PRUDE OR PRUDENT?
The iPhone's Peephole »

CHINA

How We Hate NBC's Olympics Coverage: A Statistical Breakdown

The coverage of the Winter Olympics on NBC has been painful to watch. In addition to the tape delays which ruined the outcomes for anyone paying attention to any other news, sports or social media outlet other than NBC, there are a lot of other complaints. In between the

(a) An article on the web. (b) A desirable printed page.

Fig. 1.2: Comparison of an actual web page and a desirable printed page.

1.3.2 Seach Engine Indexing

For a search engine to be able to quickly locate a document containing a requested piece of information, it needs to create an index for data it has found through *crawling*. This problem may not be relevant to a search engine as large as Google's, but the waste of processing power and database space should be avoided for a smaller or personal knowledge management tool.

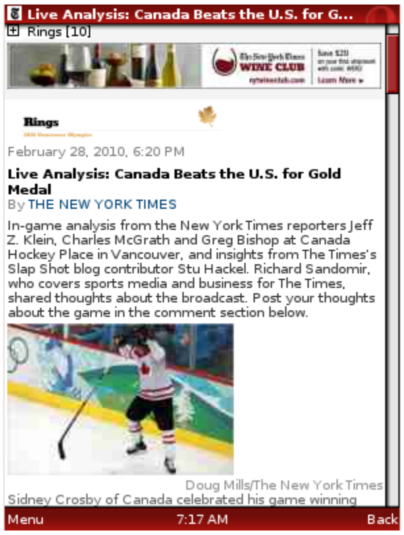
1.3.3 Mobile Web Browsing

With a small display and slow internet connection, users browsing the web with mobile phones do not want any additional images. Some websites offer a *lite* or a mobile version that excludes graphics, but many of them do not. Another solution can be done at the mobile web browser itself, as shown in Figure 1.3, Opera Mini displays a compressed version of a web page from Opera's dedicated server to reduce the bandwidth, but the advertisements are still loaded.

From the aforementioned reasons, an ability to filter out auxiliary images can be useful. This work presents a novel concept, and also an implemented demonstration, of a fast and robust method to filter out auxiliary web images. The concept considers the final rendering of web pages to partly mimic human perception of web images.

1.4 Related Work

There have been attempts to categorize web images. Cho et al. (2003) [6] proposed a set of 16 web image features, including the image file size which requires downloading the image. Downloading images again may take long time, and their method



Live Analysis: Canada Beats the U.S. for G...

Rings [10]

Rings

February 28, 2010, 6:20 PM

Live Analysis: Canada Beats the U.S. for Gold Medal

By THE NEW YORK TIMES

In-game analysis from the New York Times reporters Jeff Z. Klein, Charles McGrath and Greg Bishop at Canada Hockey Place in Vancouver, and insights from The Times's Slap Shot blog contributor Stu Hackel. Richard Sandomir, who covers sports media and business for The Times, shared thoughts about the broadcast. Post your thoughts about the game in the comment section below.

Doug Mills/The New York Times

Sidney Crosby of Canada celebrated his game winning

Fig. 1.3: A page of the New York Times from Opera Mini web browser; advertisements can be noticed at the top of the page.

considers only the HTML source code, not the rendered page. Hu et al. (2004) [7] categorized web images into functional uses. Their feature set requires frequency domain analysis, which also take time to process, and still, they consider only the HTML source code. Maekawa et al. (2006) [9] proposed a comprehensive set of 37 features that are mostly based on the composition of images but they did not take page rendering into account. Li et al. (2008) [8], on the other hand, use the features extracted from images in a rendered web page to find one dominant image. However, they used only three page-level features to classify images.

Our initial work [10] included image processing techniques such as face detection and optical character recognition as features. It yielded accuracy of 94.09% but the calculation time was much slower and the data set was relatively smaller. This work tries to reduce the computational cost by discarding some features while improve the accuracy with larger data set.

Chapter 2

Methodology

2.1 Features for Image Representation

By using information from a successfully-rendered web page, CSS and JavaScript are taken into the calculation of each image's properties. This means the values of the extracted features are close to what seen by eyes. To achieve a practical speed of feature extraction, all features are based on the DOM. A feature list is described with presumptions below.

1. **Position on X-coordinate:** Informative images tend to be located more on the left half of the page because of the majority of people reading left to right.
2. **Position on Y-coordinate:** Informative images tend to appear in the upper part of the page.
3. **Area (in square pixel):** Informative images tend to be larger.
4. **Width:** Informative images tend to be wider.
5. **Height:** Informative images tend to be higher.
6. **Width/Height Ratio:** Informative images tend to have this ratio close to one.
7. **Is Used as a Hyperlink or Not:** Auxiliary images tend to be used as hyperlink.
8. **Is Used as a Hyperlink to Another Website or Not:** Advertisement images are always linked to other websites.
9. **Is Used as a Hyperlink to a Local Page or Not:** Auxiliary images tend to be linked to other pages.
10. **Length of Alternate Text (or Image Caption):** Informative images tend to have longer caption.
11. **Length of Neighboring Text:** Informative images tend to appear with text.

12. **Order of Appearance Among Images:** Informative images tend to be arranged in the middle of the sequence.

13. **Number of Appearance:** Informative images tend to appear only once in a page.

14. **Number of Images with the Same Area:** Auxiliary images such as menu in a page tend to have the same size.

15. **Number of Images with the Same Width:** Auxiliary images such as menu in a page tend to have the same width.

16. **Number of Images with the Same Height:** Auxiliary images such as menu in a page tend to have the same height.

17. **Number of Neighboring Images with the Same Area:** Menu or advertisement images tend to be in a group of similar images.

18. **Number of Neighboring Images with the Same Width:** Menu or advertisement images tend to be in a group of similar width.

19. **Number of Neighboring Images with the Same Height:** Menu or advertisement images tend to be in a group of similar height.

20. **Distance from the Bottom of the Page:** Auxiliary images tend to appear near the bottom of the page.

2.2 Feature Extraction

Most of the features are extracted from properties in the DOM straightforwardly. However, there are relative features that need specific calculations. Neighboring elements are defined by children that share the same parent which is not a BODY or TD element in the DOM hierarchy. This shows a concept of being contained and related to each other for elements and text.

The numbers of images with the same area, width, or height are calculated by traversing into each IMG element. These numbers can tell the function of an image, e.g., image links in a vertical menu are usually the same in height.

2.3 Classification

2.3.1 Support Vector Machines (SVMs)

Support vector machines were chosen to be a classifying method for this work because of its accuracy, and there have been many resources available. SVM is a supervised machine learning method. By learning from a given set of labelled data, SVM can predict a category for unknown data according to the previously-learned model.

To simplify an explanation for SVM, a linearly separable case is first to be considered. For a set of data to be classified into two groups, there may be many possible solutions as shown in Figure 2.1. It is reasonable that an optimal solution should maximize the distance between the separating *hyperplane*, or a plane in a high dimensional space of data, and the decision boundaries, which are specified by datapoints called *supportvectors*.

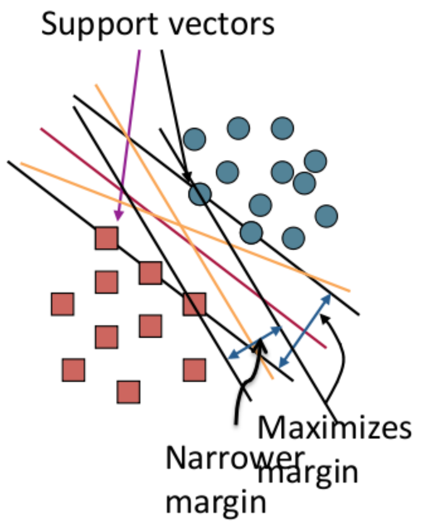


Fig. 2.1: Possible solutions for data classification. Source: Christopher Manning, *Slides for Introduction to Information Retrieval: Lecture 14* (Stanford University, 2009), 5.

The separating hyperplane is defined by a normal vector $\vec{w}$ and an offset b. All the points $\vec{x}$ on the hyperplane need to satisfy $\vec{w}^T\vec{x} = -b$. If the binary classification result is either +1 or -1, the linear classifier can be derived as:

$$f(\vec{x}) = sign(\vec{w}^T\vec{x} + b) \tag{2.1}$$

However, one can see that the size of $\vec{w}$ is not limited. Different size of $\vec{w}$ is still a normal vector of the hyperplane, but it affects the classifier $f(\vec{x})$. Therefore, a normalized vector $\vec{w}/|\vec{w}|$ is used instead. Now let r be a functional margin of $\vec{x}$, which is the distance from any data point $\vec{x}$ to the closest point $\vec{x}'$ on the hyperplane. We obtain:

$$\vec{x}' = \vec{x} - yr\frac{\vec{w}}{|\vec{w}|} \tag{2.2}$$

Using the fact that $\vec{x}'$ lies on the hyperplan and thus satisfies $\vec{w}^T\vec{x}' = -b$, we can derive r as:

$$r = y\frac{\vec{w}^T\vec{x} + b}{|\vec{w}|} \tag{2.3}$$

It is obvious that now $\vec{w}$ and b can be scaled and has no effect upon r. Then, $\vec{w}$ is used as a unit vector, or $|\vec{w}| = 1$. Consequently, for the sake of simplicity, r can be set as 1, and the following constraints follow:

$$\begin{aligned} \vec{w}^T\vec{x} + b &\geq 1 \qquad \text{for } y = 1 && (2.4) \\ \vec{w}^T\vec{x} + b &\leq -1 \qquad \text{for } y = -1 && (2.5) \end{aligned}$$

The geometric margin, or the gap between the two classes, becomes:

$$\rho = \frac{2}{|\vec{w}|} \tag{2.6}$$

Since the goal is to maximize this gap, or, in other words, to minimize $|\vec{w}|/2$, we want to find $\vec{w}$ and b such that:

- $\frac{1}{2}\vec{w}^T\vec{w}$ is minimized, and
- for all $\{(\vec{x}_i, y_i)\}, y_i(\vec{w}^T\vec{x}_i + b) \geq 1$

These can be considered as a quadratic function optimization problem, which is described as follows.

Find $\alpha_1, \ldots \alpha_N$ such that $\sum \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j \vec{x}_i^T \vec{x}_j$ is maximized, and

- $\sum_i \alpha_i y_i = 0$
- $\alpha_i \geq 0 \text{for all} 1 \leq i \leq N$

This problem has the following form of solutions.

- $\vec{w} = \sum \alpha_i y_i \vec{x}_i$
- $b = y_k - \vec{w}^T \vec{x}_k$ for any $\vec{x}_k$ such that $\alpha_k \neq 0$

The classifier is rewritten as follows, where $\vec{x}$ denotes the test data set.

$$f(\vec{x}) = sign(\sum_i \alpha_i y_i \vec{x}_i^T \vec{x} + b) \tag{2.7}$$

To tackle with a data set that is not linearly separable, for example in Figure 2.2, the data need to be mapped onto a higherdimensional space to be able to use a linear classifier. The method is usually called *"the kernel trick"*. The kernel function $K(\vec{x}_i, \vec{x}_j)$ can be considered as a measure of similarity of data points. Recalling the equation (2.7), the kernel function is only a dot product of two data points, or $K(\vec{x}_i, \vec{x}_j) = \vec{x}_i^T \vec{x}_j$.

A widely used kernel function which is also used in this work is a *radial basis function (RBF)*, which is calculated as $K(\vec{x}_i, \vec{x}_j) = e^{-(\vec{x}_i - \vec{x}_j)^2/(2\sigma^2)}$. However, the detail of the RBF kernel is not in the scope of the work and thus omitted.

2.3.2 LibSVM

The classification in this work is implemented by using LibSVM [4]. LibSVM is a software library for performing SVM classification. It provides executable files to be run externally in both Windows and Linux, and also interfaces for programming languages like C++, Java, MATLAB, Python, and many more. It allows users to adjust a broad range of parameters, mainly as follows:

- Type of algorithm
- Type of kernel function

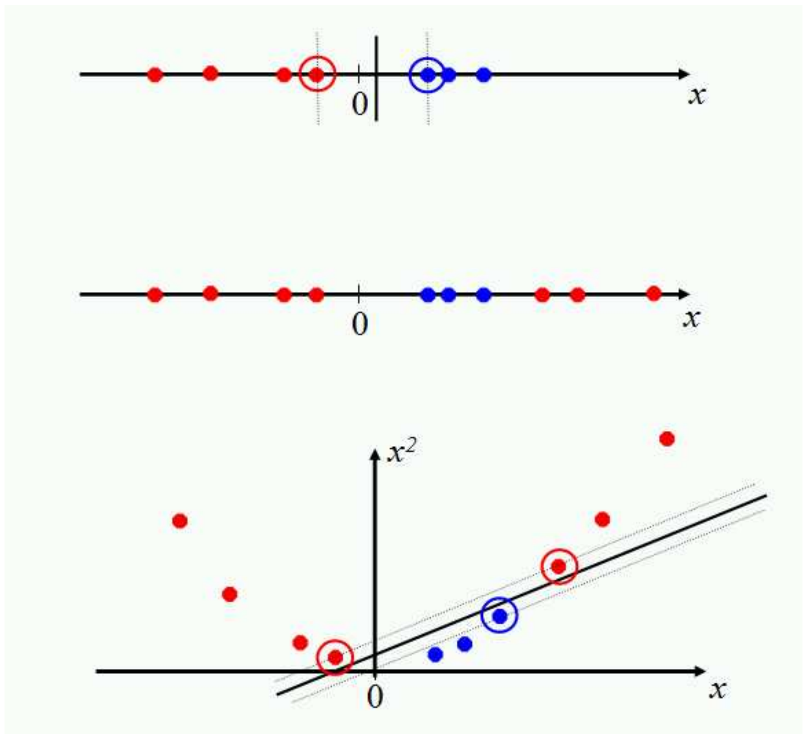


Fig. 2.2: Example of nonlinear SVMs. Source: Christopher Manning, *An Introduction to Information Retrieval* (Cambridge University Press, 2009), 331.

- degree of kernel function
- Gamma variable in kernel function
- Cost or *slack* variable in kernel function
- Weight of each class

Main executable files are svm-train, svm-scale, and svm-predict. The svm-train file is used for training an input training set file to a model file. The svm-scale file is used for scaling a training set file to an appropriate range, which is commonly [-1,1], and also for scaling a testing set file to have the same range as the model. The svm-predict is used for generating an output file from a model and a testing set file. The output is either 0 or 1.

2.3.3 Grid Search Technique

To obtain an optimal classification model, an optimal set of parameters is needed. For a common kernel function like the RBF kernel, only two parameters, C and $\gamma = \frac{1}{2\sigma^2}$, are used. C is the cost variable that allows some errors or creates a $softmargin$ to avoid overfitting. γ is a part of the aforementioned RBF kernel function, as the coefficient of the difference between trained data points and test data points.

Grid search technique is a brute force technique that tests possible pairs of parameters from a grid to find the best one. This technique is considered time consuming but easy to implement.

2.3.4 Data Collection and Preprocessing

In this work, the classification model is trained from 1,720 images from 77 web pages in various categories. Each image was labelled manually as either an informative image or an auxiliary one.

We used grid-search method to find an appropriate pair of the SVM kernel parameters. Also, with the imbalanced data set (1:15 for informative:auxiliary), we adjusted the weight of each class accordingly to improve sensitivity.

2.4 Implementation

The whole method presented in this paper can be implemented by two approaches, a server-based approach, and a client-based approach.

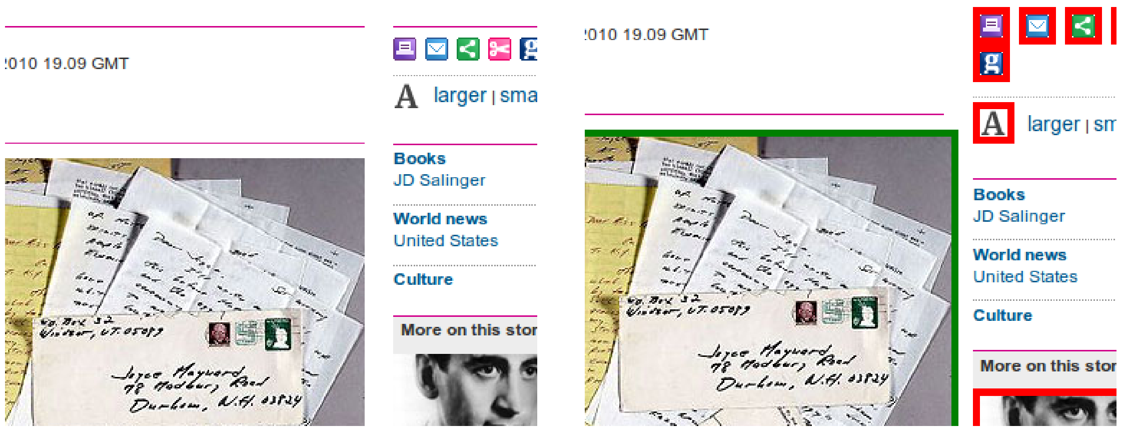
2.4.1 Server-Based Approach

In this approach, the process of classification is done on a server. A software with a rendering engine and an ability to program is needed to be installed on the server, then the classification model is added into it. A practical example is Jaxer [2], an AJAX server that uses an open source web browser Firefox's rendering engine. An input of the system is a web page URL, and an output can be a DOM tree of which images are classified and annotated. This approach is appropriate for all applications because all computation is done remotely.

2.4.2 Client-Based Approach

In this approach, the process of classification is done on a client software. The software has to contain, or to be, a web browser, with the classification model added. The rendered DOM is processed locally and the result is instantly shown to the user.

We have chosen this approach for a demonstration and a prove of concept. The implementation is done as an extension for Firefox. A classifying executable file and a trained model file are included in the extension.



(a) Before classification. (b) After classification.

Fig. 2.3: Demonstration of firefox extension

With direct access to the DOM, it classifies images and annotates them with borders of different colors as shown in Figure 2.3. It also lets users to override the result in case any error occurs. This approach is appropriate for web page printing.

2.5 Optimization

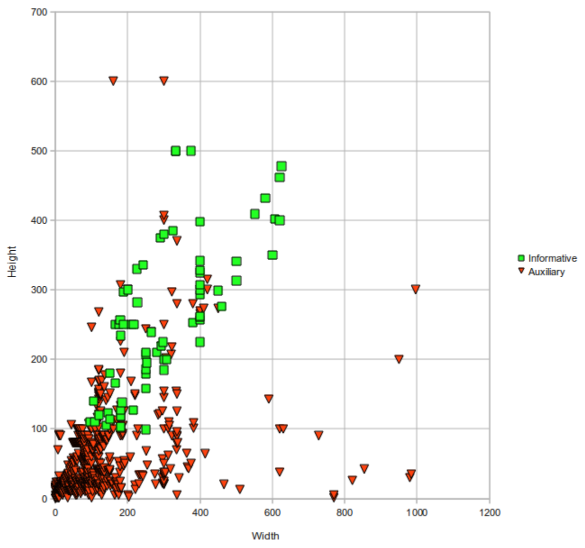


Fig. 2.4: Distribution of image dimensions.

In our implementation, calculation time for the whole web page depends on the number of images. With a simple rule to ignore images that are obviously auxiliary, the calculation time can be improved.

According to a plot of image dimensions in Figure 2.4, it is obvious that there is no informative image that has the dimension of smaller than 200 x 100 square pixel. Therefore, we can set a filter according to this information.

Chapter 3

Results and Discussion

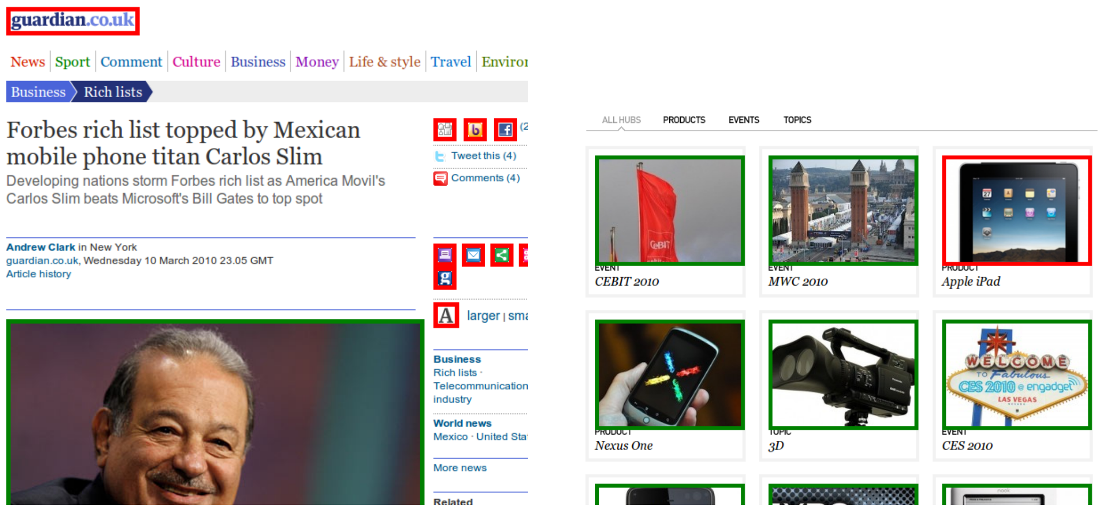
Classification model evaluation was done by 10-fold cross-validation method. We split the labelled data set into two parts. The first part of randomly-selected 80% was used to train the model, and the rest was used to test it. The test was repeated for 10 times, each time with a new selection of data.

Table 3.1: Confusion Matrix of Average Prediction Results

	Informative	Auxiliary
Informative (Predicted)	20.2	10
Auxiliary (Predicted)	2.3	328.4

To evaluate the results as in the table 3.1, four measurements are defined. A ratio of the correct prediction in total prediction is defined as an accuracy, which in this case is 96.59%. The other two quantities, a specificity and a sensitivity, are defined to measure the real performance of the classification and thus take into granted the effect of unbalanced results, that is; a high accuracy can be achieved by predicting all images as auxiliary, but that would have no use. The high specificity of 97.04%, or the ratio between the results correctly predicted as auxiliary and real auxiliary data, assures that most of the auxiliary images are removed. On the other hand, the high sensitivity of 89.78%, or the ratio between the number of results classified as informative and that of real informative data, tells that only a few of informative images are missed.

The last measurement is a precision. It is defined by a proportion of the results correctly predicted as informative in all results predicted as informative. This classification has a precision of 66.89%, which seems to be low but that is actually acceptable. With the low number of informative images in the data set, it is rational to allows a not so high precision while maintaining a high specificity, because we do not want to risk many informative images to be predicted as auxiliary images, while those that are obviously auxiliary are well-classified.



guardian.co.uk

News | Sport | Comment | Culture | Business | Money | Life & style | Travel | Enviro

Business › Rich lists

Forbes rich list topped by Mexican mobile phone titan Carlos Slim

Developing nations storm Forbes rich list as America Movil's Carlos Slim beats Microsoft's Bill Gates to top spot

Andrew Clark in New York
guardian.co.uk, Wednesday 10 March 2010 23.05 GMT
Article history

Tweet this (4)
Comments (4)
larger | sma

Business
Rich lists ·
Telecommunication industry
World news
Mexico · United Sta
More news
Related

(a) Before classification. (b) After classification.

Fig. 3.1: Classification results from an article page and a tabular page

The current classification model works well with, and was trained from, article-formatted web pages like one in Figure 3.1, because they are printed the most among other styles. If there is more training, the model will be able to perform better with other kinds of web pages, for example, a tabular page as in Figure 3.1, but it may lose its performance with the article pages as it becomes more generalized.

The classification results have proved that a larger data set makes a better classification model. An increase from about 600 images to 1700 images resulted in 2% more accuracy. An improved feature set has reduced the computational time and shows that complex features are probably not necessary.

In addition, a rule based classification based on image dimensions dramatically reduced the avarage computational time from 6 to 1 second on a PC with Ubuntu Linux OS, 2.53GHz CPU and 4GB RAM.

The problem of imbalanced data set that caused fault negative results was basically resolved by adjusting weigths for the two classes. However, there are still other techniques such as SMOTE [5] available to be applied and could be done in the future.

The current implementation of the proposed method as a Firefox add-on still does not cover all the applications as described in Chapter 1. The server-based approach would be more versatile, as it possesses the functions of the client-based approach and still be able to serve the classification results to other applications. However, this approach is much more sophisticated because it requires a modification of a web browser.

Chapter 4

Conclusion

A classification of web images to keep only the informative ones can help three applications, web page printing, search engine indexing, and mobile web browsing, to be more efficient. A set of Document Object Model properties were selected as features for the classification. We manually labelled 1,720 web images from 77 web pages to train a classification model with Support Vector Machines. Appropriate parameters for SVMs were then chosen by a grid search technique and the weight of each class is adjusted by trial-and-error. A rule-based classifier was added to reduce the workload of the SVM classifier and thus reduce the computation time. These cause an accuracy to be so far achieved as 96.59%, with a specificity of 97.04%, a sensitivity of 89.78%, and a precision of 66.89%.

An advantage of the proposed method is the use of information rendered completely by a web browser, which not many previous works have taken into account. It makes the extracted data to be as accurate as what we see by eyes, which reading the HTML source code alone cannot achieve.

The current classification model was trained for web pages that contain articles. To use it with other styles of web page, it needs to be trained with more data set. However, to be more generalized, the model may lose its accuracy in specific domains that may be more important. To create a separate classification models for different styles of web pages can solve this problem.

The implementation to demonstrate the proposed method was done as an extension for Firefox web browser as a client-based approach. Users can download this extension, which is open source, and used it before printing web pages. This extension is mainly written in JavaScript, which is ready to be applied to a server-based approach with slight modification.

The future work is focused on the implementation of the server-based method. In additional, a larger training data set is desirable.

References

[1] Hp smart web printing. http://www.hp.com/global/us/en/consumer/digital_photography/free/software/smart-web-printing.html.

[2] Jaxer: The world's first ajax server. http://jaxer.org.

[3] Printfriendly: Save money & the environment. http://printfriendly.com.

[4] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[5] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[6] Soosun Cho and Chi-Jung Hwang. Image features for machine learning based web image classification. In Simone Santini and Raimondo Schettini, editors, *Internet Imaging IV*, volume 5018, pages 328–335. SPIE, 2003.

[7] Jianying Hu and Amit Bagga. Categorizing images in web documents. *IEEE MultiMedia*, 11(1):22–30, 2004.

[8] Zhiwei Li, Shuming Shi, and Lei Zhang. Improving relevance judgment of web search results with image excerpts. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 21–30, New York, NY, USA, 2008. ACM.

[9] Takuya Maekawa, Takahiro Hara, and Shojiro Nishio. Image classification for mobile web browsing. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 43–52, New York, NY, USA, 2006. ACM.

[10] Tewson Seeoun, Choochart Haruechaiyasak, and Toshiaki Kondo. Identifying auxiliary web images using combination of analyses. In *MM '09: Proceedings of the seventeen ACM international conference on Multimedia*, pages 1125–1126, New York, NY, USA, 2009. ACM.

Appendix A

Software Download

The implemented software can be downloaded from `http://tewson.com/auximg`. It has been tested with Firefox 3 on both Windows XP and Ubuntu Linux.

Appendix B

Publications

In the early stage of this work, a paper, *Identifying Auxiliary Web Images Using Combination of Analyses*[10], was submitted to the Multimedia Grand Challenge track at ACM Multimedia 2009 in Beijing, China, and received an Honorable Mention Prize from Hewlett-Packard.