This paper proposes an automated web site evaluation using machine learning to extract evaluation criteria from existing evaluation data. Web site evaluation is a significant task because evaluated web sites provide information useful to users in estimating sites validation and popularity. Although many practical approaches have been taken to present possible measuring sticks for web sites, their evaluation criteria are manually determined. We developed a method to obtain evaluation criteria automatically and rank web sites with the learned classifier. Evaluation criteria are discriminant functions learned from a set of ranking information and evaluation features collected automatically by web robots. Experiments confirmed the effectiveness of our approach and its potential in high-quality web site evaluation.

Keywords: evaluation criteria extraction, web site evaluation

1. Introduction

Internet users number over 1.7 billion [1] and the number of web sites has grown exponentially. A November 2009 survey estimated 233 million [2] of these rich, highly updated information sources. Publishing on the Internet is easy, cheap, and fast. However, the web’s very nature means that not all information is adequately or clearly attributed, or complete and trustworthy.

This had increased the use of web site evaluations in fields such as business, universities, government, and E-commerce. Based on Jupiter Research Reports, budget earmarked for web site evaluation products and services exceeded 1 billion US dollars annually by 2006.

No unified criteria currently exist that are fully trustworthy for web site evaluations. Each user has different criteria, and it is almost impossible to get a positive result using the same evaluation criteria on web sites of different categories. Moreover, the evaluation changes over time even for the same web site. Although many tools for evaluating the usability, accessibility, and security of web sites [3–6] have started to appear, the evaluation tasks are basically still done manually [7]. There are limitations as to how many web sites can be manually evaluated, and an automated evaluation system has been seen as a necessary step.

In this paper, we propose an automated method to extract evaluation criteria. This involves focusing on categorized web sites and setting evaluation criteria for individual categories applying machine learning to deal with ranking. We then use web robots to collect the features of the evaluation objectives and evaluate them automatically based on their features and the criterion of the corresponding category.

Establishing evaluation criteria requires that evaluation data be obtained regularly and automatically from the general public – a task conventionally done by questionnaires. This has changed however, with the advent of registration ranking sites – real-time web site ranking services based on user recommendations and votes. The most popular of these, such as SiteRank1, have hundreds of thousands of web sites registered and users make categories, register new sites, vote, and write comments frequently. In this research, we use categorized ranking information from registration ranking sites to set up web sites evaluation criteria. In other words, we treat automated web site evaluation as a classification learning problem in which discriminant functions are learned by categorized ranking information used as training data. We compared three major ranking algorithm – Ranking SVM [8], Prank [9], and Multiclass SVM [10] – to find the one best adapted to solving the ranking problem.

2. Background

Web site evaluation task requires appropriate evaluation measures. Researches such as Web WISDOM [11] and WebMac [12] use indexes from library and information science findings as meta-information to create evaluation measures, although these do not particularly target automation.

Aside from the approach of the questionnaire sur-

very, Velayathan and Yamada [13] proposed automatically learning evaluation criteria based on user browsing behavior, but evaluation objectives were web pages, not web sites, and users had to browse evaluated objectives to ascertain certain browsing behavior.

Tools focusing on evaluating web site usability, accessibility, and security are roughly divided into (1) server performance analysis, (2) usage analysis based on log data, (3) guideline compliance checking, (4) navigation text analysis, (5) cyber agent navigation simulation [14]. Automatically evaluating web sites by these tools, however, involves evaluation criteria predetermined by experts.

3. Evaluation Criteria Extraction

We define evaluation criteria as discriminant functions learned from a set of ranking information. In this section, we introduce algorithms that solve the ranking problem and detail evaluation features and procedures.

3.1. Ranking Algorithms

Our ranking problem uses ranking information as both input and output. Algorithms such as Ranking SVM, Prank, and Multiclass SVM used to solve this kind of problem are detailed below.

3.1.1. Ranking SVM

As presented graphically in Fig. 1, the distance from hyperplane $w$ of datapoint $x$ is mapped onto one-dimensional (1D) space in which $\theta_1, \ldots, \theta_{K-1}$ are different thresholds against which distance is compared. Data point $x$ having rank $i$ satisfies $\theta_{i-1} < w^T x_j < \theta_i$. Checking which section $w^T x_j$ goes into helps determine the rank of $x$. Define $\alpha_i = \theta_{i+1} - \theta_i$; $1 \leq i \leq K-1$. No lower bound exists for data items belonging to rank 1 and all data items belonging to rank $K$ have no upper bound. Note that data point $x_j$ having rank $i > 1$ satisfies the following:

$$w^T x_j > \theta_{i-1}; w^T x_j + \alpha_i > \theta_i; w^T x_j + \sum_{k=i}^{K-2} \alpha_k > \theta_{K-1}.$$ 

Similarly, for an example with rank $i < K$, we have the following:

$$w^T x_j < \theta_{i-1}; w^T x_j + \alpha_i < \theta_i; w^T x_j + \sum_{k=i}^{K-2} \alpha_k < \theta_{K-1}.$$ 

Based on these observations, define $\bar{w} = [w, \alpha_1, \alpha_2, \ldots, \alpha_{K-2}]$ and for example $x_j$ with rank $1 < i < K$, define $\bar{x}_j^+$ and $\bar{x}_j^-$ as $n + K - 2$ dimensional vectors as follows:

$$\bar{x}_j^+ [l] = \begin{cases} x_j[l] & 1 \leq l \leq n \\ 0 & n < l < n + i - 1 \\ 1 & n + i - 1 \leq l \leq n + K - 2 \end{cases}$$

$$\bar{x}_j^- [l] = \begin{cases} x_j[l] & 1 \leq l \leq n \\ 0 & n < l < n + i \\ 1 & n + i \leq l \leq n + K - 2 \end{cases}.$$ 

This leads to the following:

$$w^T \bar{x}_j^+ > 0 \Rightarrow w^T x_j > \theta_{i-1}; w^T \bar{x}_j^- < 0 \Rightarrow w^T x_j \leq \theta_i.$$ 

Once $\bar{x}_j^+$ and $\bar{x}_j^-$ are defined, the ranking problem is simply reduced to learning classifier $\bar{w}$ in $n + K - 2$ dimensional space so that $w^T \bar{x}_j^+ > 0$ and $w^T \bar{x}_j^- < 0$. This standard classification problem has at most $2m$ training examples, half of which have label $+1$ (examples $\bar{x}_j^+$) and the other half label $-1$ (examples $\bar{x}_j^-$).

Many real-world problems may not yield a linear function powerful enough to learn data ranking, so nonlinear mapping $\phi()$ is used over the original feature vector. If $x_j$ has rank $i$, then Eq. (4) is satisfied. However:

$$\kappa(\bar{w}, \bar{x}_j^+) = \phi(\bar{w})^T \phi(\bar{x}_j^+) > 0 \Rightarrow \phi(\bar{w})^T \phi(\bar{x}_j) > \theta_{i-1}.$$ 

is not satisfied due to the nonlinearity of $\phi()$. To get around this problem, a new kernel function has to be defined. For kernel function $K$ and corresponding mapping $\phi$, define new kernel function $\kappa$ with the corresponding mapping $\phi()$ as follows:

$$\phi(\bar{w}) = [\phi(w), \bar{w} [n+1 : n+K-2]]$$

$$\phi(\bar{x}) = [\phi(x), \bar{x} [n+1 : n+K-2]].$$ 

A kernel trick becomes available with the new kernel function:

$$K(\bar{w}, \bar{x}) = \kappa(w, x).$$ 

$$\bar{x} [n+1 : n+K-2].$$ 

3.1.2. Prank

Prank, a perceptron-based online learning algorithm, trains a perceptron model, retaining both weight vector $w$ and threshold vector $b$, with the objective of finding a perceptron model that projects all instances into $k$ subintervals defined by $b$. Fig. 2 shows the Prank algorithm.

3.1.3. Multiclass SVM

Multiclass SVM, a generalization of two-class SVM systems, deals with multiclass problems mainly by breaking down single multiclass problems into multiple binary problems, each yielding a binary classifier assumed to produce an output function giving relatively large values
Initialize: Set $w^1 = 0, \ldots, w^k = 0, b^1 = \infty$
Loop: For $t = 1, 2, \ldots, T$
- Get a new rank-value $x \in \mathbb{R}^r$
- Predict $y_t = \text{nn}(x, j)$: $w^j \cdot x - b^j < 0$, $j = 1, \ldots, k$
- Get a new rank $y^*$
  - If $y^* \neq y_t$ update $w$ (otherwise set $w^j = w^j, j \neq l$)
  1. For $r = 1, \ldots, k - 1$: If $y^* \leq r$ Then $y_t' = r$
  2. For $r = 1, \ldots, k - 1$: If $(w^j \cdot x - b^j)_{y^*} \leq 0$ Then $y_t' = y_t$
  Else $y_t' = 0$
2. For $r = 1, \ldots, k - 1$: If $y_t \neq y_t'$ update $b^j$
  1. $b^j = b^j + \frac{1}{t} (y_t' - y_t)$
3. Update $w^j = w^j + \frac{1}{t} \sum_{i=1}^{T} x_i$
Output: $H(x) = \text{arg min}_{y \in \mathbb{R}^l} w^j \cdot x - b^j$ (8) $\forall j$

Fig. 2. Prank algorithm.

to examples from the positive class and relatively small values to examples belonging to the negative class. To deal with k-class classification, for example, it generates set of 2-class classification SVMs $f^1, \ldots, f^k$ that discriminate between one class and the rest, then classifies multi-classes based on Eq. (8) to integrate $f^1, \ldots, f^k$.

$$g^j(x) = \sum_{i=1}^{l} y_i \alpha_i^j \cdot k(x, x_i) + b^j$$ \hspace{1cm} (9)

The final complex function is as follows:

$$f^j(x) = \text{sgn}(g^j(x))$$ \hspace{1cm} (10)

### 3.2. Evaluation Features

We use ranking information from registration ranking sites as training data. Each data consists of multiple evaluation features and their values, and the data value is the rank determined by ranking sites. Using discriminant functions learned from training data, we rank unranked web sites based on their features. Data from these unranked web sites is test data.

Sinha et al. [15] and Ivory et al. [16] studied “best” web sites selected by experts and users to identify common features. These include content, structure and navigation, visual design, functionality, and interactivity. Palmer [17] concluded from three consecutive tests that web site success depends on speed, navigation, content, interactivity, and response. These researches deal with both subjective and objective user features. Our goal, however, is to automate evaluation, so the features we need must be objective and obtained automatically from web servers. Table 1 lists 31 features categorized in 10 fields currently used in our research and detailed below.

(i) **Global Link Popularity** shows a web page’s popularity by examining the link structure of the web. We use Google PageRank here.

(ii) **Freshness** is the contrast of new and old information.

We use update frequency and the amount of text information changed as a guide.

(iii) **Indexable Text Information** is the amount of visible HTML text information obtained by calculating the size of HTML source files with tag information removed.

(iv) **Multi-media Content** means the number of files including images, videos, audios and flashs.

(v) **Spelling and Grammar Accuracy** counts the number of misspellings.

(vi) **HTML Document Accuracy** checks features such as whether character code and image size are specified.

(vii) **Content Security** checks features related to the control of cache, scripts, and web robots.

(viii) **Content Constitution** uses the proportion of multimedia content to text information as a guide.

(ix) **Design** considers information on BGM, background image, frames, style sheets, etc.

(x) **Others** includes the number of links on the top page, the number of links overall, whether the page author is specified, and whether the page description is specified.

<table>
<thead>
<tr>
<th>Field</th>
<th>The number of features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top page’s global link popularity</td>
<td>1</td>
</tr>
<tr>
<td>Freshness</td>
<td>2</td>
</tr>
<tr>
<td>Indexable text information</td>
<td>1</td>
</tr>
<tr>
<td>Multi-media content</td>
<td>4</td>
</tr>
<tr>
<td>Accuracy of spelling and grammar</td>
<td>1</td>
</tr>
<tr>
<td>Accuracy of HTML documents</td>
<td>4</td>
</tr>
<tr>
<td>Content security</td>
<td>5</td>
</tr>
<tr>
<td>Content constitution</td>
<td>4</td>
</tr>
<tr>
<td>Design</td>
<td>5</td>
</tr>
<tr>
<td>Others</td>
<td>4</td>
</tr>
</tbody>
</table>

These 31 features were selected under the restriction of currently available web server technology. Although other features will become available as web technology advances, we believe these features are appropriate for our purpose.

### 3.3. Automated Web Site Evaluation

As shown in Fig. 3 overview of our web site evaluation system, in the learning phase, categorized ranking information from registration ranking sites and evaluation features collected by web robots are input to the ranking learning machine to learn categorized discriminant functions. In the test phase, ranks of unranked web sites are predicted automatically based on site’s features and corresponding discriminant functions.
4. Experiments

We compared ranking algorithms Ranking SVM, Prank, and Multiclass SVM experimentally to find the one best adapted to this research. We then verified the adequacy of extracted evaluation criteria and studied the correlation between evaluation features and classification results and compared our evaluation to a web site importance estimation approach.

The dataset consisted of 735 web sites in seven categories in which each category had web sites divided into five equal groups and had labels assigned ranging from 1, Excellent, to 5, Not expected. The 31 evaluation features discussed in Section 3.2 were selected. LibSVM2.84 was used as the SVM tool.

4.1. Evaluation Measures

We used Mean Absolute Error (MAE) – a statistical quantity used to measure how close forecasts or predictions are to eventual outcomes – to evaluate performance. MAE is given by:

\[ MAE = \frac{1}{N} \sum_{i=1}^{N} |p_i - q_i| \]  

where \( q_i \) is prediction, \( p_i \) the true value, and \( N \) the number of items – the smaller the MAE, the higher the classification accuracy.

4.2. Data Preprocessing and Kernel Selection

Values of the evaluation features vary from 0 to millions, so we scaled data pre-experimentally to [0,1]. LibSVM provides four kernels – linear, polynomial, radial basis function, and sigmoid. Experimentally studying kernel accuracy, we decided to use the RBF kernel due to its superior accuracy and reprogrammed part of the kernel source code to apply the kernel trick for Ranking SVM.

We did a grid search using cross-validation to find the best value of RBF kernel parameters – \( c \) and \( \gamma \) – for each of the seven web site categories.

4.3. Algorithm Comparison

We experimentally compared Ranking SVM, Prank, and Multiclass SVM, getting the results shown in Table 2. To get detailed information in addition to MAE, we evaluated the three algorithms by calculating the minimum number of adjacent transpositions [18] needed to bring the predicted value to ground truth. As Table 2 shows, Ranking SVM significantly outperformed Prank and Multiclass SVM, implying that the ranking problem in this research is nonlinear.

<table>
<thead>
<tr>
<th>Category</th>
<th>MAE</th>
<th>Minimum number of adjacent transpositions</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.78</td>
<td>132.7</td>
</tr>
<tr>
<td>B</td>
<td>0.93</td>
<td>217.4</td>
</tr>
<tr>
<td>C</td>
<td>1.12</td>
<td>264.5</td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.4. Criteria Extraction Accuracy Evaluation

In an experiment to verify the adequacy of extracted evaluation criteria, we conducted 10-fold cross-validation in each category, as shown in Fig. 4, to minimize the influence of biased data. Average MAE is 0.78 and standard deviation is 0.3, meaning that, in most cases, the predicted value is either equal to the true value or shifted one rank from it. We looked at studies [19–21] that also use MAE with a 5-level evaluation and found that keeping MAE under 1 is sufficiently effective.

4.5. Feature Contribution Ratio Experiment

It is important for users and web site authors to know the correlation between evaluation features and classification results. Since the ranking problem in this research is nonlinear and we use a kernel trick in Ranking SVM, we cannot obtain the feature contribution ratio directly from the learned discriminant functions, so we used a wrapper [22], widely used in feature selection, to determine the feature contribution ratio. Table 3 shows the top five...
Table 3. Top five features by category.

<table>
<thead>
<tr>
<th>Category</th>
<th>Feature (Computer science)</th>
<th>Feature (Music)</th>
<th>Feature (Health and diet)</th>
<th>Feature (Movies)</th>
<th>Feature (Sports)</th>
<th>Feature (Games)</th>
<th>Feature (Fashion)</th>
</tr>
</thead>
<tbody>
<tr>
<td>computer science</td>
<td>Indexable text information</td>
<td>Percentage of updated text information</td>
<td>Number of links (site)</td>
<td>Specification of author information</td>
<td>Top page’s PageRank</td>
<td>Number of links (top page)</td>
<td>Number of audio files</td>
</tr>
<tr>
<td>music</td>
<td>Number of links (top page)</td>
<td>Number of audio files</td>
<td>Number of links (site)</td>
<td>Top page’s PageRank</td>
<td>Percentage of audio files</td>
<td>Number of links (site)</td>
<td>Specification of image size</td>
</tr>
<tr>
<td>health and diet</td>
<td>Indexable text information</td>
<td>Number of links (top page)</td>
<td>Number of image files</td>
<td>Specification of background image</td>
<td>Top page’s PageRank</td>
<td>Number of links (site)</td>
<td>Specification of image size</td>
</tr>
<tr>
<td>movies</td>
<td>Number of image files</td>
<td>Number of links (top page)</td>
<td>Indexable text information</td>
<td>Specification of background image</td>
<td>Top page’s PageRank</td>
<td>Number of image files</td>
<td>Specification of image size</td>
</tr>
<tr>
<td>games</td>
<td>Indexable text information</td>
<td>Percentage of updated text information</td>
<td>Number of image files</td>
<td>Specification of cache expiration date</td>
<td>Specification of font code</td>
<td>Number of links (top page)</td>
<td>Specification of image size</td>
</tr>
<tr>
<td>fashion</td>
<td>Indexable text information</td>
<td>Percentage of updated text information</td>
<td>Number of image files</td>
<td>Specification of cache expiration date</td>
<td>Specification of font code</td>
<td>Number of links (top page)</td>
<td>Specification of image size</td>
</tr>
</tbody>
</table>

5. Discussion

5.1. Category Influence

As shown in Fig. 4, differences in evaluation accuracy between categories are great. Categories with low accuracy exist for two reasons:

1. It is difficult to set unified evaluation criteria for some categories due to their characteristics, e.g., users visiting health and diet web sites may have very different objectives – looking for health advice, food, recipes, or even body-building equipments – from users visiting movie and music category web sites, meaning that different purposes lead to different evaluation criteria.

2. Important features that are the key to determining evaluation criteria are not acquired experimentally, e.g., if beautiful high-resolution images are preferred in the fashion category, it is difficult to set a high accuracy criterion without a feature describing it. More evaluation features are likely to become available with continuing improvement of web robots and web technology. With more adapted features, things improve for the low-accuracy categories.

5.2. Training Data Quantity and Quality

Our data came from over 700 web sites. To get enough for each category, we collected data from multiple registration ranking sites, although this may potentially decrease evaluation accuracy due to changes in user group and the web site quality bias. With further development of registration ranking sites, enough data will become available from a specific site, at which point evaluation accuracy is expected to improve. With registration ranking site categories becoming more specific, we expect evaluation accuracy to increase as categories become more specialized.

5.3. Evaluation Features and Contribution Ratio

Table 3 shows that features such as indexable text information and numbers of links play an important role in most categories. However, the fact that some features are effective only for corresponding categories shows that features reflect category characteristics. Features in Table 3, for example, contribute much to criteria extraction accuracy, although some features hardly have any effect or even an opposite effect – in which case it may be advantageous to remove such features at this time. Because
6. Conclusions

Improving the quality of web content is imperative for further web development, because good measuring sticks are needed for web sites. Many practical approaches have been made in this regard, but have required manual intervention, preventing dealing with the entire web. The automated web site evaluation we have proposed involves defining web site evaluation as web site ranking and applying machine learning to extract evaluation criteria. We compared three ranking algorithms, finding Ranking SVM to be the best adapted to solving the ranking problem. Our evaluation experiments confirmed the feasibility of extracted evaluation criteria. We also compared our evaluation to an existing web site importance estimation approach and verified that our evaluation has the advantage of evaluation accuracy. We also conducted feature contribution experiments to determine the correlation between evaluation features and classification results.

We have presented a practical approach to automatically evaluating web sites that has high potential in high-quality web site evaluation.

References: