

Barriers in (Vector) Space: A Clustering Approach to Web Accessibility Evaluation

Annika Nietzio
Forschungsinstitut Technologie-Behindertenhilfe (FTB)
der Evangelischen Stiftung Volmarstein, Grundschötteler Str. 40
58300 Wetter (Ruhr), Germany
eiao@ftb-net.de

Abstract

The analysis of web accessibility data generated by automatic evaluation can have many different objectives. This paper explores the application of unsupervised machine learning algorithms to identify underlying structures of the data. It also introduces the theoretical background for clustering experiments. The experimental results give interesting new insights into the data. In the future this might lead to optimised assessment methods and analysis approaches.

1 Introduction

The goal of the EIAO project¹ is to get an overview of the accessibility situation on a large part of the web (European web sites). To conduct such a large scale evaluation efficiently automatic tools are needed. The main characteristic of automatic web accessibility assessment is that it can only be used to check some features of web pages but it can check them exhaustively, i.e. every occurrence of a given feature is identified. The resulting data has to be preprocessed in order to enable users or policy makers to draw meaningful conclusions.

King et al. [4] report their findings from profiling the accessibility of another (smaller and more homogeneous) part of the web: the web sites of a big company. To reduce the number of pages that have to be (manually) tested they perform a clustering of the resources. Only a few pages from each cluster have then to be checked manually. This approach is based on the assumption that pages with the same barrier profile concerning automatically testable barriers exhibit also a similar distribution of barriers that are identified by expert checking.

A possible explanation for this is that web developers with similar backgrounds, using similar tools, also introduce similar accessibility problems in their pages (which can be either automatically detectable or not). Even though this assumption may be more workable for pages within one web site, we suspect that it holds to some extent also in the general case.

Therefore we propose to apply clustering methods to the EIAO data. Apart from providing new insights, this also allows the development of a new aggregation approach for automatic test results.

The next section gives a short overview of unsupervised machine learning and clustering algorithms. In section 3 we introduce vector space models and show how they can be applied to represent the barrier profile of a web page. Subsequently, we explore which insights about the data can be gained from this approach and report the findings from some preliminary experiments. We conclude with a discussion of possible extensions and mention some directions for further research.

¹The EIAO project is co-funded by the European Commission, under the IST contract 2003-004526-STREP.

2 Unsupervised Machine Learning

The objective of a machine learning algorithm is to identify the underlying structures and patterns in a set of data objects that cannot be classified by deterministic functions. The algorithms rely on the features of the data and take into account statistical properties. If the data is labelled into categories, supervised machine learning can be performed. If no labels are available, unsupervised learning – often also called *clustering* – has to be applied.

2.1 Clustering Algorithms

Cluster analysis, also called data segmentation, has a variety of goals. All relate to grouping or segmenting a collection of objects into subsets or “clusters”, such that those within each cluster are more closely related to one another than objects assigned to different clusters. For a more detailed introduction to cluster analysis see chapter 6 of [3].

For the experiments in this paper we will apply some basic clustering algorithms that result in flat cluster models, i.e. all the clusters have the same level, there is no hierarchy involved. Furthermore, we assume that all data is represented in a vector space.

Initialisation: K fixed. Choose K clusters randomly.
Step 1: Calculate the means of the k clusters \bar{x}_k for $k = 1, \dots, K$
Step 2: Assign each observation to the closest cluster mean

$$C(i) = \arg \min_k d(x_i, \bar{x}_k)$$

Repeat step 1 and step 2 until assignments do not change.

Figure 1: The K-Means algorithm

Figure 1 gives an overview of a simple iterative descent algorithm: the *K-Means* algorithm, where K is the number of clusters that are to be detected. The input data are denoted by x_1, \dots, x_n . In the iteration step each object x_i is assigned to the cluster $C(i)$ with the closest mean \bar{x}_k . An important part of the algorithm is the distance function $d(\cdot, \cdot)$ that serves as a measure of how similar or dissimilar two data objects are.

Another widely used clustering algorithm is the *expectation maximisation* (EM) algorithm. It is an algorithm for finding maximum likelihood estimates of parameters in probabilistic models, where the model depends on unobserved latent variables.

2.2 Similarity Measures

The selection of the similarity measure is crucial for the success of the clustering method. The *squared Euclidean distance*

$$d(x_i, x_j) = \|x_i - x_j\|^2$$

can be used if the data vectors have comparable length. Another widely used measure is the *cosine similarity measure*

$$\cos(x_i, x_j) = \frac{\langle x_i, x_j \rangle}{\|x_i\| \cdot \|x_j\|}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product. The cosine is related to the angle between the two vectors. Vectors that point into the same direction have a high cosine value (close to 1).²

²It would be useful to consider also other similarity measures, e.g. measures that take into account the length of vector, which is related to the number of barriers.

The K-means algorithm is appropriate when the similarity can be measured via the squared Euclidean distance. The *K-Medoids* algorithm is a generalisation of the K-Means algorithm that allows the usage of arbitrary similarity measures.

3 Barrier Vectors

The clustering algorithms introduced above operate on vector spaces. This section explains how a web page and its accessibility situation can be represented in a vector space.

3.1 Vector Space Model

Data that is made up of attribute-value-pairs (where the value is numerical) can be represented in a vector space. The set of attributes defines the dimensions of the vector space. Ideally, the attributes should be independent features of the data.

This model was first proposed by Salton et al. [7] who applied it to document classification. Each text document is represented by a vector containing term frequencies. The dimensions of the model correspond to (indexing) terms. By introducing a similarity measure the vectors can be compared, e.g. documents with similar topics can be identified.

3.2 Barrier Vector Space

In the web accessibility evaluation context the dimensions of the vector space are the different barrier types that are assessed. The values are the number of failed tests.

Example: In the experiments we are using the automatically generated accessibility data from the EIAO observatory. The EIAO web accessibility metric module (WAM module) checks 74 single barrier types.³ So the barrier vector space has 74 dimensions. The *barrier vector* for a web page p might for instance look like this:

$$\vec{v}_p = (0, 1, 0, 0, 3, 2, \dots, 0, 1)$$

On page p there was no barrier of type 1, 3, or 4 detected. The test for barrier type 2 yielded one fail result, barrier type 5 has three fail results, etc.

In this way each of the web pages that have been inspected is represented as a single vector in the barrier vector space.

3.3 Interpretation of Results

Web pages that are located in one cluster have similar barrier profiles. The clusters need to be investigated in order to find out what information they bear. There are several potential categorisation schemes. On the one hand it is possible to identify clusters that consist of web pages with similar accessibility situations. This leads to a cluster model of web accessibility which distinguishes several levels of severity. Section 5.1 outlines a method that assigns severity labels to barrier vectors based on a cluster model.

On the other hand the cluster analysis can identify the characteristic clusters for each web site. It highlights the areas of a web site that have the biggest accessibility problems. The average cluster vector, which represents the cluster (i.e. is some kind of aggregated measurement), contains information about the type of barriers. The cluster model can also be used to select a subsample of pages for manual evaluation as described in [4].

³The tests are derived from WCAG 1.0 [5]. Only requirements that can be tested fully automatically are included.

4 Barrier Vector Clustering

This paper reports the findings of two initial clustering experiments. The data in both of the experiments is the automatic assessment of five European prime minister web sites performed by EIAO in May 2006. One hundred pages from each site were retrieved and tested, resulting in more than 280 000 single test assertions (i.e. pass or fail results).

4.1 Setup of the Learning Environment

To be able to study several algorithmic approaches and parameter settings easily and with little implementation effort, we chose an existing machine learning framework.

YALE (Yet Another Learning Environment) [2] offers a wide variety of algorithms and data processing tools. It also provides a dedicated clustering plugin.

The **Barrier Vector Clustering** experiment is build up from a series of components.

First the **example source data** is read from a file. In the next step a simple **preprocessing** rule is applied. The attributes that have the same value for each barrier vector object are removed because they do not provide any useful information to distinguish the vectors.

After this the **similarity measure** for the algorithm is set up. In the figure the Euclidean distance is selected.

When all the initialisations are completed, the **clustering algorithm** – in this case K-Medoids – is started.

Finally, an **example visualiser** is added to the experiment. This enables different visualisations of the resulting cluster model. Apart from the cluster illustration the program also generates a machine readable cluster representation.

4.2 Experiment: Exploring the Data

The first experiment is designed to explore the data. It also serves as a proof of concept for the barrier vector clustering approach.

As input data we use 200 barrier vectors. 100 from the dutch and norwegian prime minister web site respectively. To get a clue for the choice of K (i.e. how many clusters should be identified) we run the EM algorithm, which determines the number of clusters dynamically.

Figure 3 shows the resulting cluster structure. The algorithm has identified two clusters. Each of the clusters corresponds (almost) completely to one of the web sites. Cluster A contains 100 no-pages and 6 nl-pages, cluster B consists of the remaining 94 nl-pages.

Our hypothesis that pages from the same web site have a similar barrier profile holds in this experiment. This result supports one of EIAO’s modelling assumptions: “It makes sense to aggregate the accessibility results from single pages to an overall result for the web site.” Additionally, the sampling approach of the observatory can be justified. Because of the underlying similarities of the barrier profiles it is not necessary to inspect all the pages from a web site to get an indicator for the accessibility situation of the site. Furthermore, the current sample size of one hundred pages per site seems to be sufficient.

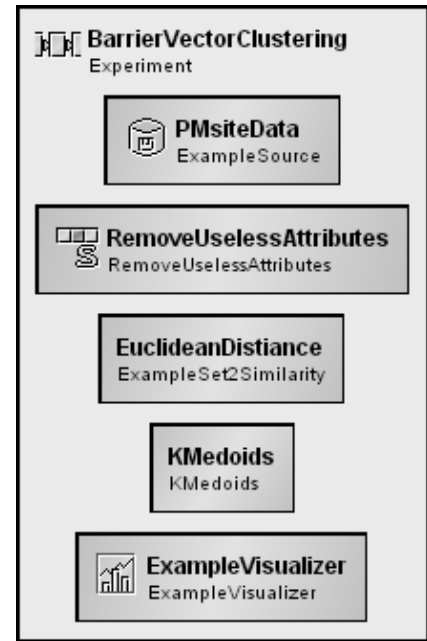


Figure 2: Clustering experiment (YALE)

	euclidean		cosine	
	nl	no	nl	no
Cluster A	97	94	12	0
Cluster B	0	1	5	0
Cluster C	0	1	38	0
Cluster D	1	0	0	49
Cluster E	0	1	8	0
Cluster F	1	0	14	0
Cluster G	0	1	5	0
Cluster H	0	1	1	49
Cluster I	1	0	12	0
Cluster J	0	1	5	2

Table 1: Comparison of the two similarity measures

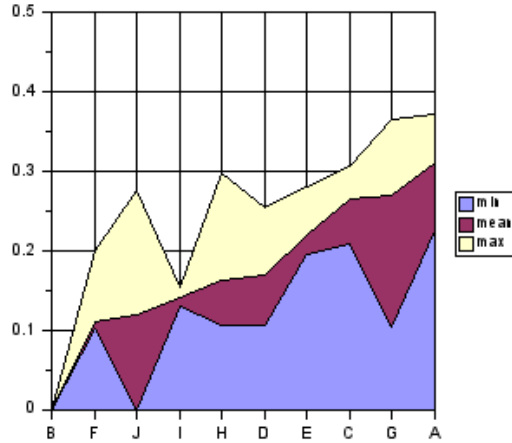


Figure 4: Relation between clusters ('cosine' from table 1) and barrier probability

5 Discussion

The barrier vector model has been shown to provide a useful representation of the accessibility situation of a web page. This section describes applications and extensions of the model.

5.1 Application of the Model: Predicting Accessibility

Once a cluster partition of the barrier vector space is found, it can be used to predict the accessibility situation of further web pages. First, a classifier is constructed from the cluster labels. There are a variety of classification algorithms that can be used in this step. For example, it is possible to apply a *K nearest neighbour* classifier directly to the barrier vector data. The similarity measure used in the classification step should be the same as in the clustering step.

After the training phase, where the classifier is “learned”, new web pages can be assessed. The web pages are evaluated with the automatic tool. Then the classifier is applied to the resulting barrier vector to determine the accessibility category of the page.

A detailed investigation of the outcome is needed to determine which similarity measure and algorithm yields the most useful categorisation. The “usefulness” is evaluated by comparison to the results from expert and user testing.

5.2 Relation to Other Approaches

The clustering approach also supports the aggregation from web page to web site level. After the average barrier vector for the web site has been computed, the procedure described in section 5.1 can be applied.

The clusters could also be chosen to allow a direct translation of the cluster labels into a scorecard presentation (as defined in UWEM 1.0 [6]).

An advantage over other aggregation approaches (e.g. the ones described in [1]) is the “multidimensionality” of the clustering approach. Many features of the data are taken into account simultaneously whereas in other approaches the feature values are summarised into a single number at an early stage.

5.3 Possible Extensions

Further analysis of the data can be performed to identify the dimensions with highest discriminational power (e.g. *principal component analysis* – (PCA)). The feedback from this analysis might allow a dimension reduction of the barrier vector space and thus a reduction of the number of barrier type tests that have to be carried out for each web page.

Another potential area of research is the clustering according to disability group. The task is to find out which regions of the vector space contain web pages with accessibility problems for a certain disability group. It has to be explored if the similarity measures have to be adapted for this task. Furthermore, a detailed user evaluation is needed to guarantee meaningful results.

6 Conclusion

The proposed clustering approach is capable of identifying structures in accessibility assessment data that we are so far unaware of. Therefore it can contribute valuable knowledge in the area of web accessibility assessment.

If the results are convincing and coherent with expert and user judgement the approach might even be usable to determine the accessibility level of web pages only from automatic testing results.

References

- [1] Christian Bühler, Helmut Heck, Olaf Perlick, Annika Nietzio, and Nils Ulltveit-Moe. Interpreting results from large scale automatic evaluation of web accessibility. In *ICCHP 2006*, LNCS 4061, pages 184 – 191. Springer, 2006.
- [2] Simon Fischer, Ralf Klinkenberg, Ingo Mierswa, and Oliver Ritthoff. *YALE: Yet Another Learning Environment – Tutorial*. Collaborative Research Center 531, University of Dortmund, no. ci-136/02 edition, 2002.
- [3] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer, 2001.
- [4] Matthew King, James W. Thatcher, Philip Matthew Bronstad, and Robert Easton. Managing usability for people with disabilities in a large web presence. *IBM Systems Journal*, 44(3), 2005.
- [5] W3 Consortium. Web content accessibility guidelines 1.0. Available at <http://www.w3.org/TR/WCAG10/>, 1999.
- [6] Web Accessibility Benchmarking Cluster. D-WAB4 Unified Web Evaluation Methodology (UWEM 1.0). Available from <http://www.wabcluster.org/uwem1/>, 2006.
- [7] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11), 1975.