A Survey of SOM Based Approaches to Document Classification

Chris Jordan

Technical Report CS-2003-10

December 2003

Faculty of Computer Science
6050 University Ave., Halifax, Nova Scotia, B3H 1W5, Canada

A Survey of SOM Based Approaches to Document Classification

Chris Jordan
cjordan@cs.dal.ca
December 2003
Dalhousie University

Table of Contents

Abstract

Document classification has been an object of study for many years. The Web along with other technologies such as digital libraries has facilitated the growth of document collections both in size and popularity. Classification is important as it allows users to effective browse and to quickly understand the general contents of these corpuses. Most approaches to classification have used supervised learning. This has been acceptable in the past since collections have been small enough for teams of experts to generate representative training datasets; this is not feasible for very large corpuses. Such repositories require unsupervised learning methods that will be able to find the clusters with a minimum of human direction. A self organizing map (SOM) is one such clustering algorithm which places clusters that are similar to other each close on a lattice. It is attractive in that it offers users an intuitive interface for browsing the document collection. This paper will discuss what a SOM is and what data preprocessing needs to be done before it can be employed on a document set. Following this, an analytical survey of four popular works involving SOM based algorithms for document classification will be presented.

1. Introduction

Document classification has been a subject of analysis long before the days of the Internet. Basic examples of this are how articles are divided into sections in a newspaper and how filing systems are employed in office routines. It is a primary activity in information management. Its purpose is two fold; it presents an intuitive interface to users for accessing documents while also giving a quick understanding of the collection's general contents. Effectively categorized document sets allow users to quickly determine which items will satisfy their information needs and which will not.

With the introduction of the Web, information accessibility levels soared. Users across the world could distribute and retrieve information on virtually anything. A side effect of the Web is information overload, there is more data available than can be reasonable processed. This hampers the usability of this medium; users experience difficulty in finding information that will satisfy their needs. To combat information overload, the same collation activities that were used in offices and libraries are being applied to the Web. However, this transition is far from being smooth. The problem that many researchers in this field are encountering is scalability; most collation systems were designed to handle collections that contain thousands or millions of items, not the billions that the Web has. A further complication is that most collation systems assume that the collections they assimilate are static; the Web however is dynamic.

Currently there is no single solution for classification of Web documents. Its size and volatility pose significant problems. Furthermore, it is not heterogeneous; it contains a large assortment of different types of data such as images, sounds, videos, and software. These issues will not be discussed in this paper; the focus will be on the classification of text documents which makes up about 80% of the Web currently. While the amount of multimedia content on the Web is growing, analysis of it here will only detract from the main point of the paper.

Self organizing maps (SOM) are a form of competitive neural network. The original algorithm was derived by Kohonen in 1982. The basic architecture of a SOM consists of a lattice that acts an output layer with input nodes connected to it. Input patterns that are judge to be similar are mapped to the same node and related patterns are placed nearby. When a lattice is one or two dimensions, it creates a visual representation of the different clusters that exist in the document set where similar ones are physically near each other. The SOM has the advantage over other clustering procedures in that it is able to implicitly determine the similarity between groupings. Greater discussion of the SOM will be carried out in section 3.

This paper is broken into five sections including this one. The second section will discuss previous approaches taken to document classification. The third will offer an introduction to the SOM algorithm. The SOM itself is not a complex function but it is mathematically difficult to analyze; this analysis will not be explored in this paper. The fourth section will present a survey of four SOM based document classification systems. The final section of the paper will close with some concluding remarks about document classification and the SOM.

2. Related Work

Many approaches, both supervised and unsupervised, have been used for document classification. Supervised learning algorithms require training data that represents the problem domain well in order to derive rules for classification. Creation of this data can be very prohibitive as it demands the creators to have extensive knowledge of the documents and the problem domain; this is amplified when the corpus under examination is very large or complex. Errors in the training data can result in poor performance by these classifiers. Unsupervised learning has no such requirement for training data and thus no knowledge of the document set is need. Learning algorithms of this form will find clusters in a repository and each cluster can be viewed as a derived class. The SOM is a form of unsupervised learning as document clusters are formed on its lattice output layer; this layer is often called a feature map.

Supervised Learning

These approaches can be segregated into two groups, binary classifiers and m-ary classifiers. Binary classifiers indicate whether a document should belong to a category or not. For example, given a set of classes, a binary classifier will indicate which classes a document should belong to and which it should not. M-ary classifiers alternatively produce a ranked list of classes for a document. This section will highlight some of the more popular supervised learning algorithms that have been examined in the past for

document classification. A complete survey of these approaches is present in [(1)Yang; 99] and [(2)Yang; 99]. Yang discusses the results of extensive experiments he has conducted on four versions of the original Reuters corpus to compare these supervised learning algorithms; empirical evidence is presented to help illustrate the true performance differences between them.

Binary Classifiers

Manually developed approaches:

One of the first approaches to document classification on the Web was to use human operators. Organizations such as Yahoo (www.yahoo.com) and the DMOZ (www.dmoz.org) employ people to classify Web documents manually. While this is a very straightforward solution it encounters many problems. The first is the reliance on human operators; people are prone to being bias and making mistakes. Using human operators will introduce human error which will result in documents being misclassified. A second problem is scalability; there are so many documents available on the Web that it is impractical to organize them all by hand. A third complication is that the resulting classification structure is brittle and not suitable for volatile collections. While this approach does seem to be plagued with downfalls it does provide a good baseline for comparing other classification algorithms to. Furthermore, the classified document sets that are produced by these operators are very useful for supervised learning approaches. From this perspective, manual classification can be used to better autonomous methods.

Decision Trees:

This is a very popular form of supervised learning. A tree is constructed by selecting features in the training data that partition it well until a leaf, a point where separating the data further will not gain any additional insight, is reached. The features that are selected to divide the data have high levels of entropy. Documents to be classified are then passed through the tree; the leaf that a document arrives at determines the class. The power of decision tree algorithms is that symbolic rules for classification can be extracted from its

structure; these rules are derived knowledge regarding the document set. They are built from if and else statements. Popular decision tree algorithms are ID3, C4.5, and C5. The drawback of decision trees is that they are not very scalable; in constructing them, several passes through the training data must be made. This is particularly costly if not all the training data can be placed in main memory.

Naïve Bayes:

This is a probabilistic technique for classification. Basically this algorithm computes the probability of a document belonging to a class based on the terms that make it up; these term probabilities are estimated from the training data. The main advantage of the Naïve Bayesian classifier is that it is relatively easy to examine mathematically. This simplicity comes at a cost though; the assumption that terms are independent of each other is made which is obviously not true in reality. A more severe hindrance is that words and phrases in the English language suffer from polysemy, having multiple meanings, and synonymy, many terms having the same meaning. Even though the Naïve Bayesian calculation is not a true reflection of the real world problem, it has shown to perform reasonably well.

Rocchio Prototyping:

In this approach, the documents in the training set are separated into their respective classes. Then for each class, a prototype is constructed. This is done using the vector space model where the prototype vector for a particular class is composed by summing all the training vectors that belong to it and subtracting all the vectors that do not. Documents are then compared to these class prototypes to determine which one it belongs to. This algorithm has normally been used for filtering in retrieval systems. While prototype construction is computationally easy, they are not as effective at document classification as other methods.

Neural Networks:

Self organizing maps are not the only neural network approach taken to document classification. In [Wiener, Pedersen, and Weigend; 95], experiments were carried out on two types of neural network architectures, flat and modular. Neural networks naturally apply themselves to classification as they are trained to map input patterns to desired output patterns or in the case of document classification, documents to classes. During this process, these networks are able to determine which input features signify a mapping to a particular output node. Another advantage of neural networks is that a single network is capable of simultaneous classification of multiple classes. A problem with neural networks is that they do not produce set of symbolic rules; they are very much black boxes where input is fed in and output is returned. A second drawback experienced with networks is that they necessitate a substantial amount of training often requiring several thousand passes through the training data for a complex problem.

M-ary classifiers

K-Nearest Neighbors:

This is an alternative procedure for classification. Instead of learning a mapping or relationship between a group of documents and a class, documents that are to be classified are compared to the training set. The K most similar items in the training set make up a document's neighbors. The classes a document should belong to are derived from what classes the neighbors are from. This is different from other classification methods in that most of them encounter computational overhead in training. In this approach, the bulk of the computation is performed during analysis. This can be prohibitive if there are a large number of documents to be classified and K is set to be very high.

Unsupervised Learning

These types of algorithms find clusters that exist in a dataset. The strength that clustering methods have is that no a priori knowledge is required about the data before examination. This allows users to quickly understand what types of documents are in a collection. This advantage is also a drawback though as it is very difficult to incorporate any a priori knowledge that may exist into the analysis. There are two types of clustering, nonhierarchical and hierarchical. Nonhierarchical methods require that the number of clusters to be found be specified beforehand. These approaches have proven to be ineffective as it is unrealistic to assume that the number of clusters in a dataset is known a priori; nonhierarchical methods will not be discussed any further in this paper. Contrary to this, hierarchical approaches will discover the number of clusters in a set.

There are two avenues that a hierarchical clustering algorithm can take. They are agglomerative, where every document starts as its own cluster and similar clusters are merged together, and divisive, where all documents start out belonging to the same cluster and clusters are broken into smaller ones. Divisive algorithms are uncommon compared to agglomerative. The following clustering algorithms are all agglomerative. The problem with these schemes is that they are computationally expensive. Ignoring single link, the below clustering methods have complexities of at least $O(N^2)$ where N is the size of the document collection. It is simply unreasonable to apply these algorithms to large repositories such as digital libraries or the Web. [Rasmussen; 92] provides a greater discussion on these clustering methods applied to information retrieval.

Single Link

This is the most computationally easy of all the clustering approaches. Groups are formed by joining the most similar items which are not in the same cluster. The shortfall of this is that clusters formed by this procedure tend to be very loose. As well, the formation of the

clusters depends on the order which items are examined in. This means that it is difficult to reproduce the clustering results using single link.

Complete Link

Contrary to single link, complete link determines cluster similarity by using the least similar documents between them. This means that when clusters are merged, the similarity between documents is at least equivalent to the similarity of the clusters. This results in tighter clusters being formed.

Group Average Link

Cluster similarity is determined in this approach by computing the average document similar between clusters. This results in the clusters being neither tight nor loose.

Ward's Method

This approach is also referred to as the minimum variance method as analysis is performed by merging clusters that minimize the total inter-cluster sum of squared errors, a calculation based on the distance measure between items within the cluster and the centroid. Consequently, the clusters that result from this method are usually very homogeneous in nature.

A complication with document classification that plagues all of these methods as well as the SOM is that they were not designed to handle patterns with large numbers of attributes which is exactly what the vector space model for most document collections possess. As a consequence, dimensionality reduction measures such as latent semantic

indexing and term selection have to be taken when employing classification techniques to document corpuses. These methods will be discussed in the following section.

3. What are Self Organizing Maps

The SOM

A Self Organizing Map (SOM) is a powerful variant of neural network; it is a Hebbian Competitive network. It is competitive in that during the learning phase there is a winner takes all competition held between all the nodes in the output layer for every input pattern; this is an unsupervised learning function. It is Hebbian since once a winner has been decided for a particular input pattern the weight vectors of that node and its neighbors are adjusted. To facilitate this notion of distance in the output layer, the nodes are placed on a lattice. The result of training a SOM is a lattice of document clusters where similar clusters are close in proximity to each other. This is similar to how the human brain is divided into different sections where each one is responsible for different sensory input.

While the SOM is able to determine complex clusters, the basic algorithm for it is relatively simply. There are five basic steps:

1. *Initialization*: The connection weights from the input layer to the output lattice have to be randomly initialized. There is only one restriction; nodes in the lattice can not have to same set of connection weights. Another common practice is to use small values for initialization.

2. *Sampling*: During the learning phase, input patterns, the documents, are selected with uniform probability from the corpus.

3. *Similarity Matching*: The input pattern is applied to the current network. The output node with the greatest induced field is the winner. In essence, the connection weights act as a pseudo document for the input to be compared to. The induced field is simply the dot product, similarity calculation, of the two.

4. *Updating*: Once the winning node is found, the connection weights belonging to it and its neighbors are adjusted. In the first SOM algorithm Kohonen developed, all nodes in the neighborhood including the winner were adjusted equally. Current versions of SOM have altered this so the adjustments are less dramatic the further a node is from the winning one.

5. *Continuation*: The process is repeated until an epoch occurs where there are no noticeable changes made to the connection weights.

For a more detailed explanation of the SOM algorithm and its properties, refer to [Haykin; 1999]. From a high level of abstraction, a SOM is nothing more than a two layer unsupervised neural network that uses Hebbian learning. Its ability to not only cluster input patterns well but to also to place similar cluster close to each other spatially is remarkable. The mathematical justification of the SOM algorithm is beyond the scope of this paper.

The SOM is not without its drawbacks; it is susceptible to the same ailments as other neural network algorithms. In order to use a SOM effectively, the number of dimensions that input patterns can have has to be as small as possible. This means dimensionality reduction has to be performed. A further problem is that it is still computationally expensive to training the SOM often requiring several thousand passes through the dataset. There are some optimizations that can be performed to decrease this overhead; they will be discussed in section 4. One final down back is that it is often required to train

a SOM several times before getting an acceptable feature map; this is due the fact that the weight matrix is randomly initialized.

Dimensionality Reduction

The major obstacle in the classification of a document set is that they typically contain a very large vocabulary of terms, English words and phrases. The most common approach to represent a document is to use the vector space model [Baeza-Yates and Ribeiro-Neto, 1999] hence an input pattern for a SOM would be a document vector. The fault with this is that the number of elements in a document vector equals the number of terms in the vocabulary. This means the input patterns will have a large number of dimensions resulting in high computational overhead in training the SOM. A further problem is that terms suffer from polysemy and synonymy. This leads to ambiguity if documents are being classified based on term occurrences alone. A solution to this is dimensionality reduction; this works by grouping like terms into single dimensions. This implicitly gives contexts to terms thus negating their ambiguous nature. Two popular algorithms for this are latent semantic indexing and random mapping. Just a general overview of these algorithms is offered by this paper; a detailed explanation of them is beyond its scope.

Latent Semantic Indexing

This algorithm was developed specifically to tackle the issues of synonymy and polysemy in the vector space model. It is accomplished by constructing a term/document matrix, X, from the document set. Once X is formed, the singular value decomposition (SVD) can then be performed on it to get the following result:

$$X = T_0 S_0 D_0$$

Where T_0 and D_0 have orthonormal columns and S_0 is diagonal. Since S_0 is diagonal, it is possible to order its diagonal elements. This is important as it allows the elements to be arranged in decreasing order. A smaller matrix that approximates X can be created by

keeping the k largest elements in S_0 and setting the rest to zero. This results in the following:

$$X \approx \hat{X} = TSD$$

This approximation of X is a dimensionality reduced version. The problem in computing the SVD is that it has a polynomial computational complexity. The complexity of latent semantic indexing is O(Nld) where N is the number of documents in the collection, l is the average number of terms in a document, and d is the resulting desired dimensionality [Kohonen, Kaski, Lagus, Salojarvi, Honkela, Paatero, and Saarela; 2000] however the SVD is not fully computed in this algorithm. Full computation of the SVD, while it will increase performance, is not entirely necessary and doing so will increase the complexity of latent semantic indexing substantially. For more information regarding this approach refer to [Deerwester, Dumais, Furnas, Landauer, and Harshman; 90].

Random Mapping

Random mapping offers an efficient means to compress the dimensionality of an input space. It was first presented in [Kaski; 98] as a means to decrease the computational complexity of similarity calculations in clustering. The algorithm is quite simple:

$$x = Rn$$

The input vector is n and the dimensionality reduced vector is x. R represents a random mapping matrix. R is composed of random weights that are then normalized so every column has unit length. At first glance this approach does not seem to be logical. Intuitively, in order for R to reduce n effectively it should consist of only orthonormal columns. However, in input spaces that have a large dimensionality there exist a large number of almost orthogonal directions which can be used in R as approximations for true orthogonal columns. In order for randomly generated vectors to be used as good approximations for the columns of R the dimensionality of the input space has to be very

large. In fact, the greater the dimensionality and the greater the sparseness of the input vectors, the better random mapping will perform; this makes document collections ideal candidates for the algorithm. It has been shown in [Kohonen, Kaski, Lagus, Salojarvi, Honkela, Paatero, and Saarela; 2000] that random mapping is just as effective as latent semantic indexing, a more computationally intense approach. The complexity of random mapping is O(Nl) + O(n) where N is the size of the document collection, l is the average number of terms contained in a document, and n is the original dimensionality.

There is a problem with dimensionality reduction; the more dimensions that are removed from the document set's representational model the more information that is lost. This problem is amplified by the fact that documents are rarely distributed evenly amongst the clusters in a collection. If enough dimensions are removed, smaller clusters will be screen out and viewed as inconsequential. This should be taken into account, whenever dimensionality reduction is used, to minimize the chance of it occurring. The reason that the dimensions of a document set can be reduces so radically is due to document vectors typically being very sparse; an individual document will only contain a small portion of the entire vocabulary. As well, terms affected by synonymy can be squashed into a single dimension without encountering any information lost.

It is important to realize that dimensionality reduction is an unavoidable process when using a SOM for document classification. Classification is a data mining activity and a fundamental activity that must be carried out is data preprocessing. The data has to be cleaned of noise and impurities in order for optimal results to be obtained. The vocabularies that compose document sets are inherently noisy due to synonymy and polysemy. A further reason that reduction must take place is that the vocabularies are very large and clustering document vectors composed of them is computationally demanding; the more dimensions that can be remove the faster the similarity calculations can be carried out and thus cluster formation is achieved more quickly.

4. Document Classification using Self Organizing Maps

This section will present a summary of self organizing map (SOM) approaches to document classification. In each of the following subsections, a discussion of the research objectives, the proposed solution, the research methodology, description of the data, the experimental set-up, and the reported outcome will be given. Recommendations for improving the quality of the results are made along with conclusions. Comments on future areas of work and research are deferred to the final section.

WEBSOM

The roots of this work can be traced back to 1982 when Kohonen initially developed the SOM algorithm. From 1996 to 2000 Kohonen lead the WEBSOM project. WEBSOM uses the SOM architecture for clustering massive document collections. The work discussed here can be found from the final WEBSOM publication, [Kohonen, Kaski, Lagus, Salojarvi, Honkela, Paatero, and Saarela; 2000].

Research objectives

The focus of the work on WEBSOM was to improve the scalability of the algorithm. The computational complexity of clustering algorithms has prevented them from being used on large datasets. It is important that attempts to improve the scalability of WEBSOM do not have adverse effects on its clustering performance.

Proposed solution

A variety of optimization techniques have been implemented in WEBSOM to decrease the computation required to create a feature map. This includes the following:

Estimation of a larger map based on a carefully constructed smaller one

In this technique, the initial weight matrix for a large map is generated based on the asymptotic connection weights found in a much smaller map. The intent of this is to initialize the weights in a large map with good values so convergence can occur faster.

Parallelized Batch Mapping

Batch Mapping is an optimization for training a SOM. It works by using vector quantization and numerical smoothing. The first step is to initialize the weight matrix of the map. Then for every output node find the set of documents that are closest to it. For each set, a one step vector quantization is computed to derive a representative, an average of all the documents in a particular set. This representative is then used to alter the weights of the output node. This process is repeated until the node does not experience any significant weight changes. Batch Mapping lends itself well to being parallelized where the computation for each output node can be distributed. Parallelized Batch Mapping is simply that.

In addition to the above methods, WEBSOM also uses several memory saving techniques. They are not discussed in this paper as they do not make any approximations on the data; they simply use the computing hardware to the full extent of its capabilities.

Research methodology

To determine the effects of these optimization techniques a comparative analysis is conducted between the traditional SOM algorithm and WEBSOM, a SOM algorithm optimized for scalability. Comparing the classification accuracies (the separation of different classes), average quantization error (the average distance of each input to the closest output node), and performance time will indicate if the WEBSOM approach is

preferable to the traditional SOM. If it can be shown that WEBSOM is superior, it will be applied to a substantially larger document corpus for evaluation.

Description of the data

The document collection for this experiment was a database of 6, 840, 568 patent abstracts. These documents were gathered from the U.S., Europe, and Japan patent offices. For the comparative analysis between SOM and WEBSOM, a small subset of this collection, consisting of 13, 742 documents, was used.

Experimental setup

For the comparative analysis, two sets of maps are generated using SOM and WEBSOM respectively. Each set consists of five maps generated from different random weight initializations and trained for 150 iterations. The Batch Mapping portion of WEBSOM was performed for 5 iterations.

For the analysis of the entire corpus by WEBSOM, the dimensionality of the input space was reduced from 733, 179 to 500. The initial SOM consisted of 435 output nodes and was trained using the original SOM algorithm for over 300, 000 iterations. This map was enlarged twice by a factor of 16 and once by a factor of 9. After each time the map was scaled, it was fine tuned with 5 iterations of the Batch Mapping algorithm.

Both of these experiments were conducted on a six processor SGI Origin 2000. The parallel capabilities of this machine were not used in the comparative analysis though.

Reported outcome

Comparative Analysis

The following results were discovered from comparing the traditional SOM to WEBSOM.

	Classification Accuracy (%)	Quantization Error	Time (s)
Traditional SOM	58.2 ± 0.2	.799 ± 0.001	2550 ± 40
WEBSOM	58.0 ± 0.2	.798 ± 0.002	241 ± 3.5

The following computational complexities have been derived.

	Computational Complexity
Traditional SOM	$O(dN^2)$
WEBSOM	$O(dM^2) + O(dN)\ O(N^2)$

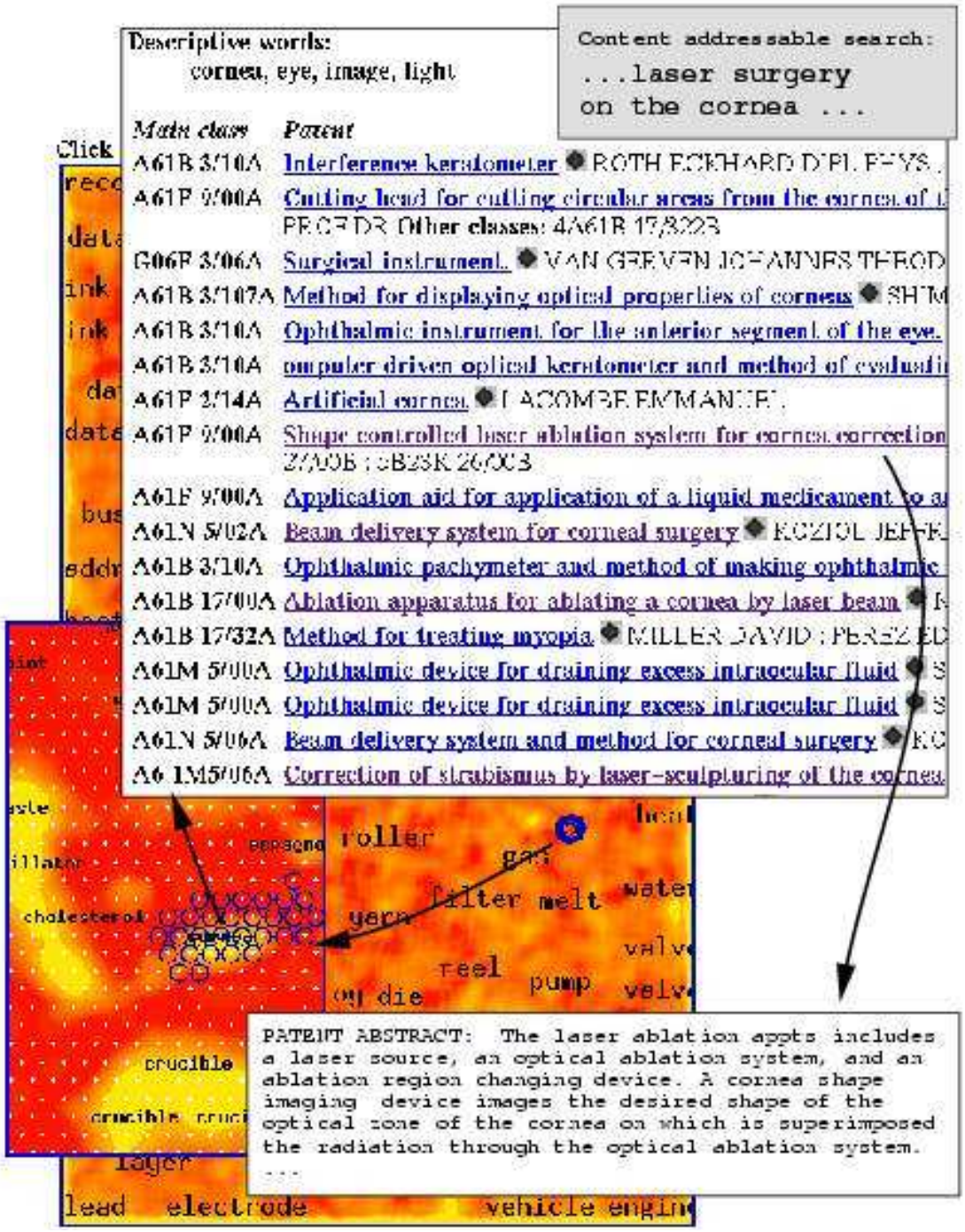
N stands for the size of the document collection, d is the dimensionality of the input vectors, and M is the number of output nodes in the small map used to approximate the weight matrix for the large map in the WEBSOM approach.

Corpus Analysis

Once WEBSOM finished the training phase, nodes on the map were labeled. This was accomplished by examining the documents that were mapped to a particular node and determining what subsection the majority of them belonged to; the subsection that made up the majority was the label for the node. Documents that are mapped to a node which is

labeled as a subsection that they do not belong to are considered as misclassifications. The accuracy, also known as node purity, was measured as 64%. 6 weeks of processing time was required to create this map.

The following is a diagram from [Kohonen, Kaski, Lagus, Salojarvi, Honkela, Paatero, and Saarela; 2000] illustrates how browsing is conducted through WEBSOM.



Browsing is accomplished by selecting a particular area of the map to zoom in on. After a series of zooms it is possible to select individual nodes. Selecting a node will produce a listing of hyperlinks to all the documents that have been mapped to it. A demonstration of

WEBSOM applied to a corpus of over a million documents from 80 Usenet newsgroups exists at the following URL: http://websom.hut.fi/websom/milliondemo/html/root.html

Recommendations & Conclusions

It has been shown that WEBSOM substantially outperforms the traditional SOM. As well, it has been shown that it can be used to perform clustering analysis on very large document collections. However, this analysis did require 6 weeks of processing on a 6 processor SGI Origin 2000. This implies that WEBSOM is not suitability for very large document collections that are also very volatile. It was not indicated in the publication whether or not the SGI Origin 2000 ccNUMA architecture played a role in enhancing computational performance.

The main issue with the results reported on WEBSOM is that it performed clustering on a document collection that is not publicly accessible. This makes it very difficult to reproduce the experiments that were performed. While the availability of a standard corpus of such magnitude does not exist, it is confusing as to why it was decided to construct such a collection that ultimately would not be publicly available.

A lot of complications with the WEBSOM results are due to a lack of knowledge regarding the data. The highest reported classification accuracy was 64%. This seems to be an unacceptable result however it is difficult to gauge whether this result is actually negative or positive since no other classification approaches have been conducted on the dataset. In other words, these results do not really indicate how well WEBSOM is performing compared to other approaches. In reality, 64% classification accuracy is unacceptable for any production document classification system. This result begs the question as to whether this corpus was poorly labeled or if it is just difficult to perform clustering on. This can only be determined by further inspection of the data and performing other classification experiments on it using different algorithms.

It seems that the WEBSOM project has overlooked a particularly important aspect, usability. Nothing is mentioned as to how usable the maps are by users. This is mostly due to the fact that an effective test group would have to be composed of users with intimate knowledge of patents; mustering this group would be very difficult.

The WEBSOM project has shown to be a definite improvement over the traditional SOM. However, it still remains to be seen if it can be used in a production environment effectively. This can only be determined by applying it to corpuses where test groups can be manufacture to evaluate its usability. This is discussed in the following subsection.

M-SOM

This is the document classification project lead by H. Chen in 1996. One of the drawbacks of using a SOM is that clusters tend not to be similar in size; bloated nodes are difficult for users to browse. This is not the fault of the SOM algorithm as the clusters are very cohesive but results from the nature of document sets. Typically document collections contain various size clusters. It is not uncommon for collections to be predominantly composed of a few groupings; this is a reflection of the 80/20 rule of thumb where 80% of the documents will belong to 20% of the clusters. This can cause complications if a SOM is applied to a very large corpus where too many documents can be mapped to one node for a user to browse logically. M-SOM is a recursive SOM that will examine bloated nodes in the hope of enhancing the users browsing capabilities. More information regarding the M-SOM project can be found in [Chen, Schuffels and Orwig; 96].

Research objectives

The primary objective of M-SOM is to categorize Web documents autonomously in an efficient and maintainable manner in addition to offering users an intuitive browsing

interface. The most popular Web classification systems, Yahoo (www.yahoo.com) and the DMOZ (www.dmoz.org) are maintained manually by human operators. This approach is prone to human errors and requires a prohibitive amount of human effort thus making it impractical for most enterprises to implement. The intention of the M-SOM project is to develop a less costly alternative to this.

Proposed solution

M-SOM is a recursive SOM algorithm; once the initial map is created using the traditional SOM algorithm, nodes that have more than a specified maximum number, k, of documents mapped to it will be examined by a SOM. This creates a feature mapping of the documents mapped to that node. This process is repeated until every node, that has more than k documents mapped to it, has feature map constructed. Whenever one of these bloated nodes is selected during the browsing process, the representative mapping will be shown.

Research methodology

M-SOM is first applied to a small corpus to evaluate its clustering capabilities. Then M-SOM is applied to a large corpus. The corresponding mapping that is generated is then evaluated for usability by a test group of users.

Description of the data

The small data set that is examined first is composed of comments generated during a 30 minute brainstorming session; there are 201 comments in this set. The large dataset was populated with Internet entertainment homepages. The set consists of over 10, 000 homepages that were gathered from the Yahoo server using a Web spider.

Experimental setup

In the experiment on the brainstorming comments, a comparative analysis was carried out between a manual categorizing tool operated by an expert group and M-SOM. Each generated a list of topics and the corresponding comments that belong to them. These two lists were then given to another expert group that corrected them by add or removing topics. The new resulting lists were used to calculate precision and recall.

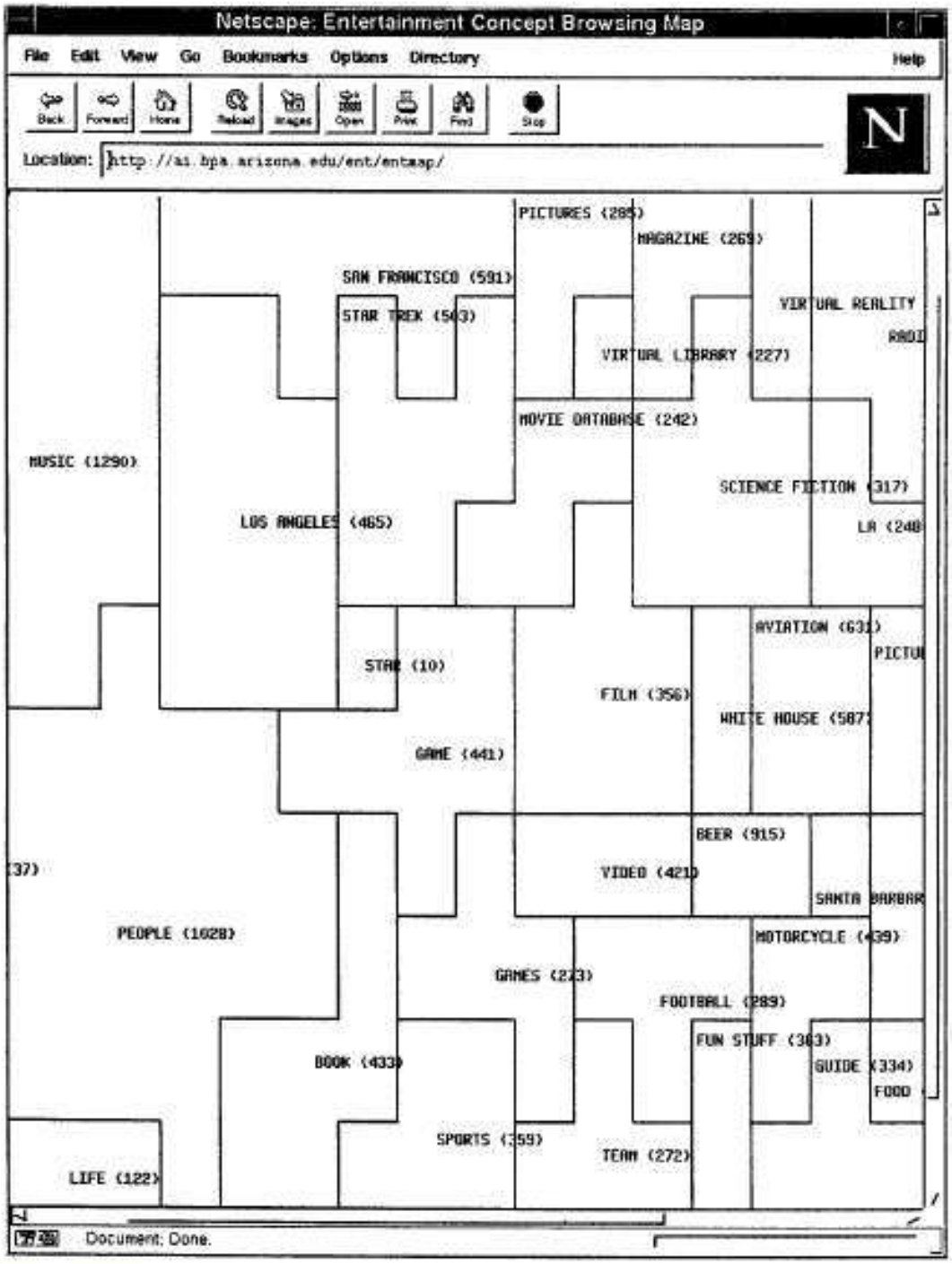
A DEC Alpha 3000/600 (200 MHz, 128 MB RAM) was used to compute the feature map for the experiment on the entertainment homepages where k, the number of documents that have to be mapped to a node for a feature map to be created for it, is set to 100. A test group consisting of 9 graduate students and 1 system administrator was used to evaluate the usability of the map. The users were given three searches which were performed on both the M-SOM generated mapping and the Yahoo webpage hierarchy. The evaluation of these systems was based on the time required for the users to perform the searches.

Reported outcome

Brainstorming Comment Analysis

	Precision (%)	Recall (%)	Computation time (min)
Categorizer tool with expert group	81	88.5	45
M-SOM	55	81	4

M-SOM required a few hours to generate the feature map for the homepages. Most searches that were conducted by the test group required 1 to 10 minutes to complete. A few searches required less than a minute and a few required more than 10. Overall, the test group found that both systems were difficult to use. The following is an image of the top most level from feature map generated by M-SOM.


Netscape: Entertainment Concept Browsing Map
File Edit View Go Bookmarks Options Directory Help
Back Forward Home Reload Images Open Print Find Stop
Location: http://ai.bpa.arizona.edu/ent/entaap/
PICTURES (285)
MAGAZINE (269)
SAN FRANCISCO (591)
STAR TREK (503)
VIRTUAL REALITY
RADI
VIRTUAL LIBRARY (227)
MOVIE DATABASE (242)
MUSIC (1290)
SCIENCE FICTION (317)
LOS ANGELES (465)
LA (248
AVIATION (631)
PICTU
STAR (10)
FILM (356)
WHITE HOUSE (587)
GAME (441)
BEER (915)
37)
VIDEO (421)
SANTA BARBAR
PEOPLE (1628)
MOTORCYCLE (439)
GAMES (223)
FOOTBALL (289)
FUN STUFF (303)
BOOK (433)
GUIDE (334)
FOOD
SPORTS (359)
TEAM (272)
LIFE (122)
Document: Done.

Recommendations & Conclusion

While the importance of computational efficiency was brought up, the main focus of this project was on the usability of the feature maps create by M-SOM. The initial experiment on the brainstorming comments was used to illustrate that M-SOM could provide reasonable classification results quickly. The main experiment, on the entertainment homepages, was designed to discover if the resulting feature map was intuitive for users to browse or not and how it compared to a traditional webpage hierarchies such as the one maintained by Yahoo. The unfortunate result is that the test group found neither system to be very effective.

For both systems, the test group encountered the same problem; it was very easy to become lost in "cyberspace", the webpage hierarchy. The complaints that the test group made about M-SOM were that the clusters did not seem to be logically created or organized. This is attributed to two reasons. The first is that document collections tend to be very complex thus difficult to cluster. The solution to this problem exists in enhancing the data preprocessing phase; perhaps introducing a feature sensitive calculation in combination with the dimensionality reduction process will generate improvements. The seemingly randomness of the cluster locations is due to the fact that the connection weights in a SOM are initialized with random values. To solve this, a method needs to be developed to determine logical starting values for connection weights.

It appears that feature maps developed by SOM algorithms are not as intuitive to navigate as originally hypothesized. However, this result is not entirely surprising due to the stochastic nature of them. In order to develop more logical mappings, more logical initial connection weights have to be derived. It is this author's opinion that the answer lies in developing a method for incorporating a priori knowledge into the SOM; this new approach will be a semi-supervised neural network.

PRIS

PRIS is a SOM based document classification project lead by Azcarraga. One of the major obstacles in using a SOM is the long training times required. This is a problem intrinsic to neural networks; the training data must be examined over several iterations. The goal of PRIS is to find a method to reduce the number of documents that must be examined in order to form an effective feature map. The details of this work can be found in [Azcarraga and Yap; 01].

Research objectives

The aim of PRIS is to determine if summarizing the training data through use of prototyping will still yield effective feature maps. Prototyping will result in a set of exemplars that represent the corpus. This set will be orders of magnitude smaller than the original collection. Thus training using these prototypes will be orders of magnitude faster.

Proposed solution

The SOM will be applied to a set of prototypes that represent the corpus. These prototypes will be constructed during the data preprocessing phase via an incremental learning algorithm. Once the feature map is constructed, documents are then feed into it. A document is attached to the node with the most similar prototype mapped to it.

Research methodology

A comparative analysis is conducted between PRIS and WEBSOM. The clusters produced by both of these algorithms will be examined for similarities. This analysis is

performed as PRIS is simply the WEBSOM algorithm with a prototyping step added into the data preprocessing phase. If the clustering results produced are similar then it will prove that the information lost through prototyping does not significantly affect the performance of the SOM.

Description of the data

The original Reuters corpus is examined in this comparative analysis. This is a standard corpus for experimentation and consequently has been the subject of analysis in many papers. Depending on the version, it consists of over 21, 000 news articles produced by Reuters.

Experimental setup

16 by 16 node feature maps are generated by WEBSOM and PRIS. The similarity analysis is conducted by selecting every pair of documents mapped to a node in one map and measuring the distance between them in the other map.

Reported outcome

Experiment	Distance between document pairs (accumulative percentage)				
	0 nodes	1 node	2 nodes	3 nodes	4 nodes
Comparing document pairs from WEBSOM to PRIS	43	68	76	80	83

Comparing document pairs from PRIS to WEBSOM	27	61	76	82	85

Recommendations & Conclusion

The focus of PRIS was to determine if prototyping was a viable approach for decreasing the size of the training set. From initial inspection of the results, 43% of the document pairs mapped to a node in the WEBSOM map are mapped to the same node in the PRIS map. However, the opposite is not as promising as only 27% of the document pairs are found together. This indicates some of the nodes in the PRIS map have a large number of documents mapped to them. It would be interesting to see if some of the nodes in the PRIS map contain documents pairs from different nodes in the WEBSOM map. This would indicate that PRIS could generalize the corpus at a higher level than WEBSOM.

Further testing is required on PRIS to determine how well it performs. Considering that the Reuters corpus is labeled, it would be trivial to determine the classification accuracy of PRIS and WEBSOM on it. As well, it would be interesting to see how PRIS would perform on the 7 million patents document collection that WEBSOM was tested on previously. Comparing the classification accuracies would truly determine which approach performs better and by what factor.

Computation times should have been recorded in this analysis. It is obvious that some information was lost from prototyping. Assuming that WEBSOM outperforms PRIS, the differences in computation times is an important factor that has to be considered. If the computational gains are substantial then the lost in classification accuracy may be negligible.

Another experiment that needs to be carried out is on the effects of varying the threshold parameters for prototyping. In particular, there is a similarity threshold that has to be set to determine if a new exemplar has to be generated to represent a document or not. Varying this threshold will vary the level of generalization that will be achieved by prototyping. This will directly effect how the feature map is constructed. Determining the best values for these thresholds is paramount in achieving optimal performance with PRIS. An interesting characteristic of PRIS is that if the similarity threshold during prototyping is set to an extreme where documents must exactly match the prototype then PRIS reverts to the WEBSOM algorithm.

Corpus reduction for training purposes is an issue that has to be addressed when using a SOM on very large datasets. The size of the corpus is directly related to the computation time. It is evident there are some underline redundancies in these giant collections. A method that removes them will minimize the effect that a corpus will have on computational overhead.

Distributed SOM

One assumption that has been made in previous SOM based approaches is that the corpus is in a single location. In reality, very large corpuses tend to be distributed. While merging these partitions into a single corpus is one solution, computing a SOM on each one and then merging the resulting feature maps may provide a more computationally efficient scheme. Research regarding this can be found in [Rauber and Merkl; 98]

Research objectives

The purpose of this work is to determine if it is possible to create a reasonable feature map by merging several smaller ones. If this is feasible then computation of the SOM can then be distributed which will decrease training time. Furthermore, this can be applied to

distributed corpuses which are found in digital libraries thus making each partition responsible for developing its own feature map.

Proposed solution

For each partition of the corpus, a feature map is constructed. To merge these into a single large SOM, the weight vectors of all the nodes in every map are used as input patterns. In this fashion, the clusters from the smaller maps are preserved and simply arranged on the larger map.

Research methodology

A corpus is divided into five random portions; each examined by the SOM algorithm. This results in the creation of five 7 by 7 node feature maps. The maps are then merged into a single 7 by 7 node SOM. This SOM will then be visual inspected for cluster cohesion.

Description of the data

The corpus used during this experiment was the 1990 edition of the CIA Worldfactbook. It consists of 245 documents, each regarding the geographic, economic, and political characteristics of a particular country, region, or island. Five partitions were created randomly, each containing 50 documents. This means five documents were represented twice.

Experimental setup

The five lower order maps are created and inspected for cluster cohesion manually. Then they are merged into a single SOM which is also inspected manually.

Reported outcome

On the surface it appears that the document clustering was working properly. Similar countries, regions, and islands are mapped to the same nodes and neighborhoods.

Recommendations & Conclusion

Finding a methodology for distributing the computation of a SOM is important. Firstly, allows for reduced training time as portions of the corpus will be examined in parallel resulting in representative weight vectors that will be used in training the final SOM oppose to the entire corpus. A second advantage is if a partition of the corpus is modified then only that part has to be reexamined; the other maps are unaffected by this update. Rauber indicates that the final mapping does not need to be retrained if the degree of change is small however this is incorrect. If this was a supervised learning process, the developed rules would demonstrate some resilience to small degrees of change in the training data. However, unsupervised learning algorithms are very sensitive to these changes. Such changes can cause cluster collapse or explosion; not retraining the final map will result in contaminated clusters.

In essence, the distributed SOM approach is just another means to prototype the corpus. In the previous section, PRIS, prototyping was carried out via an incremental learning algorithm. In this experiment, the prototyping is performed by a SOM. A comparative analysis should be conducted between the two approaches. More testing in general is required in order to evaluate the usefulness of distributing the SOM computation in this

manner. In particular, a more traditional document collection should be examined such as the original Reuters corpus so classification accuracy can be determined and compared with other SOM approaches. As well, experiments using different clustering algorithms, such as complete link and Ward's method, on the partitions should be conducted. The purpose of using a SOM is not only to find the clusters but to determine how similar they are to each other. Since cluster similarity is not used in the creation of the final map, then other clustering algorithms can be used interchangeable.

Corpus reduction is an obvious means for reducing computation time. Further experimentation is required to determine which clustering algorithm will best accomplish this. Distribution of the corpus will allow for faster derivation of these centroids. Using centroids oppose to the entire corpus will allow for faster computation of the SOM. A careful balance has yet to be struck between the degree of corpus reduction and the level of classification accuracy.

5. Conclusion

Document classification is a problem that is not going to go away; repositories on the Web are growing everyday. As they get larger, it becomes increasingly difficult to logical organize them. Traditional approaches to deal with this information overload have been supervised and unsupervised. Supervised learning approaches are faced with the problem of creating training sets that are representative of the corpus; this requires extensive knowledge of the document collection. Unsupervised learning algorithms are desirable because they do not require this knowledge. Their shortcoming is that the ones that perform well have computational complexities that are polynomial.

This paper has presented the SOM, a promising clustering algorithm. Most of the research that was surveyed in this paper was aimed at improve the computational complexity of it. Both dimensionality and corpus reduction have shown that if they are

employed that they will decrease computation time however, not without decreasing classification accuracy too. A balance has to be found among input reduction methods and classification accuracy. However, as illustrated in the work on M-SOM, even if this equilibrium is found, there still remains the problem that the feature maps are not as intuitive to users as theorized previously.

M-SOM has shown that users become lost in cyberspace. This disorientation is a result of users having different perspectives on the data. The feature map, used as a browsing tool, assumes all users to have the same perspective it has. In order for the feature map to be an effective browsing utility, it has to reflect the perceptions and the intuitions of the users. To even attempt to accomplish this, a means for initializing the connection weights in a SOM with intelligent values has to be developed; this variant of the SOM would be a semi-supervised learning algorithm.

It is the opinion of this author that the future of the SOM in document classification is in semi-supervised learning. Supervised and unsupervised approaches are doomed to fail in the future due to the continuous grow of these corpuses. Furthermore, it is rare to have data that is completely unlabeled. The information attached to these labeled entries can not be wasted; there has to be an effective way of incorporating this a priori knowledge. The starting values for the weight vectors are the most obviously part of the algorithm to apply this to. The random initialization of these weights causes the performance of the SOM to be erratic. A method for deriving good starting weights has to be found if SOM based approaches are to be used for documentation classification in production environments.

References

Azcarraga A. and Yap T. Jr. (2001). *SOM-Based Methodology for Building Large Text Archives*. 7th International Conference on Database Systems for Advanced Applications, DASFAA01. pages 66-73. http://citeseer.nj.nec.com/475183.html

Baeza-Yates R. and Ribeiro-Neto B. (1999). Modern Information Retrieval. Addison Wesley

Chen H., Schuffels C. and Orwig R. (1996). *Internet Categorization and Search: A Self- Organizing Approach*. Journal of Visual Communication and Image Representation, Special Issue on Digital Libraries. Volume 7. Number 1. pages 88-102

Deerwester S., Dumais S. T., Furnas G. W., Landauer T. K. and Harshman R. (1990). *Indexing by latent semantic analysis*. Journal of the American Society for Information Science. 41(6). pages 391-407. http://citeseer.nj.nec.com/deerwester90indexing.html

Haykin S. (1999). Neural Networks: A Comprehensive Foundation (2nd Edition). Prentice Hall

Kaski S. (1998). *Dimensionality reduction by random mapping: Fast similarity computation for clustering*. In Proceedings of IJCNN'98, International Joint Conference on Neural Networks,. volume 1. pages 413-418

Kohonen T., Kaski S., Lagus K., Salojärvi J., Honkela J., Paatero V. and Saarela A. (2000). *Self Organization of a Massive Document Collection*. IEEE Transactions on Neural Networks, Special Issue on Neural Networks for Data Mining and Knowledge Discovery, volume 11, number 3, pages 574-585

Rasmussen E. (1992). *Clustering Algorithms*. Information Retrieval: Data Structures & Algorithms. Prentice Hall. pages 419-442

Rauber A. and Merkl D. (1998). *Organization of Distributed Digital Libraries: A Neural Network-Based Approach*. Intelligent Data Engineering and Learning: Perspectives on Financial Engineering and Data Mining. 1st International Symposium, IDEAL'98. pages 283-288. http://citeseer.nj.nec.com/45240.html

Wiener E. D., Pedersen J. O. and Weigend A.S. (1995). *A neural network approach to topic spotting*. Proceedings of SDAIR-95, 4th Annual Symposium on Document Analysis and Information Retrieval. pages 317-332. http://citeseer.nj.nec.com/wiener95neural.html

(1) Yang, Y. (1999). *An evaluation of statistical approaches to text categorization*. Journal of Information Retrieval. volume 1. number 1/2. pages 69-90. http://citeseer.nj.nec.com/article/yang98evaluation.html

(2) Yang Y. and Liu X. (1999) *A re-examination of text categorization methods*. Proceedings of {SIGIR}-99, 22nd {ACM} International Conference on Research and Development in Information Retrieval. pages 42-49. http://citeseer.nj.nec.com/yang99reexamination.html