Adaptive Algorithms for Hypertext Clustering

by Natalija J. Vlajic

A thesis submitted to the faculty of graduate studies in partial fulfilment of the requirements for the degree of

Master of Science

Department of Electrical and Computer Engineering
University of Manitoba

Winnipeg, Canada

July, 1998

0-612-32276-9

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES

*****

COPYRIGHT PERMISSION PAGE

ADAPTIVE ALGORITHMS FOR HYPERTEXT CLUSTERING

BY

NATALIJA J. VLAJIC

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University of Manitoba in partial fulfillment of the requirements of the degree

of

MASTER OF SCIENCE

Abstract

This thesis describes the development of an algorithm for adaptive hypertext clustering (AHC), which is intended to provide accurate and rapid classification of Web documents. The thesis begins with an overview of the traditional techniques for document clustering on the WWW. These techniques mostly, if not exclusively, focus on information delivered in textual form. Moreover, they exploit similar concepts in order to extract the most helpful words, within a collection of document, for clustering purposes. The experimental results obtained show that the traditional techniques require improvements in the term- or word-vector representation of Web pages, especially when applied to Web collections dealing with one or a few particular topics. On the other hand, the results obtained by experimenting with the modified TF/IDF model and the use of word correlation prove that, following term and inter-term statistics, it is possible to extract valuable data on word importance. This knowledge can be used in the compression of irrelevant or less salient words, and lead to a more appropriate and rapid classification of word-vectors. In addition to suboptimal performance in the area of word-vector space creation, the traditional techniques have another significant limitation in common. They usually employ mathematically simple, and for that reason slow, clustering algorithms. For every new set of documents added to a previously classified collection, these algorithms require a complete reclassification. Artificial neural networks (ANNs) based on unsupervised learning have a powerful ability to organize themselves to learn categories of patterns, and then to recognize subsequent patterns in terms of learned categories. From the perspective of document classification and retrieval, this implies that ANNs enable cluster search and thereby reduce search time. Furthermore, they can provide the accommodation of new documents without a complete reclassification as required with some other algorithms. However, a number of results obtained, and presented in this work, show that some ANN algorithms, such as the self-organizing map (SOM) algorithm and hard competitive learning (HCL), produce results dependent on the input data distribution density, and therefore may not be appropriate for document clustering tasks. On the other hand, a modified adaptive resonance theory (ART2) is shown to overcome the main drawbacks of the SOM and HCL, and provide perfectly stable multi-hierarchical clustering. Moreover, ART2 in conjunction with competitive Hebbian learning (CHL) exhibits a very interesting ability to preserve the topology of input data, and enable the retrieval of related or relevant groups of documents. The WWW is a hypertext collection, and a more sophisticated clustering algorithm, intended to provide satisfactory results in the general case of Web page categorization, has to incorporate all available information. The main problem of combined hyper-text clustering is regarding the requirement for a multi-space representation of Web documents. The adaptive hypertext clustering (AHC) algorithm, based upon the modified ART2, is shown to successfully cope with this problem, and depending on the required mode of operation may produce either pure text-based, hyper dimension - based, or combined hypertext clustering.

Acknowledgements

I would like to thank Prof. Howard Card for his support, many helpful suggestions and interesting ideas given throughout my work on this thesis. Thanks to Dean McNeill, Alex McIlraith, Fred Corbett and Imran Khan for being an inspiring research group.

Also, many special thanks to my brother Sasha for all his encouragement and patience.

Contents

CHAPTER 1

CHAPTER 2

CHAPTER 3

List of Figures

Nomenclature

TF/IDF	term frequency / inverse document frequency
IR	information retrieval
ANN	artificial neural networks
SOM	self-organizing map
HCL	hard competitive learning
ART	adaptive resonance theory
CHL	competitive Hebbian learning
AHC	adaptive hypertext clustering

CHAPTER 1

Introduction

1.1 Motivation

According to certain statistics [1.1], the WWW consists of several million sites, its size is doubling every 53 days, and 3000 new sites are added daily. There is no doubt that the Web has already become the largest hypertext and one of the largest digital collections or libraries in the world. There are many situations in which recognizing and grouping similar or related Web pages is an important issue. Examples are search engines, on-line catalogs, and personal bookmarks. In view of the size of the Web, it is easy to understand that any kind of manual classification will eventually be prohibitively time-consuming. Therefore considerable emphasis is currently placed upon rapid, accurate, automatic hypertext clustering algorithms.

Most of the existing techniques for page classification on the World Wide Web are based on text-only analysis. Although the techniques based on text-only analysis give rather good results, it should not be forgotten that the Web is a hypertext collection. "Hypertext is a system that links documents to other related documents on the same machine or across networks"[1.1]. Hypertext documents, beside their textual content, have some additional semantic information embedded in links. Recently, several advanced techniques for hypertext classification have been proposed. These promise improved results for clustering based on combined textual and link information, but mostly when applied to some special cases. Moreover, they often assume the knowledge of some not publicly available data, and therefore can not be employed from a common user's desktop.

It may be observed that nearly all currently used document classification systems on the Web, the traditional and the advanced, have several limitations in common. First, they usually employ a mathematically simple, but for that reason slow, clustering algorithm. It is usually complete link clustering [1.2, 1.3] (more details in section 3.2.2), entirely based on scalar inner products of word vectors (defined in section 2.2). In a complete link clustering process for a collection, each document is initially a cluster by itself. For every pair of pages with their corresponding vectors, the inner product must be computed. The best matching document pairs are successively clustered into progressively larger clusters, and the process continues until clusters reach a sufficient size which has been defined in advance. For every new document added to the collection, its distance from all existing clusters has to be determined before the document can be grouped with the semantically closest cluster. This method is computationally expensive and, since it doesn't involve any type of learning, does not have the ability to discover the real nature of the information space to which it is applied. The second common limitation or weakness is that the word vector dimensionality is very large, even for small collections. For example, in [1.4] each document was presented with a 500-dimensional vector (collection size unspecified), in [1.5] with a 669-dimensional vector (for a collection consisting of 33 documents), and in [1.6] with a 360-dimensional vector (for 240 on-line articles). Any clustering or related computations based upon word-vectors of this size is expected to result in very slow document processing. Third, we are aware of no text-only classification technique which explores correlations between the most salient words in a collection. This knowledge can be very helpful in dimensionality reduction.

1.2 Overview

Chapter 2 describes the purpose and importance of word vector space creation in terms of text-only Web page categorization. The discussion is followed by the results regarding a collection of Web documents related to the subject of neural networks. In particular, section 2.3 describes a conventional method for word vector space creation, pointing out its main disadvantages. It also presents a new technique (modified TF/IDF) demonstrating its advantages over earlier methods. Section 2.4 introduces the concept of word correlation, and explains how it could be useful in order to achieve a rapid and appropriate clustering. In section 2.5 some results obtained applying modified TF/IDF and word correlations to a smaller set of Web documents on various topics are presented.

Chapter 3 presents the most frequently used unsupervised artificial neural network (ANN) learning techniques and explains why they should or should not be used for page classification on the WWW. Section 3.2 gives a brief review of the theory of information clustering and retrieval, and introduces the main requirements that an algorithm has to satisfy in order to provide a satisfactory Web document categorization. Section 3.3 introduces the concept of unsupervised ANN learning. Section 3.4 presents the self-organizing feature map (SOM) algorithm, while section 3.5 presents standard or hard competitive learning (HCL). Experimental results given in each section verify that neither of the two algorithms is appropriate for document clustering tasks. Section 3.6 introduces a new algorithm, which is a variation on adaptive resonance theory (ART). Several examples given in this section demonstrate that the modified ART2 algorithm successfully deals with growing collections of non-uniform document distribution, and produces stable hierarchical clustering. Section 3.7 presents competitive Hebbian learning (CHL), and demonstrates how modified ART2 in conjunction with CHL could be a very sophisticated Web clustering tool.

Chapter 4 introduces a new method for universal hypertext clustering based on modified ART2. In particular, section 4.2 describes the characteristics of the WWW as a hypertextual collection, and presents some of the existing techniques for hypertext clustering. Section 4.3 presents the most important functional hypertext categories encountered on the Web, and introduces twelve hyper dimensions that can provide for a successful discrimination among the functional categories. This section also contains a comparison of the text-only and hyper dimensions - only based clustering for a group of documents on the same topic. Section 4.4 describes the main problems of a combined hyper-text clustering, and explains how these problems could be overcome with the adaptive hypertext clustering (AHC) algorithm introduced in this chapter. The AHC algorithm is applied to three collections of different thematic and functional profiles, and the corresponding results are shown.

Chapter 5 gives a brief presentation of the Java software implementation which was written to perform the experiments.

Chapter 6 summarizes the work, and describes some ideas about related research which appears promising.

References

[1.1] Robert Orfali, Dan Harkey, Jeri Edwards 1996. *The Essential Client/Server Survival Guide.* New York: Wiley Computer Publishing.

[1.2] Gerard Salton, Chris Buckley 1990. *Approaches to Global Text Analysis.*
http://cs-tr.cs.cornell.edu/Dienst/UI/1.0/Display/ncstrl.cornell/TR90-1113

[1.3] C. J. van Rijsbergen 1975. *Information Retrieval.* Boston: Butterworths.

[1.4] Chaomei Chen 1997. *Structuring and Visualising the WWW by Generalised Similarity Analysis.* Proceedings of the Eighth ACM Conference on Hypertext: Hypertext 97, Southampton, UK, 1997.
http://www.brunel.ac.uk/~cssrccc2/papers/ht97.pdf

[1.5] Mehran Sahami, Salim Yusufali, Michelle Q. Wang Baldonado 1997. *Real-time Full-text Clustering of Networked Documents.* In *AAAI-97:* Proceedings of the Fourteenth National Conference on Artificial Intelligence, p. 845, Menlo Park, CA: AAAI Press.
http://robotics.stanford.edu/users/sahami/papers-dir/sonia-abst.ps

[1.6] Mehran Sahami, Marti Hearst, Eric Saund 1996. *Applying the Multiple Cause Mixture Model to Text Categorization.* In ICML-96: Proceedings of the Thirteenth International Conference on Machine Learning, pp. 435-443, San Francisco, CA: Morgan Kaufmann.
http://robotics.stanford.edu/users/sahami/papers-dir/ml96-mcmm.ps

CHAPTER 2

Word-Vector Space Creation using Modified TF/IDF and Word Correlation

2.1 Introduction

Traditional techniques for page classification on the World Wide Web are based on text only analysis, and they often provide satisfactory if suboptimal results. It is observed that most of these methods require improvements in terms of word-vector representation of Web pages, especially when applied to Web collections dealing with one or a few particular topics. This chapter introduces a modification of a well known rule for information retrieval, and the utilization of word correlation. The algorithm was employed in clustering 218 Web pages related to the subject of neural networks. The results were useful in arriving at an efficient word- or term-vector representation, in order to achieve a rapid and appropriate clustering based on content of on-line documents. The term vectors derived using this algorithm were classified using a modified adaptive resonance theory (ART) algorithm, an unsupervised learning method in artificial neural networks which is proven to provide very accurate and sophisticated clustering (Chapter 3). Examples of the results are presented in the chapter, suggesting several benefits of employing the methods.

2.2 Background

Most of the existing techniques for automatic classification of on-line documents are based on independent work by Salton [2.1,2.2] and Rijsbergen [2.3]. Although Salton's work is oriented toward information clustering and Rijsbergen's toward information retrieval, their basic ideas are very similar. Both assume that the content or meaning of a page is uniquely defined by its textual or word content. "Text words are content identifying units" [2.1]. Weighting the words according to how often they appear in a document, and how often they appear in the overall collection, it is possible to obtain a vector representation of the text. This provides one representation of the content or meaning of the document. Also, the similarity between two pages can be determined by comparing their corresponding word-vectors.

One can usefully regard the WWW as an information space. Accordingly, we can imagine Web pages or hypertext documents as points or vectors in that space, where each vector dimension corresponds to a chosen word. The amplitudes associated with the various vector dimensions indicate the presence or frequency of the corresponding words. Since the WWW contains documents on various topics, the number of distinguishable words appearing in all existing Web pages is indefinite, which means that the WWW is a space of effectively infinite dimensionality. However, if one focuses on Web documents related to a specific topic such as biology, music, or in our case neural networks, the set of words appearing in these documents is less heterogeneous. For example, it is not common to find words such as cell, molecule, or DNA outside of biology related pages. This means that Web documents dealing with one or a few particular topics form a subspace in the global indefinite information space, of a finite dimensionality. The dimensionality is defined by the number of distinguishable words appearing within the relevant documents. Every collection of such documents embodies sub-topics, and the main task for clustering techniques is to identify those subsets. A clustering algorithm is intended to form a number of smaller clusters such that the documents within a cluster are more similar, or closer according to a particular distance metric, than the documents in neighboring clusters.

Although Web documents dealing with one particular topic form a finite subspace in the global information space, clustering within such a collection may be a difficult task. Ambiguity of some documents often results in a number of partially overlapping clusters, and implies that some words (dimensions) appear in more than one sub-group. Therefore, sub-group identification should not be based on the presence of individual words only, but should incorporate inter-word statistics. The following sections present some methods that provide improved word-vector space creation and more efficient clustering within a set of Web documents on the same topic.

2.3 Word-Vector Space Creation

2.3.1 Zipf's Law

Let us assume that a given collection is spatially bounded inside the global Web information space, which means it contains a finite number of distinguishable words. An accurate and computationally efficient clustering would depend upon mapping this document collection onto a smaller number of dimensions or words. These words would be those most effective in discriminating among documents. The most commonly used approach to this problem is to simply gather all words appearing in a set of documents, and to exclude all so-called stop words which are not useful in discrimination because they are too common in the English language. After suffix stripping or stemming, which helps to detect equivalent stems (words with the same root), a prescribed number of the most frequently occurring terms is used for creating the word vector space. This is a direct implementation of Zipf's Law. More details appear in [2.3]. We have applied this standard approach to a collection of 218 Web pages dealing with the subject of neural networks in its various aspects. A total number of 7000 words or stems was distinguished, and according to the Zipf's law, the most highly ranked terms were:

neural, network, learn, system, pag(e), comput(e), hom(e), model,
inform, applic, function, fuzzi, group, algorithm, new, previou, ...

The URLs of the 218 Web pages employed in the collection is given in Appendix 1. The list of stop words employed was from [2.3] (Appendix 2), and stemming was done according to [2.4] (Appendix 3).

It is observed that words such as *neural* and *network* are the primary representatives of the group in this simple approach, but they obviously don't have any discrimination power within this collection, since they are expected to occur in every document. One possible modification of this implementation would be to eliminate a number of the most highly ranked terms. But how many of these should be eliminated? Since document classification is intended to be automatic, that number must be precisely defined in advance. Based upon our experience in dealing with this problem, it is impossible to provide a unique solution for an arbitrary collection. The number of words that should be removed depends both upon the size of the collection and upon the diversity of its documents.

2.3.2 Modified TF/IDF Model

Since the results obtained using Zipf's law were not very satisfactory, another approach which was a modification of the well known term-frequency / inverse document-frequency model, was tested on the same collection [2.2, 2.3]. This model was originally created to improve term weighting for the purpose of effective information retrieval. The standard form of this method, which was employed in our work for word vector calculation of individual documents, is represented by

$$w_{ik} = tf_{ik} \times \log\left(\frac{N}{n_k}\right) \tag{2.1}$$

where w_{ik} is the weight of term T_k in document D_i and tf_{ik} is the frequency of occurrence of term T_k in document D_i , N is the number of documents in the collection, and n_k the number of documents containing T_k . In order to find the most salient words within the entire collection, we have replaced the standard form with

$$w_k = ntf_k \times \log\left(\frac{N}{n_k}\right) \tag{2.2}$$

where w_k is the weight of term T_k to the collection and ntf_k is the normalized frequency of occurrence of term T_k in the collection. In this approach the model takes on a different meaning. According to (2.2), words which appear sufficiently often in only a few documents, meaning that they might be representatives of sub-groups within the main collection, are expected to be highly ranked. At the same time, words which appear either very rarely or too often in member documents are scored low. Figure 2.1 illustrates the dependence of word discrimination power on document frequency and normalized term frequency within a collection of 100 documents.

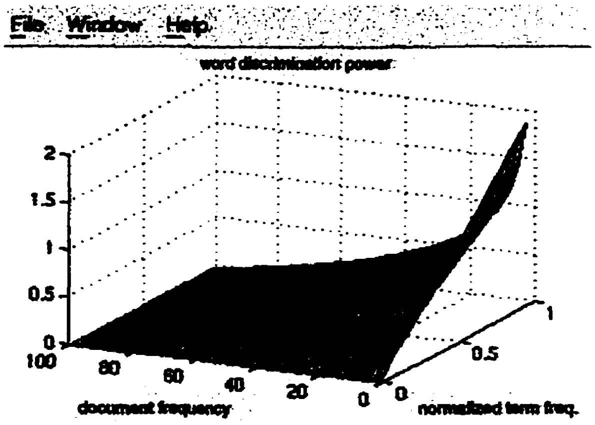


Figure 2.1 Word discrimination power within a collection of 100 documents

The most highly ranked terms or stems in this case turned out to be

volum(e), fuzzi, learn, competi(tive), group, net, neuron, hom(e), previou(s), product, reinforc(ement), neural, pag(e), siren, content,...

It is observed that by randomly employing a number of words from the top of this list without human observation, it is possible to automatically form a word vector space of sufficient discrimination power for a sensible clustering. It is also obvious that some words such as *Siren*, and *IEEE*, although not common (often institution names), had high rankings. This was the result of a relatively small collection size, so their normalized importance was not diminished. Often those words had appeared in short documents, so that their normalized term frequencies tf_k were large, or they had appeared in a single document, so that their inverse document frequencies $\log(N/n_k)$ were large. Regarding normalization, we have attempted to avoid favoritism based on document length. Words from longer documents tend to exhibit larger tf_{ik} in Eqn (2.1) which implies larger overall term frequency tf_k . Each tf_{ik} was in our method divided by the total number of distinguishable words or stems found in the same document. The normalized tf_{ik} were summed to calculate the total tf_k

$$ntf_k = \sum_{i=1}^{N} \frac{tf_{ik}}{ndw_i} \tag{2.3}$$

where ndw_i is the number of distinguishable words found in document D_i . Combining a number of the most highly ranked terms from this new list based on the modified TF/IDF model, an initial 192-dimensional word vector space was created. This list is available in Appendix 4.

Figure 2.2 shows the clustering obtained within the initial word vector space. As it can be observed, the requested number of clusters was 40. Analysing the formed clusters and the corresponding documents, it was apparent that the network recognized the main thematic subcategories. However, in a number of cases it failed to place documents on the same or related topics to one mutual group, primarily because these documents were based on different vocabulary. Thereby, for example, even though the documents from the groups *book* and *handbook* were thematically very close, they remained within separate cluster because they had different words dominating.

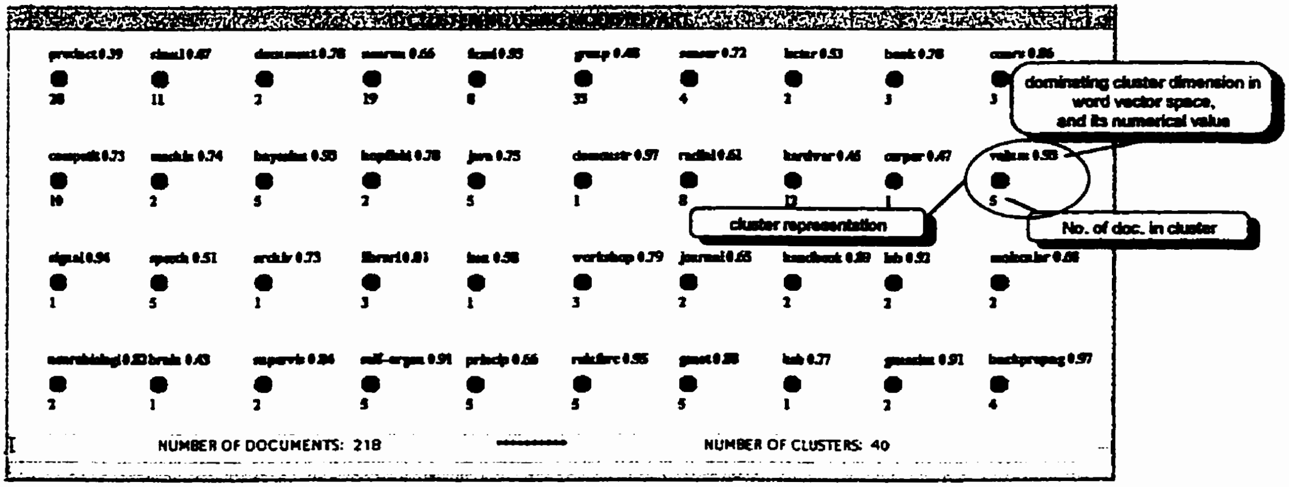


Figure 2.2 Document clustering in 192-dimensional space

2.4 Word Correlations

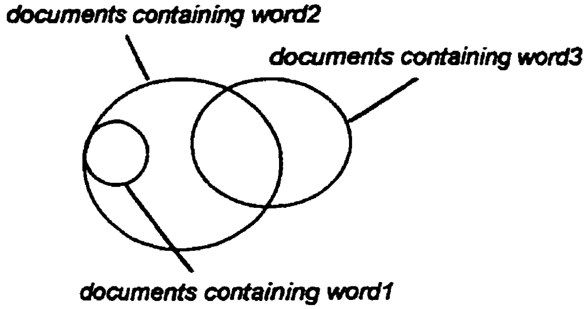
It is a common classification problem to work in a space, containing a set of vectors to be clustered, whose dimensionality considerably exceeds the number of expected groups within the given set. In such cases a few dimensions are likely to appear in a similar manner, while they rarely co-occur with some other dimensions. More specifically, there exist strong positive and negative correlations among the vector components or dimensions, and the unification or compression of strongly correlated dimensions is a reasonable basis for dimensionality reduction. This approach is commonly employed in multivariate statistical methods such as principal components analysis [2.5]. For on-line document classification, this assumption has a deeper meaning. Correlations and anticorrelations among words or dimensions suggest semantically close or distant concepts, forming a language corpus for a particular sub-topic. Let us assume that corr_{ij} is the correlation between words T_i and T_j within the entire collection, and corr_{kij} is the correlation between T_i and T_j within document D_k. If

$$\mathrm{corr}_{kij} = \begin{cases} 0, & \text{when } T_i \text{ doesn't appear in } D_k \\ -1, & \text{when } T_i \text{ appears and } T_j \text{ doesn't appear in } D_k \\ +1, & \text{when } T_i \text{ appears and } T_j \text{ appears in } D_k \end{cases} \tag{2.4}$$

and

$$\mathrm{corr}_{ij} = \frac{\sum_{k=1}^{N} \mathrm{corr}_{kij}}{n_i} \tag{2.5}$$

where n_i is the number of documents containing T_i, then corr_{ij} ($\forall i,j$) ranges between -1 and +1. If T_i usually appears when T_j appears, corr_{ij} is a positive real number close to +1. On the other hand, if T_i seldom appears when T_j appears, corr_{ij} is a negative real number close to -1. Finally $\mathrm{corr}_{ij} \neq \mathrm{corr}_{ji}$ which implies a general antisymmetry. This final feature is of special importance, since the correlations between two words T_i and T_j in both directions automatically describe the relation between the group of documents containing T_i (G_i) and the group containing T_j (G_j). For example, if $\mathrm{corr}_{ij} \approx 1$ and $\mathrm{corr}_{ji} \approx -1$, then G_i is certainly a small subset of G_j, while for $\mathrm{corr}_{ij} \approx -1$ and $\mathrm{corr}_{ji} \approx -1$ G_i and G_j have almost no common elements. On the other hand $\mathrm{corr}_{ii} = 1$ ($\forall i$) implies an autocorrelation or word self-correlation.


documents containing word2
documents containing word3
documents containing word1

$corr_{12}=1,\ -1<corr_{21}<0,\ corr_{13}=-1,\ corr_{31}=-1$
$-1<corr_{23}<0,\ corr_{32}\approx 0$

Figure 2.3 Relations between groups of documents containing particular words

The implementation of (2.4) and (2.5) on the most salient 192 words from our collection of 218 neural network Web pages verified the existence of strong inter-term correlations, but it also indicated the necessity for some preprocessing. In particular, results such as

applet → java	*1.0*
stock → market	*0.846*
dendrit(e) → neuron	*0.777*
VLSI → hardwar(e)	*0.636*
professor → univers(ity)	*0.555*
institut(e) → research	*0.485*

where the number to the right is the correlation, could have been expected since words such as stock and market are conceptually close, and often occur together in everyday conversations. From a different point of view, the strong correlation

radial → function	*1.0*
speech → recognit(ion)	*0.846*
genet(ic) → algorithm	*0.739*
Heb(bian) → rul(e)	*0.600*
Markov → model	*0.571*
princip(al) → compon(ent)	*0.428*

despite the semantic distance between the words in everyday language was particularly encouraging, since, for example, radial basis functions are well known in neural network terminology. Specifically, this meant that positive word correlation could detect common domain-specific phrases. Other results such as

self-organ(izing) → som	*- 1.0*	*som → self-organ(izing)*	*- 1.0*
book → handbook	*- 1.0*	*handbook → book*	*- 1.0*
email → e-mail	*- 0.851*	*e-mail → email*	*- 0.833*
model → prototyp(e)	*- 0.894*	*prototyp(e) → model*	*- 0.090*
tutori(al) → lectur(e)	*- 0.894*	*lectur(e) → tutori(al)*	*- 0.733*

were opposite to the normal meaning of anticorrelation, although this also might have been anticipated. Anticorrelation, as mentioned earlier, may imply that two words do not appear together because they are semantically distant. However, in this case, the words model and prototype, for instance, did not appear together for a different reason, because they were almost synonymous.

A method to avoid those computationally correct, but conceptually unreasonable, cases was to form groups of identical, similar, or conceptually related terms, called thesauruses. Following thesaurus creation, the 192-dimensional word-vector space was compressed into a 125-dimensional space. This aspect of compression implies that certain dimensions in the new term-vector space corresponded, not to words, but to terms or concepts captured by sets of related words. Note that this step of dimensionality reduction can be automatic provided an on-line set of thesauruses was available. This compressed list of 125 thesauruses is available in Appendix 5.

Using the modified expressions (2.4) and (2.5)

$$\mathrm{corr}_{kij} = \begin{cases} 0, \text{ if no } T_m \in \text{thesaurus } TH_i \text{ appears in } D_k \\ -1, \text{ if any } T_m \in \text{thesaurus } TH_i \text{ appears and no } T_n \in TH_j \text{ appears in } D_k \\ +1, \text{ if any } T_m \in \text{thesaurus } TH_i \text{ and any } T_n \in TH_j \text{ appears in } D_k \end{cases} \qquad (2.6)$$

$$\mathrm{corr}_{ij} = \frac{\sum_{k=1}^{N} \mathrm{corr}_{kij}}{n_i} \qquad (2.7)$$

where n_i is the number of documents containing any T_m from TH_i , new results were obtained. Note that this is the same equation as (2.5) but the meaning of the terms are different. According to these results, if a group of words TH_i was strongly positively correlated to a number of other groups it was eliminated. This was done in such a way that it was added to those TH_j in proportion to its correlation factor. Thus, in calculating the term frequency tf for thesaurus TH_j, all of its original words contribute with their full tf_j , while those from TH_i contribute with the product $corr_{ij} x\ tf_i$.

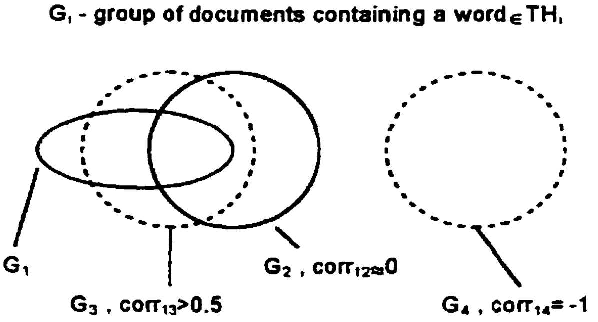
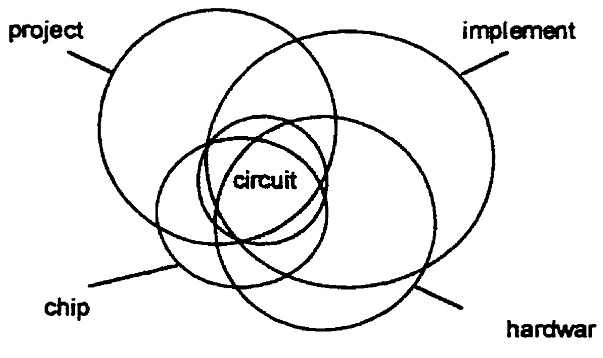


Figure 2.4 Thesaurus correlation as a precondition for thesaurus elimination

Figure 2.4 explains why correlations ≥0.5 or ≤ -0.5 were considered as being of special importance. For this case, if a document D_n belongs to G_1 (i.e. contains a word from TH_1), and $corr_{12}\approx 0$, it means the probability that D_n belongs to G_2 is only ½. In general, the correlations around zero value don't offer any relevant information. On the other hand, $corr_{13}\geq 0.5$ indicates that D_n belongs to G_3 with a probability in the range from 0.75 to 1, while $corr_{14}=-1$ indicates that D_n is definitely not an element of G_4. This means that all strong correlations (either positive or negative) provide useful information.

Only groups of words with at least one correlation greater than 0.5, excluding autocorrelation, were considered for compression. It made sense to insist on such a strict criterion for the following reasons. First, according to figure 2, the value of 0.5 considerably reduces the expected error probability, in working with the reduced word set. Second, if a thesaurus (TH_i) doesn't have significant positive correlations, then the other groups of words are above a minimum semantic distance. It was shown that elimination of a TH_i which did not have at least one corrleation greater than 0.5 caused the disappearance, or unsuitable replacement, of the corresponding concept in the word-vector space.



circuit → chip 0.571	chip → circuit 0.0
circuit → hardwar 0.571	hardwar → circuit -0.56
cirucit → project 0.714	project → circuit -0.688
cirucit → implement 0.857	implement → circuit -0.633

Figure 2.5 Experimental result

Figure 2.5 is based on real experimental data. The stem *circuit* (i.e. the group of words it represented) had a few strong positive correlations, and it was expected that the elimination of this thesaurus would not result in a conceptual loss. The clustering results obtained verified that words *chip*, *hardwar*(e), *project* and *implement* were able to semantically compensate for its removal. On the other hand, a single modest positive correlation between *hardwar*(e) and *implement* (0.28) was not used in the same manner, since it could have caused considerable conceptual errors. There were 14 eliminated words

scienc(e), registr(ation), schedul(e), analog, devic(e), manufactur(e), servic(e), proceed, report, circuit, processor, run, stock, busi(ness)

within the 125-dimensional word-vector space, so this aspect of dimensionality reduction resulted in a 111-dimensional term-vector space. This is a significant improvement in view of the well known curse of dimensionality in multivariate statistics [2.5]. This list of 111 thesauruses is available in Appendix 5.

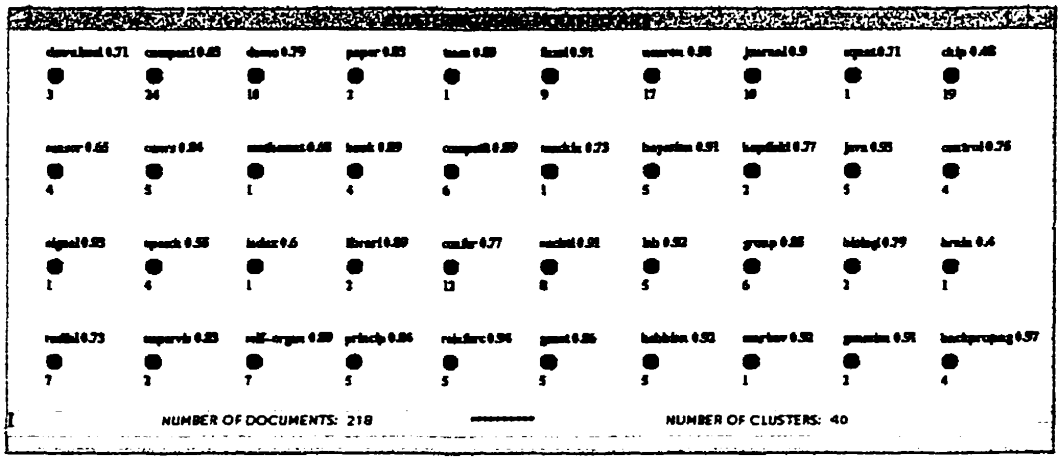


Figure 2.6 Document clustering in 111-dimensional space

The clustering obtained within the 111-dimensional word vector space is shown in Figure 2.6. A comparative analysis of the clusters formed within the initial and final word vector space, using the same clustering algorithm, has shown the following:

Although based on an identical set of words, organized in a different manner, the initial and final term-vector space did not result in identical clusters. When compared to human performance of Web document classification, the final word vector space provided very accurate results. Moreover, when tested on a set of new documents that had not been included in the process of word vector space creation, the neural network based on the final word vector space performed very satisfactory classification. On the other hand, it was observed that an optimal categorization was impossible within the initial word vector space, mainly because strongly related words were treated as completely independent or uncorrelated dimensions. An example is the clusters *volum*(e), *issu*(e), and *journal* in Figure 2.2. Even though Web documents from these clusters were thematically close, they did not contain the same words, or their meaning was expressed through some other non-word data type. Therefore they ended up in separate clusters. However, when represented and clustered in a word vector space that recognized inter-term relationships, these documents became members of exactly one group *journal* (Figure 2.6). The other very obvious example is the groups *lectur*(e), and *cours*(e) in Figure 2.2. Both groups contained documents on NN courses, but their meaning was based on the domination of different terms. However, the two groups merged into one cluster *cours*(e) for the case in Figure 2.6. The lists of all clusters from Figure 2.2 and Figure 2.6 with their Web page sources are available in Appendix 6 and Appendix 7. In addition to the improvement regarding more sophisticated clustering, dimensionality reduction based on term and inter-term statistics offers another significant advantage:

The clustering conducted in the initial 192-dimensional word vector space required nearly 40 percent more time than the clustering in the final 111-dimensional space, although they were performed for the same set of documents and for the same number of output nodes (clusters). In general, it is observed that lower dimensionality substantially reduces computation time, which may alternatively result in considerable improvements in clustering a particular collection of documents in a given time, particularly when applied to large collections of several thousand documents.

2.5 Discussion of Modified IF/IDF and Word Correlation

The methods presented in the previous sections were tested on a number of Web collections of various profiles and sizes. In all the cases the modified TF/IDF model provided better identification of the most salient words when compared to the results obtained by Zipf's law. Dimensionality reduction based on term and inter-term statistics was extremely useful for Web collections dealing with one or a few particular topics, but did not have the same effectiveness for smaller collections with many different (mutually exclusive) topics involved. An illustration is regarding the collection presented in Table 2.1. The corresponding URLs are given in Appendix 8.

topic (category)	number of documents
tennis	6
volleyball	4
accordion	4
jazz	5
philosophy	3
neural networks	2
java	1

Table 2.1

Words with the highest discrimination power according to the modified TF/IDF were:

tenni, philosophi, icon, volleybal, jazz, java, accord, neural, ball, music, document, tabl, network ....

Employing 25 words from the top of this list, without human intervention, it was possible to form a word vector space of sufficient discriminatory power. In particular, the clustering obtained in the given space using modified ART2 (more details in section 3.6.3) was completely satisfactory. It was observed that in this case the information on inter-term statistics could not provide any further improvements in clustering.

2.6 Conclusions

Traditional techniques for on-line document clustering, founded on independent work by Salton and by Rijsbergen, may not always provide satisfactory performance. The results obtained by experimenting with the modified TF/IDF model and the use of word correlation have proved that, by considering term and inter-term statistics within a collection, it is possible to extract valuable data on word importance. This knowledge can be used in the compression of words that are irrelevant or less salient, particularly for clustering tasks. It has been shown that a clustering performed in the compressed word vector space significantly better emulates the way in which humans organize information. Moreover, word compression leads to subsequent analysis of the corresponding group of documents in a lower dimensional term-vector space. Lower dimensionality implies substantially reduced computation, either for simple clustering algorithms or for advanced learning methods. It has been proved that many of these tasks grow exponentially with dimensionality. Therefore, word vector space creation based on combined TF/IDF model and word correlation can ensure an accurate and rapid clustering based on content of Web documents.

References

[2.1] Gerard Salton, Chris Buckley 1990. *Approaches to Global Text Analysis.*
http://cs-tr.cs.cornell.edu/Dienst/UI/1.0/Display/ncstrl.cornell/TR90-1113

[2.2] Gerard Salton, Chris Buckley 1987. *Term Weighting Approaches in Automatic Text Retrieval.*
http://cs-tr.cs.cornell.edu/Dienst/UI/1.0/Display/ncstrl.cornell/TR87-881

[2.3] C. J. van Rijsbergen 1975. *Information Retrieval.* Boston: Butterworths.

[2.4] William B. Frakes, Ricardo Baeza-Yates 1992. *Information Retrieval, Data Structures & Algorithms*. Englewood Cliffs, New Jersey: Parentice Hall.

[2.5] Christopher M. Bishop 1995. *Neural Networks for Pattern Recognition*. Oxford: Clarendon Press.

CHAPTER 3

Neural Networks for Document Clustering

3.1 Introduction

In our research we are investigating the relative advantages and disadvantages of unsupervised artificial neural network (ANN) learning algorithms based upon winner-take-all hard competitive learning algorithm, the self-organizing map (SOM) algorithm, and the adaptive resonance theory (ART), for the purpose of an efficient document classification and retrieval. The following sections provide theoretical background on each of the algorithms and present some of the results we have obtained when with these methods. A new algorithm, which is a variation on adaptive resonance theory, and the corresponding results, is also introduced. This algorithm is demonstrated to overcome the main limitations of standard ART, and can successfully deal with growing collections of non-uniform document distribution. Finally, a method that provides topology preservation (competitive Hebbian learning) is introduced, and some results regarding the utilization of the modified ART algorithm with CHL are shown.

Most of the experiments presented in this chapter are related to the Web document collection from Table 2.1 . However, some results are also shown regarding its approximate two-dimensional projection. The purpose of these experiments was primarily to provide a better understanding of the algorithms and their functionality, by making the final results observable since they are only two-dimensional.

3.2 Background

3.2.1 Information Retrieval

As indicated in Cpahter 1, due to the incredible increase in the amount of electronically available information of the past few years, efficient methods of *information retrieval* (IR) have recently been given a great deal of attention. Generally speaking, information retrieval is as a common name for a number of different methods which are intended to provide accurate and rapid access to information of interest. When applied to search engines, on-line catalogs, and personal bookmarks, the word *information* could be simply replaced by *document*. Accordingly, in the case of document retrieval the information of interest is the subset of documents which are relevant to a query - search request .

While document retrieval focuses on finding a particular set of documents in a database, *document browsing* provides a general knowledge of what type of documents the database contains. Therefore document browsing is considered to be a supplement to conventional IR by allowing the reader to discover retrieval cues that successively can be used for further query formulation.

Conventional document-retrieval techniques are mainly based on *serial search*, which means that a given query has to be matched with each document in the collection in order to find the relevant ones. Although *serial search* may provide adequate results in some cases, it is acknowledged to be extremely slow. This especially applies to real-time information systems which are expected to simultaneously provide service to many users.

Sophisticated and efficient document-retrieval systems are based on *cluster search strategies*. Instead of conducting a search through the entire collection, these systems first classify documents into subject areas, then confine the search to certain groups only. In other words, they use the following overall strategy:

* Clusters of related documents are constructed by comparing their identifiers (word vectors).
* Each document cluster is represented by a special word vector, known as the *cluster centroid*.
* A given query is compared against the centroids of all document groups, and only documents located in clusters with sufficiently high query-centroid similarities are retrieved.

Undoubtedly, reducing the number of required comparisons between documents and queries, *cluster search* provides enhanced retrieval. Moreover, systems that employ this strategy are very convenient for browsing operations, since each cluster centroid may be considered as a compact representation of a number of documents. The outline of a clustered document collection is given in Figure 3.1.

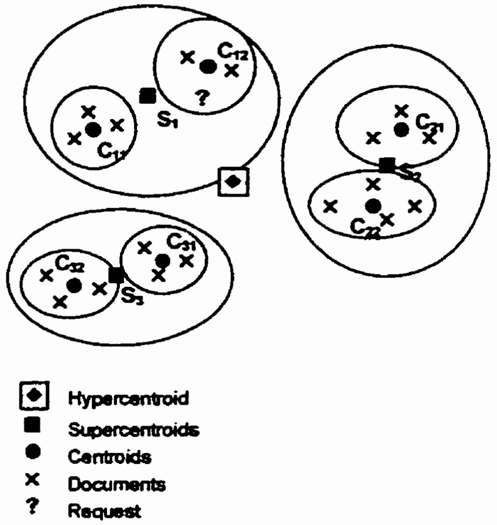


Figure 3.1 Clustered document collection

According to Figure 3.1, a collection may have several types of centroids: *hypercentroid* - which represents the center of the complete collection, *supercentroids* - which represent the next level of granularity, *centroids* - which represent regular clusters. Within such a hierarchically organized system, a search for relevant documents is conducted by comparing the query first against the highest-level centroids. Then, only for those higher-level centroids that are shown to be sufficiently similar to the query, the search is continued.

The following figure shows a search tree for the clustered collection of Figure 3.1 .

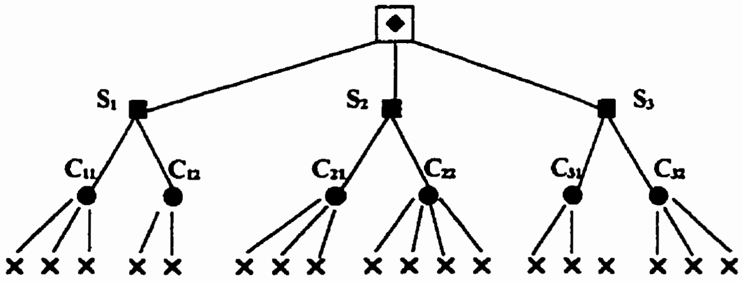


Figure 3.2 Search tree for clustered collection of Figure 3.1

Multi-level cluster search, based on a hierarchical cluster structure as presented in Figure 3.2, provides significant improvements in retrieval efficiency. As a simple illustration, the following table shows the number of comparisons that should be performed in order to find the best match to the given request from Figure 3.1 (the two documents within cluster C_{12}), using different search techniques.

search technique	number of comparisons
serial search	18
two-level cluster search (based on centroids, and documents)	6
three-level cluster search (based on supercentroids, centroids, and documents)	5

Table 3.1 Comparisons of some search techniques

Although hierarchical document clustering evidently reduces *search time*, and increases the efficiency of document retrieval, enhanced efficiency does not necessarily provide improved accuracy. Retrieval accuracy, also known as *precision* and *recall*, entirely depends on the nature of the clustering algorithm applied. If the algorithm cannot simulate the human organization of information, precision and recall will be compromised.

According to Rijsbergen [3.1], clustering algorithms for information classification and retrieval should satisfy three conditions:

1) The algorithm should produce a clustering which is unlikely to be altered drastically when further objects are incorporated, i.e. it is stable under growth
2) The method is also stable in the sense that small errors in the description of the objects lead to small changes in the clustering
3) The resultant clustering is independent of the initial ordering of the objects.

"Any cluster method which does not satisfy these conditions is unlikely to produce any meaningful experimental results." [3.1]

3.2.2 Single-Link Clustering and Complete-Link Clustering

In document classification *single-link clustering* and *complete-link clustering* are frequently used techniques. Both algorithms are hierarchical, and exploit the following principle: initially each document is a cluster by itself, then the two most similar items are combined into a bigger cluster. The clustering process continues recursively until only a single cluster remains. The only difference between these algorithms is the criterion which is applied when two documents, or clusters, are to merge into a larger cluster. In single-link clustering the similarity between two clusters is defined as the similarity between the most similar pair of items from the two clusters. On the other hand, in complete-link clustering the cluster similarity is the similarity between the least similar pair of items.

The main disadvantage of single-link and complete-link clustering is that, whenever two clusters are considered for merging, all their pairwise similarities have to be known in order to find the most (or least) similar pair of items from the two clusters. Therefore, cluster centroids are not exploited throughout the process of clustering. Put another way, no learning is involved. Furthermore, for a new set of documents added to a previously classified collection, these algorithms require a complete reclassification. That may be very computationally expensive and inefficient, especially for large collections, or collections that are frequently changing.

3.3 Learning in Unsupervised Artificial Neural Networks

Artificial neural networks can be viewed as an extension of conventional techniques in statistical pattern recognition. [3.2] However, the feature that significantly distinguishes ANNs form the other techniques is the constraint that pattern-information processing be carried out in a self-consistent manner, in a network of elemental processors - artificial neurons. With the nervous system analogy as the main inspiration, ANNs have received and exploited a number of useful principles inherited from neurobiology and psychology. However, from the point of view of statistical pattern recognition, the most important property of ANNs is their powerful ability to organize themselves to learn categories of patterns, and then to recognize subsequent patterns in terms of learned categories. There are two basic types of neural network learning: supervised and unsupervised.

In *supervised learning* for each input pattern the value of the desired output is specified. The desired output is defined as the optimum action to be performed by the neural network. The network parameters are adjusted under the combined influence of the input vector and the error signal. The error signal is the difference between the actual response of the network and the desired response.

In *unsupervised learning* the use of target data (desired output) is not involved. Instead, the goal is to discover the distribution of the ensemble of input patterns. In other words, if the mechanism that generates the patterns also segregates them into clusters in a meaningful manner, then by using unsupervised learning the location and distribution of these clusters can be identified. Moreover, neural networks based on unsupervised learning (self-organization) have the ability to employ statistical knowledge about the input data to classify new 'previously unseen' patterns. This ability is called generalization, and is at the root of all of the most effective information agents.

From the perspective of ANNs, document classification, which ideally should be performed without any human intervention (target data), is a standard unsupervised learning problem. On the other hand, from the perspective of document clustering and retrieval, ANNs based on self-organization are appealing for several reasons.

First, providing cluster identification, ANNs can undoubtedly enable cluster search and thereby reduce search time. Furthermore, cluster identification is the key to efficient browsing. Finally, the generalization in terms of a previously clustered collection means tha new documents can be automatically added and classified, without a complete reclassification as required with single and complete link clustering. All this suggests that the utilization of unsupervised ANN learning for document clustering tasks can offer a number of advantages.

Among the most widespread unsupervised ANN learning methods used are standard (hard) competitive learning, the self-organizing map (SOM) algorithm, and adaptive resonance theory (ART). Although quite different in the rules they employ and in the results they provide, these three algorithms are based on the same concept: for a new input vector the output neurons of the network compete among themselves with the result that only one output neuron is activated for any one input vector.[3.3] This concept, called *winner-take-all*, enables the individual neurons of the network to specialize on sets of similar patterns, and thereby to become feature detectors.

3.3.1 Winner-Take-All Competition

One way to build the winner-take-all concept into a neural network is to include lateral connections among the neurons, as shown in Figure 3.3. Through the lateral connections each node performs lateral inhibition, or, in other words, tends to inhibit the neurons to which it is connected.

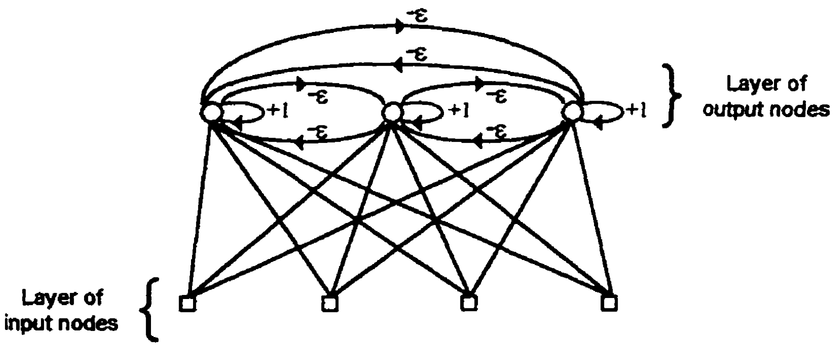


Figure 3.3 Network formed on 'winner-take-all' principle

Let $w_{kj}(t)$ denote the lateral connection weight from output node k to output node j at time t. Then,

$$w_{kj} = \begin{cases} 1 & \text{for } k = j \\ -\varepsilon & \text{for } k \neq j, \quad \varepsilon < \dfrac{1}{M} \end{cases} \tag{3.1}$$

Based on (3.1), the output $y_k(t)$ of neuron k at time t+1 is given by

$$y_k(t+1) = f\left(i_k(t) - \varepsilon\sum_{j\neq k} y_j(t) + y_k(t)\right) \tag{3.2}$$

where $i_k(t)$ is the total external control exerted on neuron k by the weighted effect of the input signals.

$$i_k(t) = \sum_{l=1}^{N} w_{kl}x_l(t) \tag{3.3}$$

w_{kl} denotes the weight connecting input node l to output node k. N is the number of input nodes - the input data dimensionality.

The function *f* in (3.2) is defined as

$$\begin{aligned} f(\mu) &= \mu \quad \text{for } \mu > 0 \\ f(\mu) &= 0 \quad \text{for } \mu < 0 \end{aligned} \tag{3.4}$$

From (3.3) it is evident that the output $y_k(t)$ is inhibited by all the other outputs. Therefore, for neuron k to be the winning node its $i_k(t)$ for a specified input pattern x(t) must be the largest among all the neurons in the network.

3.4 The Self-Organizing Map Algorithm

Results regarding the utilization of the SOM algorithm for the purpose of improved document clustering and retrieval are reported in [3.4]. According to the authors the SOM has proven to be a useful tool in organizing a collection of documents, and in facilitating browsing and searching of related items. However, the results we have obtained experimenting with the SOM, presented in section 3.4.2, showed that the algorithm produces results that are very dependent on the distribution of input data, and thereby fails to satisfy Rijsbergen's first criterion. Accordingly, it is unable to provide optimal document retrieval (search), but nevertheless exhibits some interesting feature in terms of document browsing.

3.4.1 An Overview of the SOM Algorithm

The self-organizing feature map algorithm was developed by Kohonen (1982), and the main inspiration for its creation came from the essential features of computational maps in the human brain. The principal aim of this algorithm is to transform high-dimensional input data onto a one- or two-dimensional discrete map, in such a way that neighborhood relations among the input vectors are preserved as far as possible. Higher-dimensional maps are also obtainable but are not common. The discrete map itself, in fact, is a two dimensional array of neurons (nodes) that are fully connected to the input nodes. Each node has an associated *reference vector* w_c, which presents a vector in input space R^n. The number of nodes and the type of the array have to be defined in advance, before any learning occurs. The lattice type of the array can be rectangular (the most common case) as illustrated in Figure 3.4, hexagonal, circular, or even irregular.

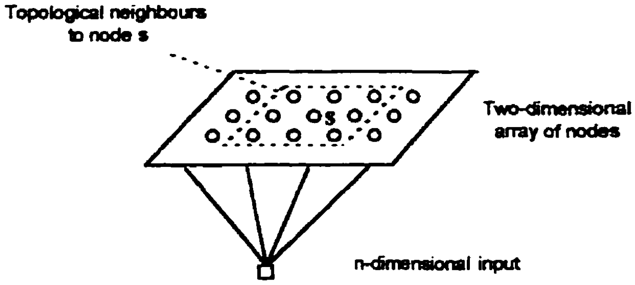

Topological neighbours to node s
s
Two-dimensional array of nodes
n-dimensional input

Figure 3.4 Two-dimensional lattice of neurons

The self-organizing map learning procedure is summarized as follows:

1. Initialize the weight vectors $w_{c_i}(0)$ ($c_i \in C$, C - set of output nodes) randomly according to p(x).

2. Choose an input signal x according to the input distribution p(x).

3. Determine the winner s such that

$$\| w_s(n) - x \| \leq \| w_c(n) - x \| , \quad \forall c \in N \tag{3.5}$$

4. Adjust each weight vector according to

$$w_c(n+1) = w_c(n) + \Delta w_c(n) \tag{3.6}$$

where

$$\Delta w_c(n) = \eta(n)\, h_{cs}(x-w_c) \tag{3.7}$$

$\eta(n)$ is a learning-rate parameter, and h_{cs} is a decaying neighbourhood function. They are defined as follows:

$$\eta(n) = \eta_i (\eta_f / \eta_i)^{n/nmax} \tag{3.8}$$

η_i and η_f are suitable initial and final values of the learning rate, n_{max} is the total number of iterations.

$$h_{cs}(n) = \exp - d(c,s)^2/2\sigma(n)^2 \tag{3.9}$$

d(c,s) is the distance on the grid between the winning node s and an arbitrary node c.

$$\sigma(n) = \sigma_i \, (\sigma_f / \sigma_i)^{n/nmax} \tag{3.10}$$

σ_i and σ_f are suitable initial and final values of the standard deviation of the Gaussian.

5. If $n < n_{max}$ continue with step 2.

The most important properties of the SOM algorithm are:

* *Topology preservation*

This property implies that input vectors which are close in R^n should be mapped onto neighboring nodes in the lattice. Or, in other words, neighbouring nodes in the lattice should have similar input vectors mapped onto them. This is a direct consequence of the learning process, which forces the weights of the winning node and, at the same time, its topologically closest neighbours to move toward the input vector x. Topology preservation leads to a number of practical application for the SOM algorithm, particularly the projection of data into a two-dimensional space for visualisation purposes. However, in many cases when data is of a complex distribution and comes from a high-dimensional space (without a natural 2-D subspace of data) this feature is not achievable. In this case the SOM algorithm provides suboptimal placement of the reference vectors.

* *Density matching*

Density matching implies that the map can reflect variations in the distribution of input data. Therefore, regions in the input space where the data shows high distribution density are mapped onto larger domains of the map (larger regions of neurons).

3.4.2 Experimental Results

The experiments presented in this section were conducted using the Matlab Neural Network Toolbox [3.5]. In each case the network was trained for 10000 iterations, although it was observed that for the collection sizes we were dealing with, learning stabilization was obtained after less then 5000 iterations. Also, in all cases network contained 12 nodes, arranged as a 4 x 3 grid. Some experiments with a reduced number of neurons failed to provide sensible results. The learning rate η was 0.1. The neighbourhood function was chosen to be rectangular, so the learning affected only the winning node and its eight nearest topological neighbours.

Experiments with two-dimensional data set

The first set of experiments was related to a two-dimensional set containing twenty-five data points. This set was an approximate projection of the twenty-five 25-dimensional word vectors, calculated for the Web collection as given in Table 2.1.

Figure 3.5 a) shows the two-dimensional collection. Each cluster is denoted with the name of the corresponding cluster from the original word-vector space. The corresponding SOM is presented in Figure 3.5 b). The red dots indicate the neurons and the positions of their reference vectors, whereas the blue lines connect neighbouring nodes. Both figures were produced using Matlab.

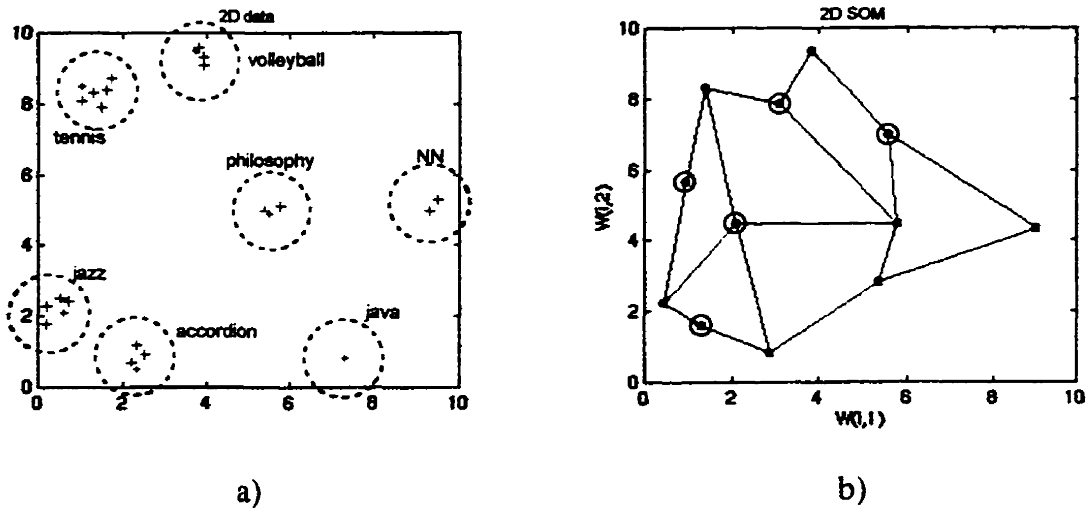


Figure 3.5 Two-dimensional data set and the corresponding SOM

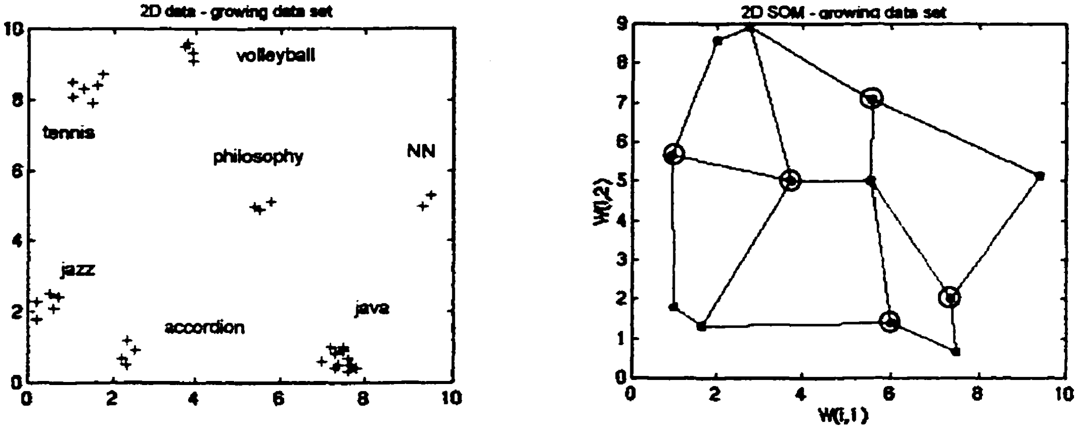
From Figure 3.5 the following can be observed:

- The shape of the map accurately matched the data distribution - most of the nodes were placed within or close to the regions with significant data distribution.
- A number of nodes correctly discovered the centres of the clusters, except for the cluster *java*. This was a logical consequence of the density matching feature of the SOM algorithm, since this cluster contained only one data point.
- Five nodes (in black circles) were positioned in the regions where the data distribution was zero, and these nodes were not the winning nodes for any of the input data points. (In the literature these are called *dead units.*) This was a consequence of the learning process which forced neighbourhood-based adjustment, and therefore placed the idle nodes according to the position of their neighbours that were often winning nodes.

In order to get an insight into the behavior of the SOM algorithm for the cases of growing data sets, and data sets with nonstationary statistics, the following two experiments were performed.

In the first case, ten new elements (points) were added to the cluster *java* of the original two-dimensional collection, while the distribution within the other clusters remained intact, as presented in Figure 3.6 a). The corresponding SOM is shown in Figure 3.6 b).

It may be easily observed that the growth of one cluster resulted in significantly altered positions of most of the reference vectors. The centre of the cluster *java* was correctly discovered this time. In addition, two more nodes were pulled toward this cluster, because of its increased distribution density. On the other hand, the centres of some other clusters (*tennis*, *volleyball*, *jazz*, *accordion*) were not properly identified. The SOM also contained five dead units.

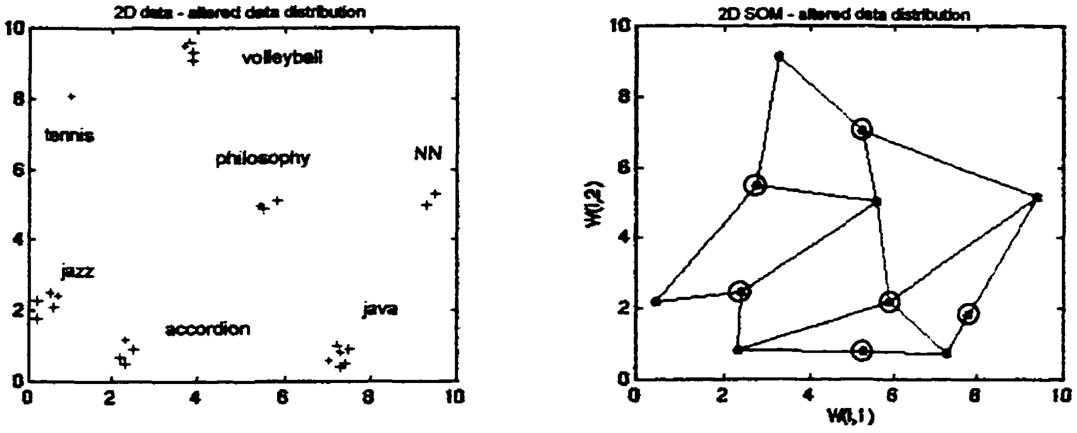


a) b)

Figure 3.6 Two-dimensional growing data set and the corresponding SOM

In the second case, the distributions densities of the clusters *tennis* and *java* were interchanged, compared to the initial distribution. As a result, the cluster *tennis* now contained only one element, while the cluster *java* contained six elements, as shown in Figure 3.7 a). The corresponding SOM is presented in Figure 3.7 b).

Again, the positions of the reference vectors were significantly changed compared to the initial case, but also as compared to the previous case. The algorithm failed to discover the cluster *tennis* this time, while the other clusters and their centres were properly identified. Six out of twelve nodes were dead units.



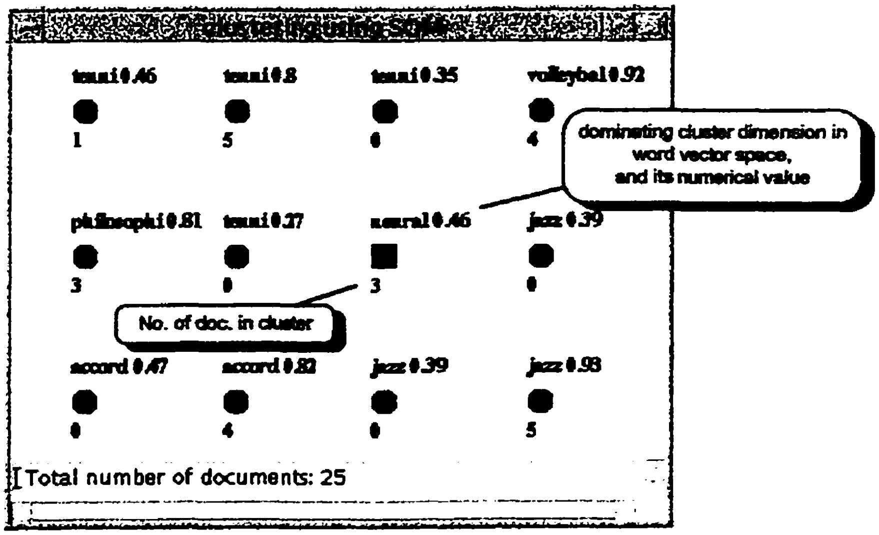
a)

b)

Figure 3.7 Two-dimensional collection of altered data distributions and the corresponding SOM

Experiments with original word-vector collection

The second experiment was performed for the original set of twenty-five word vectors, in the corresponding 25-dimensional word vector space. This time first a SOM was trained using the Matlab Neural Network Toolbox, and then the calculated reference vectors were employed by our software in order to evaluate the obtained document clustering. The clustering results, presented in Figure 3.8 and Table 3.2., exhibit very high consistency with the 2D results.



Total number of documents: 25

Figure 3.8 Document clustering using the SOM algorithm

actual clusters

output nodes

Doc. per cluster	tennis	volleyball	jazz	accordion	philosophy	NN	java
tenni 0.46	1	0	0	0	0	0	0
tenni 0.80	5	0	0	0	0	0	0
tenni 0.35	0	0	0	0	0	0	0
volleybal 0.92	0	4	0	0	0	0	0
philosophi 0.81	0	0	0	0	3	0	0
tenni 0.27	0	0	0	0	0	0	0
neural 0.46	0	0	0	0	0	2	1
jazz 0.39	0	0	0	0	0	0	0
accord 0.47	0	0	0	0	0	0	0
accord 0.82	0	0	0	4	0	0	0
jazz 0.39	0	0	0	0	0	0	0
jazz 0.93	0	0	5	0	0	0	0

Table 3.2 Document distribution per output node for the SOM of Figure 3.8

Thematically close groups (tennis and volleyball, jazz and accordion) were placed next to each other. While four out of twelve reference vectors were positioned within or close to the group on tennis, which had the highest distribution density, none of them was placed close to the group on java, which contained a single document. In other words, the presence of the document on java was not very obvious, since it was placed in the same group with the two documents on neural networks. The network contained five dead units.

Conclusion

Based on these experiments, the SOM algorithm evidently provides results that are dependent on the input data density. In other words, relatively small changes in the distribution of training data may result in the discovery of clusters of significantly altered sizes and positions. This is in conflict with Rijsbergen's first criterion, and therefore the self-organizing map algorithm should probably not be used as a document clustering algorithm for the purpose of efficient information search and retrieval.

However, the algorithm may be used for *explorative search* or browsing. In that case its main advantages and disadvantages are:

Advantages

1) Since the same or nearby map nodes are in general the winning nodes for similar input vectors, it means that a two-dimensional SOM map could reflect the semantics of documents.
2) Due to the dependence of the results on the statistics of the training data, high reference vector density within certain areas of n-dimensional word vector space is an indicator of high overall document density within the same areas. Similarly, zero

reference vector density implies zero or very low document density. Therefore, the SOM algorithm can help in obtaining an insight into the main topics that a collection is dealing with, and, approximately, how the documents are distributed among them.

Disadvantages

1) The number of nodes in the network has to be quite large, and certainly must exceed the number of existing clusters in the collection in order to obtain any sensible results. Since the size of the network is proportional to the time required for training (every input vector has to be checked against each reference vector in order to find the best match), any increase in the number of nodes considerably slows down the learning process, particularly when applied to large collections of several thousand documents.
2) Due to the nature of the SOM algorithm, most of the reference vectors are pulled toward the regions of high document density, leaving clusters with a few documents without any reference vectors allocated. In other words, the algorithm fails to indicate the presence of these groups.
3) Depending on the complexity of the data distribution the algorithm may produce a relatively large number of dead units (in our examples 40 to 50 percent of the nodes were not the winning nodes for any of the input vectors). In general the reference vectors of these nodes are placed in regions with zero data density, and falsely imply the presence of documents in those regions.

3.5 Standard or Hard Competitive Learning

3.5.1 An Overview of Hard Competitive Learning

In contrast to the SOM algorithm, which enforces on the adaptation of the winner and its topological neighbours in order to provide topology preservation and density matching, hard competitive learning adapts the winning node only. Accordingly, the main goal of hard competitive learning is not to map, but to classify input patterns into mutually exclusive recognition categories.

There are two main types of hard competitive learning: *batch* and *on-line*. In the *batch* type of learning all possible input signals are evaluated first before any adaptations are made. On the other hand, in *on-line* hard competitive learning adaptations are performed directly after each input pattern is presented to the network. In many situations, particularly when the data set is extremely large, batch learning is impractical. Therefore, on-line type learning is more frequently used in practice.

The on-line hard competitive learning procedure is summarized as follows:

1. Initialize the weight vectors $w_{c_i}(0)$ ($c_i \in C$, C - set of output nodes) randomly according to p(x).

2. Choose an input signal x according to the input distribution p(x).

3. Determine the winner s such that

$$\| w_s(n) - x \| < \| w_c(n) - x \| , \qquad \forall c \in N \tag{3.11}$$

4. Adjust the reference vector of the winner according to

$$w_c(n+1) = w_c(n) + \eta(n)\,(x - w_c) \tag{3.12}$$

5. If $n < n_{max}$ continue with step 2.

The main goal of competitive learning is the minimization of the expected quantization error. In the case of a finite input data set D the error is defined as

$$E(D, C) = \frac{1}{|D|} \sum_{c \in C} \sum_{x \in R_c} \| x - w_c \|^2 \tag{3.13}$$

where R_c is the Voronoi set of the neuron c. The definition of Voronoi regions is given in Appendix 9.

It has been shown that if the input patterns form a small number of clusters as compared to the number of output nodes, then learning eventually stabilizes and provides the best distribution of the reference vectors consistent with the distribution of the input data [3.6].

However, it has also been shown that for an arbitrary distribution of input data learning may never stabilize, and the response of the network to the same input pattern can differ on each following presentation of that input pattern. Such unstable learning is mainly caused by the nature of competitive learning, which enables prior learning to be washed away by more recent learning. In addition, it has been observed [3.7] that competitive learning may produce different results for different initializations of reference vectors. The standard initialization, as presented in step 2 of the hard competitive learning procedure, follows the distribution density of input data p(x), in order to give each reference vector the same chance to be the winner for a randomly-generated input signal x. This may lead to rather suboptimal results when the distribution is highly non-uniform.

3.5.2 Experimental Results

The experiments presented in this section were conducted again using the Matlab Neural Network Toolbox [3.5]. In each case the network was trained for 2000 iterations, and the learning rate η was 0.1.

Experiments with two-dimensional data set

The results obtained using hard competitive learning for the initial two-dimensional collection is presented in Figure 3.9. The network contained seven output nodes, which corresponded to the number of actual clusters in the data set. The green dots indicate the nodes and the positions of their reference vectors, whereas the black lines indicate the corresponding Voronoi regions.

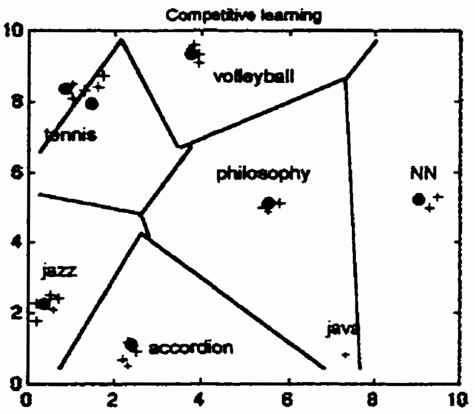
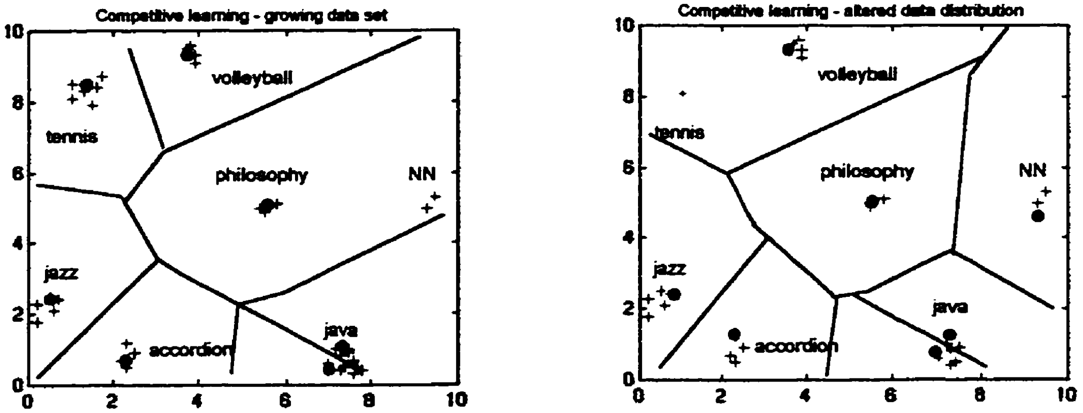


Figure 3.9 Clustering within 2D collection using hard competitive learning

From Figure 3.9 the following may be observed:

1) A number of neurons correctly discovered the centres of the existing clusters, except for the cluster *java*. This was the consequence of both the initialization, that was based on the input data distribution density, and the learning process, which was unconditionally plastic. Therefore, none of the initial reference vectors were likely to be positioned within or close to the cluster *java* due to its low data density. For the same reason the learning based on the single point from this cluster was washed away by data points from the neighbouring cluster, which appeared more frequently. Consequently, the data points from the clusters *philosophy* and *java*, although very distant, had the same winning node, and belonged to the same Voronoi region.
2) Two reference vectors were placed within the cluster *tennis*, due to its high distribution density. Therefore, the documents from this group ended up in two different Voronoi regions.

The following two figures depict the results obtained using hard competitive learning for the growing data set, and the set with altered data distribution, analogous to the cases presented in Figure 3.6 and Figure 3.7 respectively.



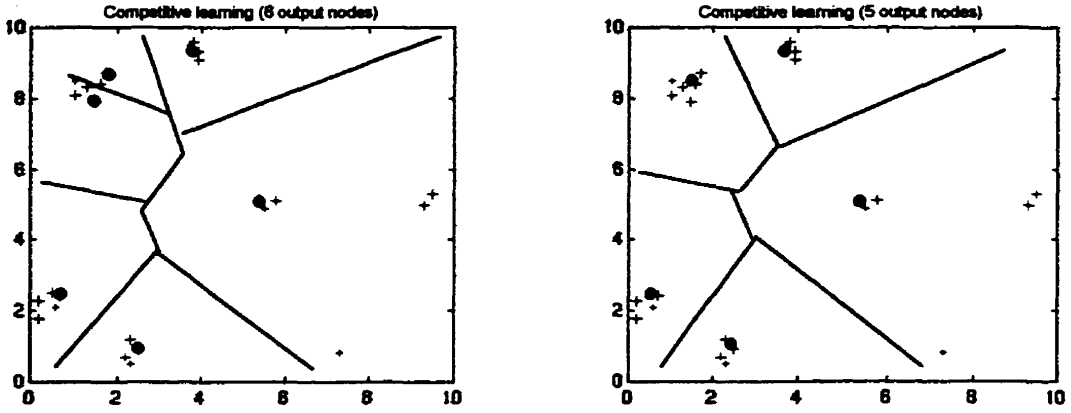
a) b)

Figure 3.10 Clustering based on hard competitive learning
for the growing data set and for the set with altered data distribution

It may be easily observed from Figure 3.10 that the growth of the cluster *java* caused significantly altered positions to a number of reference vectors. Therefore, the group *NN*, with had the lowest distribution density in this case, ended up without any reference vector allocated. On the other hand, two reference vectors were positioned within the group *java*, since now this was the group with the highest distribution density.

The interchange of distribution densities between the clusters *tennis* and *java*, as presented in Figure 3.13, again resulted in an altered clustering. This time none of the reference vectors was allocated to the cluster *tennis*.

In order to test the performance of hard competitive learning for the cases when the number of existing clusters is not known in advance, two experiments were conducted with reduced numbers of output nodes. The corresponding results are presented in Figure 3.11. In both cases the algorithm failed to discover the real nature of the data set: exactly one reference vector was allocated to three completely distinct clusters, while the high distribution density of the remaining four clusters captured all the other reference vectors.

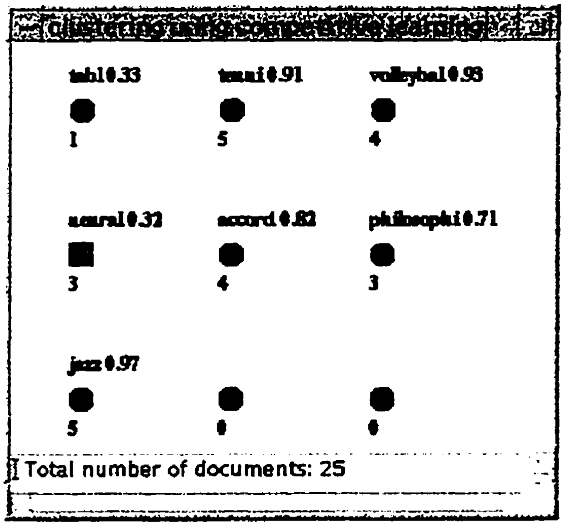


a) b)

Figure 3.11 Clustering using hard competitive learning for six and for five output nodes

Experiments with original word-vector collection

Figure 3.12 and Table 3.2 present the clustering obtained for the original 25-dimensional set of word vectors. As observed, similar to the 2D case, the clusters formed approximately matched the actual clusters. The mismatching was a consequence of the nonuniform document distribution within the collection. Thus, the large document density of the group on *tennis* resulted in the formation of two separate clusters. On the other hand, due to low distribution density, documents from the groups *neural networks* and *java* were placed in the same cluster, although thematically distinct.



tabl 0.33 ● 1	tenni 0.91 ● 5	volleybal 0.93 ● 4
neural 0.32 ■ 3	accord 0.82 ● 4	philosophi 0.71 ● 3
jazz 0.97 ● 5	● 0	● 0

Total number of documents: 25

Figure 3.12 Document clustering using hard competitive learning

actual clusters

output nodes

Doc. per cluster	tennis	volleyball	jazz	accordion	philosophy	NN	java
table 0.33	1	0	0	0	0	0	0
tenni 0.91	5	0	0	0	0	0	0
volleybal 0.93	0	4	0	0	0	0	0
neural 0.32	0	0	0	0	0	2	1
accord 0.82	0	0	0	4	0	0	0
philosophi 0.71	0	0	0	0	3	0	0
jazz 0.97	0	0	5	0	0	0	0

Table 3.3 Document distribution per output node for the network of Figure 3.12

Conclusion

Based on these experiments it is evident that hard competitive learning is unstable under growth or changes in the statistics of the input data, and produces clustering that is very dependent on the distribution of training data. Accordingly, it fails to satisfy Rijsbergen's first criterion, and for that reason should not be employed as a document clustering algorithm for the purpose of improved information search and retrieval.

However, in a similar way to the SOM algorithm, hard competitive learning could be used for explorative search or browsing. Due to its dependence on the statistics of input data it can provide an insight into the most represented topics within a collection, and into how the documents have been distributed among these topics. On the other hand, in contrast to the SOM algorithm, hard competitive learning does not provide topology preservation or spatial relationships among the reference vectors, i.e. the corresponding Voronoi regions are not obvious. In other words, there is no clear indication if documents that have two or more different winning nodes are in fact elements of the same group.

3.6 Modified ART2

3.6.1 Introduction

The experiments presented in section 3.4 and 3.5 have shown that the SOM algorithm and hard competitive learning are not appropriate methods for document clustering tasks. Neural networks based on these techniques tend to produce suboptimal results, dependent on the input data density in such a way that small changes in the distribution of training data may result in clusters of altered sizes and positions. In fact, both the SOM algorithm and hard competitive learning enable prior learning to be eliminated by more recent experience, which differs dramatically from the way in which human brains learn and organize information. A multitude of examples have shown that humans are designed to successfully adapt to environments whose rules may change, without necessarily forgetting the old knowledge. [3.8]

The learning instability experienced by the SOM algorithm and by hard competitive learning mainly occurs as a result of their unconditional plasticity, or adaptability. In particular, for a new input pattern presented to a network based on one of these algorithms, the corresponding winning node and its topological neighbours, or the winning node alone, are moved toward the input pattern, without verifying how distant they actually are. In other words, the network will attempt to learn the new pattern and encode it into one of the existing categories, even though it might significantly differ from all previous patterns and run the risk that the learned knowledge may be washed away. This drawback of the SOM algorithm and hard competitive learning addresses an issue related to learning algorithms in general: How should a system be designed in order to protect previously learned knowledge from being eliminated by new learning, while enabling new learning to be automatically incorporated into the total knowledge of the system in a self-consistent way. This problem is known as *stability-plasticity dilemma*.

Adaptive resonance theory (ART), introduced in 1976 by Grossberg, is an unsupervised learning technique partially based on the winner-take-all concept, but also influenced by the processes underlying logical human reasoning. A significant property of this algorithm is its ability to switch between stable and plastic modes, thereby overcoming the stability-plasticity dilemma. In particular, it is capable of plasticity in order to learn significant new events, yet can remain stable against irrelevant events.

There are two main adaptive resonance theory models: ART1 and ART2. While the ART1 model is capable of stably learning binary input patterns, the ART2 model shows the same behaviour for analog patterns. Since the word vectors that we were dealing with had continuous-valued features, the ART2 model was of primary interest.

The main difference between the ART2 learning model and the other two algorithms based on the winner takes all concept is related to the adjustment of the winning node for each new training pattern. In contrast to the SOM algorithm and to hard competitive learning, the ART models initiate the adjustment of the winning node only if it deems this node to be an acceptable match. In other words, a category modifies its previous learning only if the input vector is sufficiently similar to risk a further refinement of its profile. Otherwise, if no available category provides a good enough match, a new node is selected for learning a novel recognition category. However, if no adequate match exists and the full capacity of the system has also been exhausted, the network cannot accommodate the new input and learning is automatically inhibited. This mechanism defends a fully committed memory capacity against eradication by new significantly different input patterns.

When applied to document classification, the above mentioned features imply that ART2 is able to accommodate to nonstationary data distributions, and to adaptively recognize a varying number of different groups or knowledge domains. A new training item (document) either alters a previous group, if shown to be sufficiently thematically close (within a specified vigilance parameter) or it begins to represent a new group. All the other groups remain intact. This is reminiscent of how humans cope with the same dilemma.

Although ART2 evidently overcomes the main drawbacks of the SOM algorithm and of hard competitive learning, it is not an ideal document clustering algorithm. The following two features are its principal disadvantages. First, ART2 requires a fixed number of output nodes, i.e. a predefined number of clusters. If the number of clusters that corresponds to a given vigilance parameter is greater than the predefined number of output nodes, the network is incapable to learn all the categories. Therefore, clusters that appear after the full capacity of the network has been exhausted remain unaccomodated or rejected. Second, experimenting with ART2 we have observed that for low values of vigilance parameter, which should provide fewer categories, learning becomes unstable, particularly if the data set consists of a number of overlapping clusters.

In order to preserve the conceptual advantages of ART2 , while at the same time overcoming the above mentioned problems, we have proposed certain modifications to the original algorithm. The results obtained show that our modified version of ART2 significantly improves on earlier methods of document clustering. In addition, the modified ART2 can be easily adapted to the specific requirements of hypertextual collections and provides for even more sophisticated Web page clustering, as shown in Chapter 3.

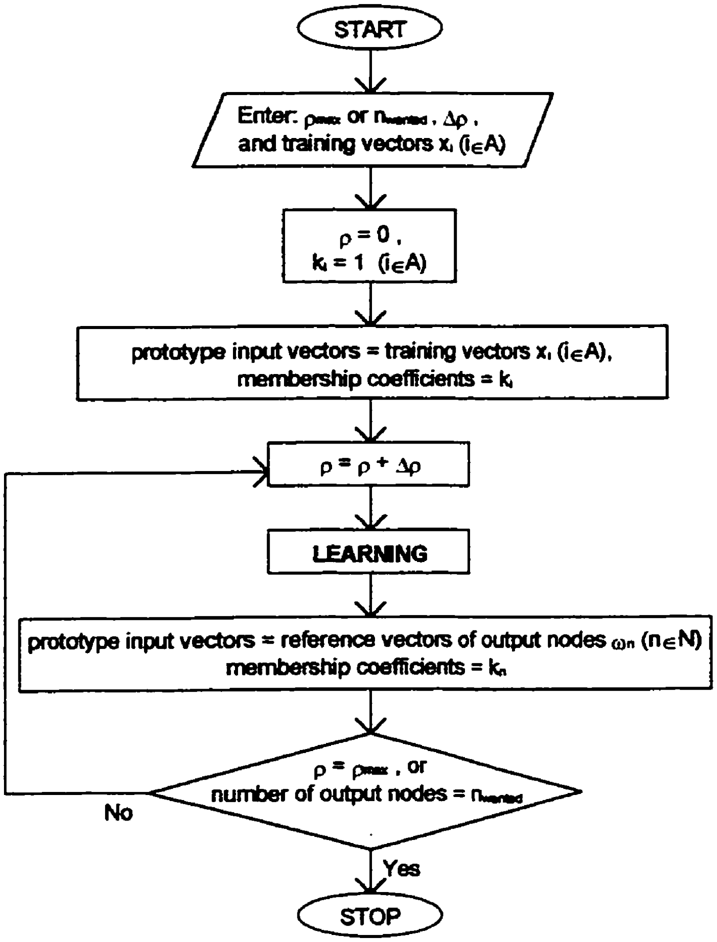
3.6.2 An Overview of the Modified ART2 Algorithm

Our modified version of ART2 for document clustering is based on the following concepts:

Each learning epoch is determined by a specific value of a *tolerance parameter* (ρ), which is a dynamic generalization of the inverse of the standard (static) vigilance parameter, and by a set of *prototype input vectors*. In the first epoch prototypes are actual training vectors, whereas for all other cases they become centers of previously discovered clusters. The tolerance parameter is a criterion for vector association, which implies that two or more prototype input vectors will merge into a larger cluster only if the level of their mismatch is less than ρ. The initial tolerance parameter is set to zero and, during learning, gradually increases in steps $\Delta\rho$ (the dynamic parameter) until it reaches a maximum value ρ_{max} at which point learning terminates. Learning can also terminate when the number of clusters discovered reaches some predetermined value n_{max}.

The parameters ρ_{max} and n_{max} are inversely related, since small ρ_{max} means that only modest differences are tolerated between training vectors and the system learns to classify input patterns into a large number of finely-divided categories. On the other hand, large

ρ_{max} enables the system to tolerate large mismatches, and thus to group together training cases into fewer categories. At any stage of learning every prototype input vector has an associated *membership coefficient*, which indicates the number of training vectors that have merged into the corresponding cluster. Initially all membership coefficients are unity, since each input vector is equal to one training vector and represents a one-member cluster. Figure 3.13 illustrates our modified ART2 algorithm.


START
Enter: ρ_{max} or n_{wanted}, $\Delta\rho$, and training vectors x_i ($i \in A$)
$\rho = 0$, $k_i = 1$ ($i \in A$)
prototype input vectors = training vectors x_i ($i \in A$), membership coefficients = k_i
$\rho = \rho + \Delta\rho$
LEARNING
prototype input vectors = reference vectors of output nodes ω_n ($n \in N$) membership coefficients = k_n
$\rho = \rho_{max}$, or number of output nodes = n_{wanted}
No
Yes
STOP

n_{wanted} - requested number of output nodes (clusters), ρ_{max} - requested inverse vigilance (tolerance) parameter, $\Delta\rho$ - dynamic parameter, A - set of training vectors, N - set of output nodes

Figure 3.13 Modified ART2 algorithm

LEARNING is the principal routine of every epoch, and is described as follows:

STEP 1: Insert one output node. Initialize its reference vector to the first prototype input vector ($w_1 = \xi_1$), and set its membership coefficient to the corresponding value of ξ_1.

STEP 2: Chose a new prototype input vector ξ.

STEP 3: Determine the best matching output node s - the unit with the nearest reference vector:

$$\| w_s - \xi \| < \| w_n - \xi \| , \ \forall n \in N \tag{3.14}$$

where N is the set of output nodes.

STEP 4: Verify that ξ belongs to the s^{th} cluster (the cluster determined by w_s and ρ) if

$$\| w_s - \xi \| < \rho \tag{3.15}$$

If so proceed to step 5. Otherwise go to step 6.

STEP 5: Adjust the reference vector of s according to:

$$w_s = (k_s/(k_\xi + k_s)) \, w_s + (k_\xi/(k_\xi + k_s)) \, \xi \tag{3.16}$$

Note that ω_s becomes the arithmetic mean of all (old and new) vectors belonging to the s^{th} cluster. Adjust the membership coefficient of s:

$$k_s = k_s + k_\xi \tag{3.17}$$

Go to step 1.

STEP 6: Since ξ does not belong to s, which was the most probable, insert a new output node. Set its reference vector to ξ and its membership coefficient to k_ξ .

Go to step 2.

The modified ART2 algorithm possesses some important properties, as follows:

Property 1. ***Stable learning***

If $\Delta\rho$ is sufficiently small, such that it provides clustering independent of the initial ordering of input vectors, the gradually increasing tolerance parameter ρ can ensure absolutely stable learning. In particular, any two input vectors that had the same winning node for a lower level of the tolerance parameter will have a mutual winning node for all higher levels of ρ, i.e. later in the learning. This implies that the algorithm produces hierarchical results, which can be represented by a tree structure or dendogram. Thus, at each level of the hierarchy or value of ρ, a set of clusters can be identified, and as moving up the hierarchy the clusters at the lower levels are nested in the clusters at the higher level.

This property makes the modified ART2 particularly convenient for information classification and retrieval tasks, as described in section 3.2.1.

Property 2. ***Mean squared error minimization within every cluster formed***

The algorithm provides the minimization of mean squared error within each cluster formed by placing the corresponding center (reference vector) at the position of the arithmetic mean of all vectors belonging to that cluster.

PROOF: Let us assume that ξ_i , i=1,..,M , is the set of vectors belonging to a developed cluster, and w_c is the center of the cluster. The corresponding mean squared error is given by the expression:

$$Q = \sum_{i=1}^{M} \left\| \xi_i - w_c \right\| = \sum_{i=1}^{M} \sum_{j=1}^{N} (\xi_{ij} - w_{c_j})^2 \qquad (3.18)$$

where, in general, ξ_i , $w_c \in R^n$. The value of ω_c that minimizes Q has to satisfy the condition:

$$\frac{\partial Q}{\partial w_{c_k}} = 2 \sum_{i=1}^{M} (\xi_{ik} - w_{c_k}) = 0 \qquad (3.19)$$

for every k=1,..,N. This further implies that

$$\sum_{i=1}^{M} \xi_{ik} - M w_{c_k} = 0 \qquad (3.20)$$

Finally,

$$w_{c_k} = \frac{1}{M} \sum_{i=1}^{M} \xi_{ik} , \qquad \forall k = 1, .., N \qquad (3.21)$$

Property 3. ***The mean squared error minimization is preserved at every level of clustering tolerance***

The merging of two clusters based on (3.16) ensures that the center of each newly formed cluster is positioned at the arithmetic mean of all vectors belonging to both initial clusters. Thereby the algorithm preserves the minimization of mean square error within every cluster and at every level of clustering tolerance.

PROOF: Let us assume that there are two clusters ξ_i , i=1,..,M and ξ_j , i=1,..,N, formed on the principle of mean squared error minimization, that should merge into a new larger cluster. w_M , k_M=M and w_N , k_N=N are the corresponding centres and membership coefficients respectively. The centre of the new cluster, according to the principle of mean square error minimization (3.21), should be

$$w_c = \frac{1}{N+M}\sum_{i=1}^{N}\xi_i \tag{3.22}$$

This expression can be rearranged into

$$w_c = \frac{k_N}{k_N + k_M}\cdot\frac{1}{N}\sum_{j=1}^{N}\xi_j + \frac{k_M}{k_N + k_M}\cdot\frac{1}{M}\sum_{i=1}^{M}\xi_i \tag{3.23}$$

which further implies

$$w_c = \frac{k_N}{k_N + k_M}w_N + \frac{k_M}{k_N + k_M}w_M \tag{3.24}$$

The last equation verifies that the centre of the new cluster can be calculated using only the centres (prototype input vectors) and the corresponding membership coefficients of the initial clusters, without knowing all of their individual vectors.

In addition to mean squared error minimization this property provides another significant advantage: it considerably reduces computation time, particularly when clusters that should merge consist of a large number of input vectors.

3.6.3 Experimental Results

The experimental results presented in this section were obtained using software developed turing this thesis. In all cases the dynamic parameter (Δρ) was set to 0.001. By experimenting with a number of different cases this value was initially determined to provide clustering that does not depend on the initial ordering of training vectors.

Experiments with two-dimensional data set

The results obtained using modified ART2 for the initial two-dimensional collection is presented in Figure 3.14. It may be observed that all the clusters and the corresponding centres were correctly discovered. This implies that, in contrast to the SOM algorithm and to hard competitive learning, the modified ART2 algorithm successfully coped with the problem of a nonuniform data distribution.

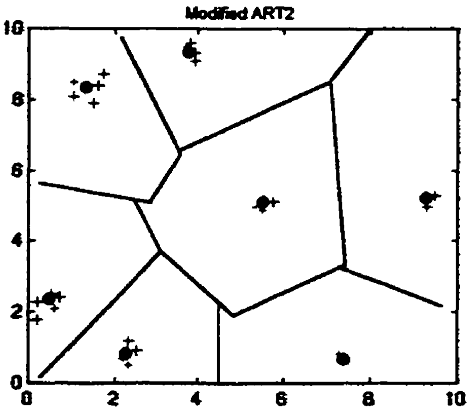
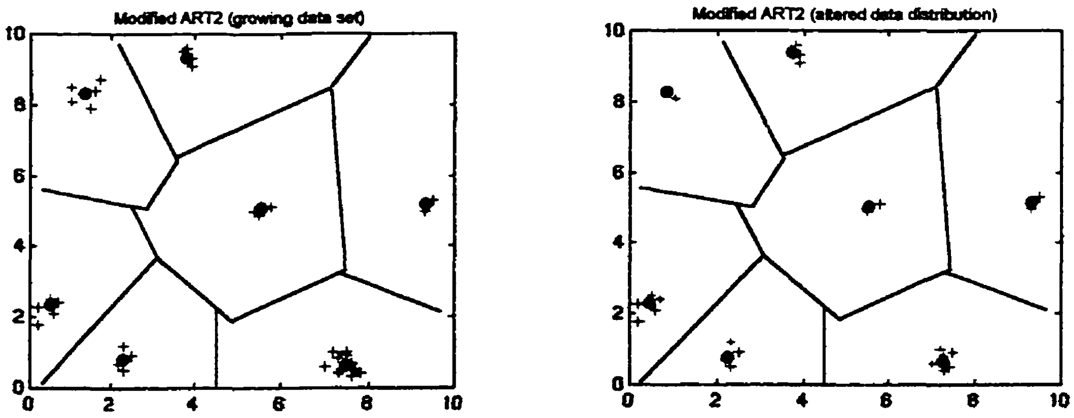


Figure 3.14 Clustering within the 2D collection using modified ART2

The results presented in Figure 3.15 show that the positions of reference vectors obtained for the growing data set and for the set with altered data distribution were almost identical to those obtained for the initial data set (Figure 3.18) . In other words, the modified ART2 satisfied Rijsbergen's first criterion.

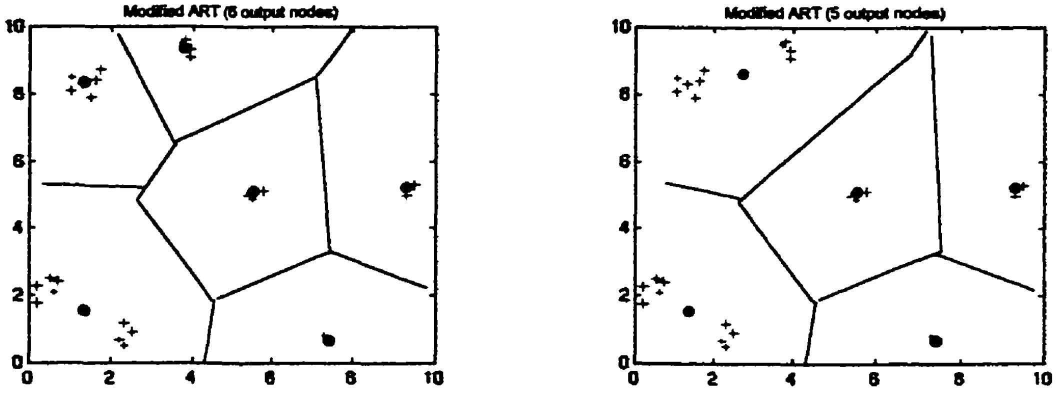


a)

b)

Figure 3.15 Clustering based on modified ART2
for the growing data set and for the set with altered distribution

Moreover, Figure 3.16 shows that modified ART2 could properly handle the reduced number of output nodes. In both cases, due to the reduction of the number of output nodes, the least distinct groups ended up within the same Voronoi region, i.e. having the same reference vector. Groups that were sufficiently distinct from the others, for the given level of clustering tolerance, were recognized as separate clusters.

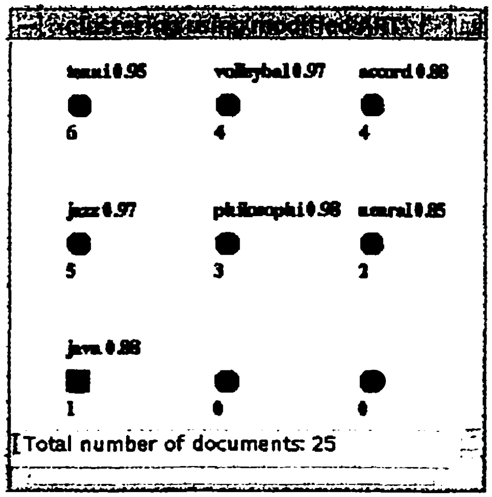


a) b)

Figure 3.16 Clustering using modified ART2 for six and for five output nodes

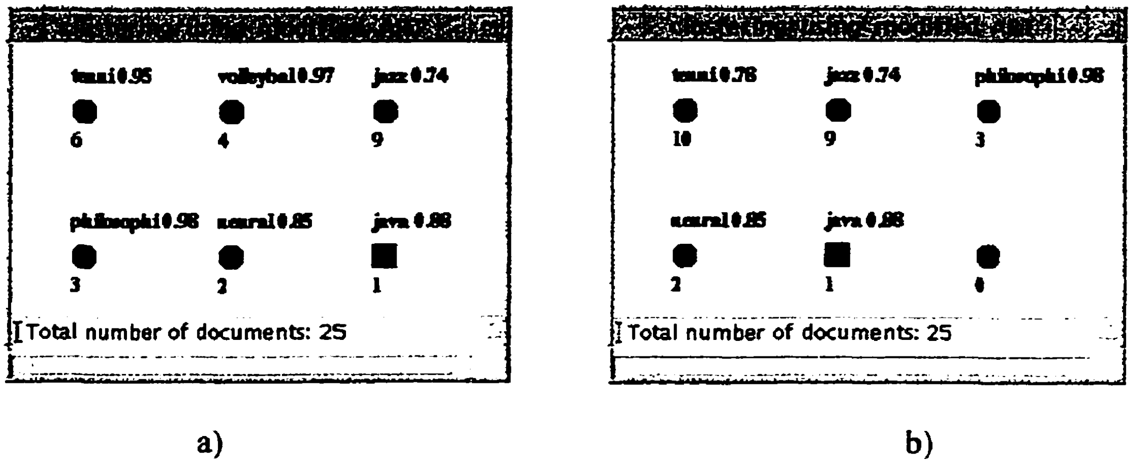
Experiments with original word-vector collection

Figure 3.17 presents the clustering obtained for the original 25-dimensional set of word vectors. For seven output nodes (ρ=1.22) the trained network correctly classified all 25 documents into seven basic groups. However, the real advantage of the algorithm was apparent when the prescribed number of output nodes was reduced to 6 and 5. For six output nodes (ρ=1.35), Figure 3.18 a), the network recognized certain similarities between the groups on *jazz* and *accordion*, placing them into one larger cluster while leaving other clusters intact. For five output nodes (ρ=1.39) a similar coalescence occurred with the groups on *tennis* and *volleyball*, Figure 3.18 b).



Total number of documents: 25

Figure 3.17 Document clustering using modified ART2



Total number of documents: 25

Total number of documents: 25

Figure 3.18 Document clustering using modified ART2 for six and five output nodes

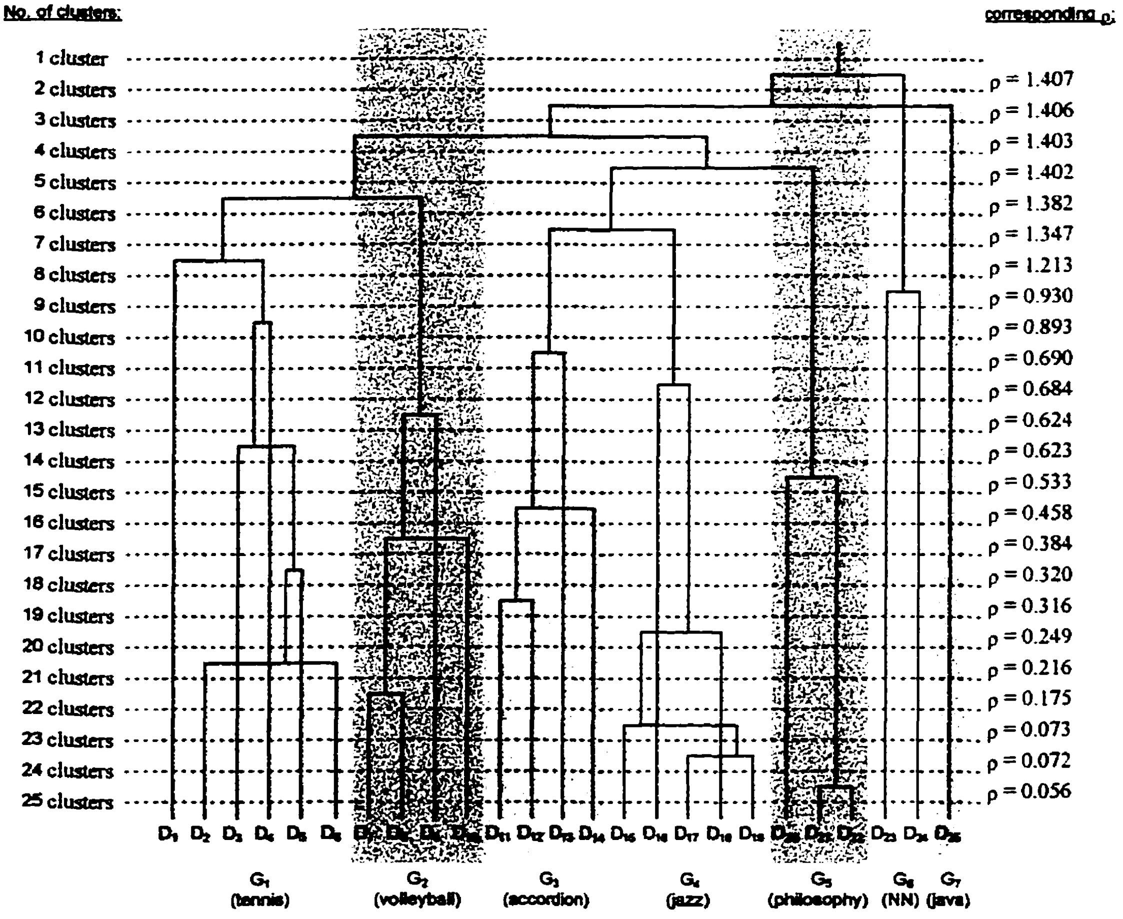


Figure 3.19 Dendogram obtained with modified ART2 for the collection from Table 2.1

All three networks (7, 6, and 5 output nodes) were also tested on a new set of 20 Web pages on the same topics, but with a modified data distribution. The resultant clusterings were appropriate in all cases, when compared to human performance of these tasks at a similar clustering tolerance. This verified once again that networks trained using the modified ART2 learning algorithm could successfully cope with the problem of expanding Web datasets with nonstationary statistics.

Figure 3.19 shows the dendogram obtained for the collection from Table 2.1 using modified ART2 with varying ρ_{max} and n_{max}. The dendogram (search tree) depicts the hierarchical nature of the clustering, and verifies that the modified ART2 algorithm is appropriate for highly efficient multilevel document retrieval, as described in section 3.2.1. Also, it is evident from Figure 3.19 that lower levels of the tolerance parameter corresponded to large numbers of finely-divided categories, whereas large ρ_{max} enabled the system to group together training vectors into fewer categories.

3.7 Modified ART2 with Competitive Hebbian Learning

Most clustering algorithms have a common disadvantage: they do not provide any information on the spatial relationships among discovered clusters. When applied to neural networks for clustering tasks, including the modified ART2, this would mean that there is no clear indication how one reference vector is positioned with respect to others. Some NN algorithms, like the SOM, or the growing cell structures model [3.9], attempt to provide this information by mapping input data onto two- or three- dimensional space, which is shown to lead to suboptimal results in a number of cases, as explained in section 3.4.

The knowledge of spatial relationships among reference vectors, and the corresponding Voronoi regions, undoubtedly helps to obtain a better insight into the nature and complexity of input data. It is particularly useful when the number of output nodes exceeds the actual number of clusters, and very likely each cluster is allocated more than one reference vector. In other words, elements of one group may end up being within several different Voronoi regions. However, since the algorithm provides no indication about the positions of Voronoi regions, it is impossible to know when some input vectors are in fact elements of the same or neighbouring clusters. Situations like that are frequently encountered because the number of actual clusters is largely unknown in advance.

Recently a new algorithm, competitive Hebbian learning (Martinetz 1993), has been proposed, and used quite successfully in overcoming the above mentioned problem, as reported in [3.9]. The following sections give an overview of the algorithm, and show the results obtained when competitive Hebbian learning is combined with modified ART2.

3.7.1 An Overview of Competitive Hebbian Learning

Competitive Hebbian learning is usually not employed on its own but in conjunction with other methods. [3.7]. The algorithm does not change reference vectors at all, but merely inserts a number of topological (neighbourhood) connections, or edges, among the units of the network. The resulting graph is a subgraph of the Delaunay triangulation [3.7], limited to those areas of the input space were data is found. In general, the Delaunay triangulation connects reference vectors having neighbouring Voronoi regions. (Appendix 9)

Competitive Hebbian learning can be described as follows.

STEP 1: Initialize the weight vectors $w_{n_i}(0)$ ($n_i \in N$, N - set of output nodes) randomly according to p(x). Initialize the connection set C with the empty set C={}, i.e. begin with no connections.

STEP 2: Choose an input signal x according to the input distribution p(x).

STEP 1: Determine nodes s_1 and s_2 ($s_1, s_2 \in N$) such that

$$\| w_{s_1} - x \| < \| w_n - x \| \text{ , } \forall n \in N \tag{3.25}$$

and

$$\| w_{s_2} - x \| < \| w_n - x \| \text{ , } \forall n \in N \setminus s_1 \tag{3.26}$$

4. If one does not already exist, insert a connection between s_1 and s_2 to C

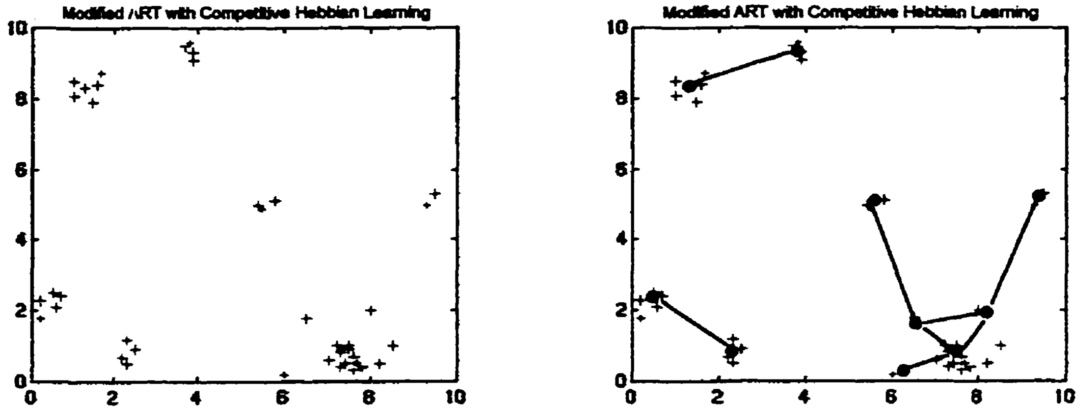
$$C = C \cup \{(s_1, s_2)\} \tag{3.27}$$

5. Continue with step 1, until the maximum number of input signals is reached.

3.7.2 Experimental Results

The experiments presented in this section were conducted using our software modified to incorporate competitive Hebbian learning. In particular, for a given dataset the system would first discover the main clusters using the modified ART2 algorithm, and then for the set of reference vectors obtained, competitive Hebbian learning would be used in order to insert the topological links.

The first experiment was performed for the modified 2D set as presented in Figure 3.20 a), and for the network containing 10 output nodes. Its purpose was to obtain an observable performance measure for the network based on the combination of the modified ART2 and competitive Hebbian learning.

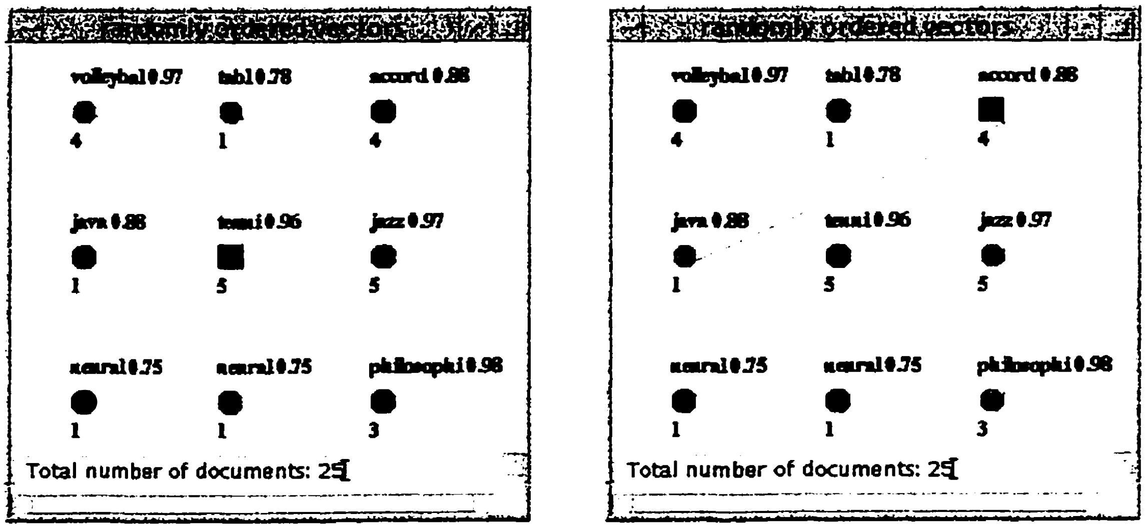


a) b)

Figure 3.20 Modified 2D dataset and the corresponding clustering obtained using modified ART2 with competitive Hebbian learning

As it apparent from Figure 3.20 b) that the network produced satisfactory results. Modified ART2 accurately discovered the centres of the existing clusters for the given number of output nodes, i.e. for the given level of the clustering tolerance. Furthermore, the connections inserted among the reference vectors, generated by competitive Hebbian learning, optimally preserved the topology of the input data.

Red rectangle - node (group) of interest
Green line - topological link
Green circle - neighbouring node (group)



volleybal 0.97 ● 4
tabl 0.78 ● 1
accord 0.88 ● 4
java 0.88 ● 1
tenni 0.96 ■ 5
jazz 0.97 ● 5
neural 0.75 ● 1
neural 0.75 ● 1
philosophi 0.98 ● 3

Total number of documents: 25

volleybal 0.97 ● 4
tabl 0.78 ● 1
accord 0.88 ■ 4
java 0.88 ● 1
tenni 0.96 ● 5
jazz 0.97 ● 5
neural 0.75 ● 1
neural 0.75 ● 1
philosophi 0.98 ● 3

Total number of documents: 25

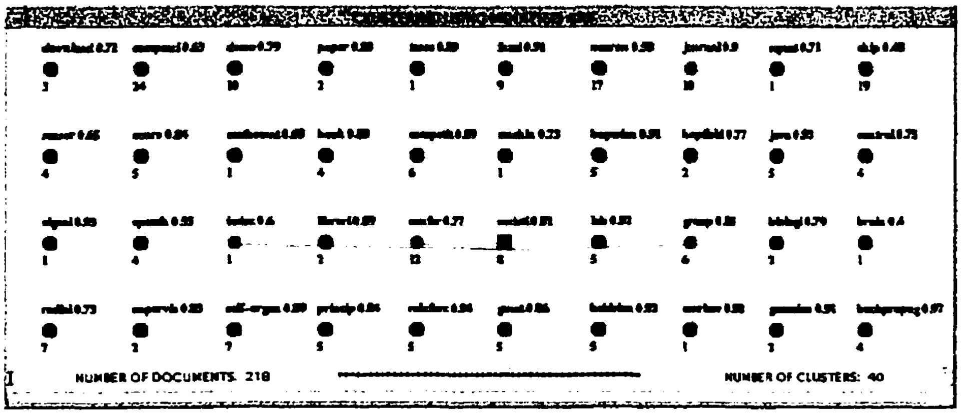
a) b)

Figure 3.21 Document clustering using modified ART2 with Competitive Hebbain Learning

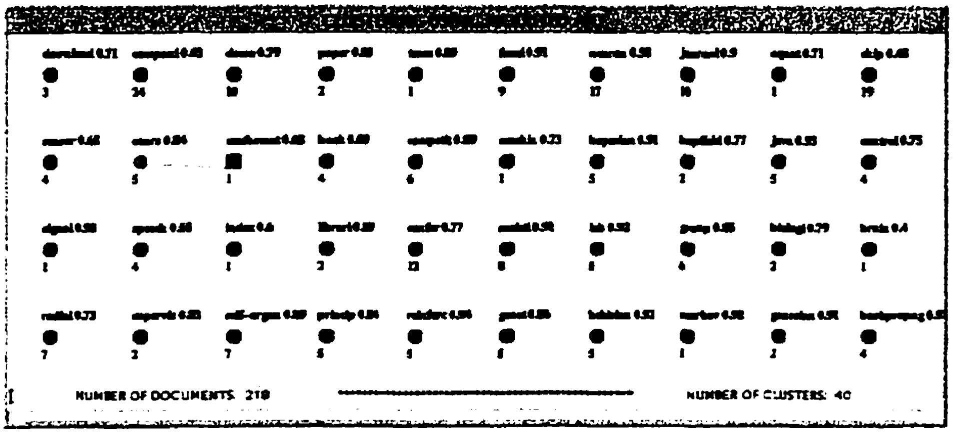
The second experiment (Figure 3.21) was performed for the collection from Table 2.1. The network contained 9 output nodes, and the purpose of the experiment was to test how the two algorithms would cope with the problem of a mismatched number of output nodes, as compared to number of existing clusters.

Figure 3.21 a) shows that, for the case when *tenni 0.96* was the group of interest, the network pointed to the groups *volleybal 0.97* and *tabl 0.78* (the last group contained a document on table tennis) as being the topological neighbours. Similarly, when *accord 0.88* was the group of interest (Figure 3.27 b)) , *jazz 0.97*, *philosophi 0.99,* and *java 0.88* were found to be in its spatial neighbourhood. It is evident that in both cases the network correctly discovered the thematically-closest groups.

ART2 with CHL was also tested on a collection of 218 Web documents on neural networks (Chapter 1). This includes artificial (ANN) and biological neural networks. Figure 3.22 a) presents the case when *societi 0.91*, the group that contained documents on NN societies, was the group of interest. Its topological neighbours were found to be *journal 0.9*, *confer 0.77*, *group 0.85*, and *index 0.6*. These groups contained Web pages related to NN journals, conferences, research groups, and an index page respectively. Undoubtedly the results obtained corresponded to human perception of semantic (thematic) relatedness, since in general societies consist of research groups, and tend to publish journals or organize conferences. For the case presented in Figure 3.22 b) the group of interest (*mathemat 0.68*) contained a Web document on some mathematical aspects of neural networks, and the group (*cours 0.84*) that was found to be the closest consisted of documents on NN courses. It is known that the mathematical basics of NNs are taught through NN courses, which verifies that the network properly identified the most related concept. Finally, the network was tested for the group that contained two Web pages on the biological background of artificial NNs (*biologi 0.79*), as presented in Figure 3.22 c). Since the research related to the biological background of NNs involves knowledge of brain structure, and has been conducted in various NN laboratories (research groups), once again the results obtained were appropriate.



a)



NUMBER OF DOCUMENTS 218 NUMBER OF CLUSTERS: 40

b)

NUMBER OF DOCUMENTS 218 NUMBER OF CLUSTERS: 40

c)

Figure 3.22 Document clustering using modified ART2 and competitive Hebbian learning for the collection of 218 Web document on neural networks

Based on the experimental results presented in this section it is apparent that in a word vector space spatial relationships are actually semantic relationships. Accordingly ART2 with CHL is convenient for document clustering when the number of existing clusters is not known in advance, and the information on topology of input data is of particular importance, or when the user is interested in discovery and retrieval of relevant or related documents.

3.8 Conclusions

Document clustering is the main precondition for efficient document retrieval. However, a clustering algorithm can provide improved efficiency for document retrieval without compromising the accuracy only if it satisfies certain (Rijsbergen's) requirements. Artificial neural networks (ANNs) based on unsupervised learning have the valuable property of generalization. This property means that, once trained, an ANN can accommodate previously 'unseen' items without a complete reclassification, as required with some other techniques in statistical pattern recognition. There are unsupervised ANN techniques, such as the SOM algorithm and hard competitive learning, which provide results dependent on the input data density, and fail to satisfy Rijsbergen's first criterion. Therefore they should be employed only for the purpose of improved information browsing. The modified ART2 algorithm overcomes the main drawbacks of the SOM and HCL, and satisfies all Rijsbergen's requirements. Moreover, enabling the creation of search trees (dendograms) it is convenient for highly efficient multi-level document retrieval. On the other hand, modified ART2 does not preserve the topology of input data, and can not provide an optimal performance if the number of output nodes significantly exceeds the number of existing clusters. It has been shown that modified ART2 in conjunction with competitive Hebbian learning can successfully cope with the problem of a mismatched number of output nodes and actual clusters. In addition, these two algorithms enable the discovery of thematically related groups of Web documents.

References

[3.1] C. J. van Rijsbergen 1975. *Information Retrieval.* Boston: Butterworths.

[3.2] Christopher M. Bishop 1995. *Neural Networks for Pattern Recognition.* Oxford: Clarendon Press.

[3.3] Simon Haykin 1994. *Neural Networks.* New Jersey: Prentice Hall.

[3.4] S. Kaski, T. Honkela, K. Lagus, and T. Kohonen 1996. *Creating an Order in Digital Libraries with Self-Organizing Maps.* In Proceedings of WCNN'96, World Congress on Neural Networks, pp. 814-817. Mahwah, NJ: Lawrence Erlbaum and INNS Press.

[3.5] Howard Demuth, Mark Beale 1996. *Neural Network Toolbox for Use with Matlab.* Natick, Mass.: The Mathworks, Inc.

[3.6] Gail A. Carpenter and Stephen Grossberg 1988. *The ART of Adaptive Pattern Recognition by a Self-Organizing Neural Network.* IEEE Computer Magazine, pp. 77-88, March 1988.

[3.7] Bernd Fritzke 1997. *Some Competitive Learning Methods.*
http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/JavaPaper/

[3.8] Stephen Grossberg 1987. *Competitive Learning: From Interactive Activation to Adaptive Resonance*. Cognitive Science 11, pp. 23-63, 1987.

[3.9] Bernd Fritzke 1997. *Unsupervised Ontogenic Networks*. IOP Publishing Ltd and Oxford University Press: Handbook of Neural Computation. Release 97/1.

[3.10] Gail A. Carpenter and Stephen Grossberg 1992. *A Self-Organizing Neural Network For Supervised Learning, Recognition, and Predicition*. IEEE Communications Magazine, pp. 38-49, September 1992.

[3.11] Yoh-Han Pao 1989. *Adaptive Pattern Recognition and Neural Networks*. Boston: Addison-Wesley Publishing Company, Inc.

[3.12] Teuvo Kohonen. 1984. *Self-Organization and Associative Memory*. New York: Springer-Verlag.

[3.13] Gerard Salton 1971. *The SMART Retrieval System; Experiments in Automatic Document Processing*. Englewood Cliffs, N.J.: Prentice-Hall.

[3.14] Gerard Salton 1989. *Automatic Text Processing: the Transformation, Analysis, and Retrieval of Information by Computer*. Boston: Addison-Wesley.

[3.15] Gail A. Carptenter and Stephen Grossberg 1987. *ART2: Self-Organization of Stable Category Recognition Codes for Analog Input Patterns*. Applied Optics, 1987, 26, pp. 4919-4930.

CHAPTER 4

Hypertext Clustering

4.1 Introduction

The WWW is an on-line hypertextual collection, and a more sophisticated algorithm for improved document clustering may have to be based on combined term-similarity and hyperlink-similarity measures. Recently several hypertext clustering algorithms have been proposed. These provide satisfactory results but only when applied to particular problems: either the collection should contain Web document from the same site, or the documents are supposed to be interlinked. Moreover, some of the techniques require knowledge of the data that is not accessible from a common user's desktop.

This chapter presents a new technique for hypertext clustering. The technique, called *adaptive hypertext clustering* (AHC), is based on modified ART2, and incorporates a few major improvements over earlier methods. AHC is not limited by the accessibility of input data, and could be equally useful when applied to smaller collections, such as bookmark sets, or larger ones on-line catalogs and search engines. This chapter contains examples of results obtained using adaptive hypertext clustering, suggesting the benefits of using AHC over the *text-only* or *hyperlink-only* based Web document clustering techniques.

4.2 The WWW - Hypertext Collection

Traditional publishing media (books, journals, ...) are *linear* in nature, which means that they have an obvious beginning and end, and a fixed number of numbered pages in between. Therefore readers always know, for example, the exact size of a book, and where they are in the book. Even the media of music, video, and film are linear in the sense that they represent a predetermined sequential presentation of sounds or images.

In contrast to linear media, *hypertext* is a collection of text and graphics that is interconnected in a complex way, and can not be conveniently presented on paper. It has no regular structure, and users are free to explore and assimilate information in different nonlinear ways. Even though hypertext, as a concept, existed long before the Internet, it was the birth and explosive growth of the WWW that mostly contributed to its popularity.

The WWW is global hypertext. In other words, it is hypertext enriched with the ability to connect with sites around the world. The basic idea is for each WWW site to store its publications, information about its research, academic or commercial interests and activities, and any other data that might be useful to others. The site then runs a *server* program which accepts electronic requests for document pages from *clients* elsewhere on the Internet, using the *hypertext transfer protocol* (http). The server transfers the requested page to the client where it can be viewed using a special *browser* program that sends http requests and displays the results.

In general, the WWW consist of nodes (Web pages) and links.

Nodes
Each node is a plain ASCII text file with embedded *hypertext markup language* (HTML) commands, and usually represents a concept or an idea. The HTML commands, also known as tags, describe the structure of a document, provide font and graphics information, and define hyperlinks to other Web pages, or other Internet resources including plain (unformatted) text, PostScript files, diagrams and pictures in X bitmap, gif and jpeg encoding, mpeg movies, and various audio formats. [4.1] Every node is uniquely defined with its URL (*universal resource locator*). In general a URL has the form:

protocol://location/file#destination

where *protocol* is http (in some cases could be ftp, or gopher), *location* is the Internet machine, *file* is the pathname of the resource on the remote machine, and the optional *destination* is the name of a *source anchor* tag marking a point within the target file.

Links
Hypertext *links* are not physical things and exist only at the conceptual level. They connect related concepts or nodes, and are normally directed. The node from which a link originates is called the *reference* and the node at which a link ends is called the *referent*. Each link has a beginning (the highlighted text within the reference node, called an anchor) and an end (the name of the destination). Web links cross site boundaries, and therefore they qualitatively differ from classical hypertext links. Since there is a huge amount of information on the Internet that is available not through http, but through other

data transfer protocols and programs, such as ftp and gopher, some links cross inter-protocol boundaries. Hypertext links within a site, in terms of the file system hierarchy, can be *upward*, *downward*, *crosswise*, and *outward*.

The Web is inherently nonlinear, and there is an endless number of ways of proceeding from one node to another, or from one physical site to another, as illustrated in Figure 4.1.

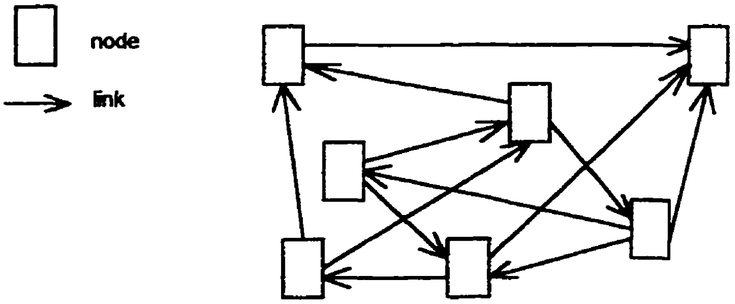


Figure 4.1 Nonlinear organization of the Web

Although the WWW is nonlinear as a whole, most of its subelements (sites) exhibit a high degree of linearity. This is mainly because each Web designer, or *webmaster*, tend to organize and connect Web pages in a logical way, and thereby make the site attractive to visitors, and encourage their return. Moreover, a well organized Web site, with clear indications of its size, content, and structure enables visitors to easily browse and retrieve the information of their interest, avoiding the well known danger of 'being lost in hyperspace'.

The following section presents the most frequently encountered linear Web locality structures.

4.2.1 Linear Web locality structures

Even though the design of Web pages, and the number and type of connections among them are the result of the webmaster's personal choice and creativity, most Web sites follow one of the two fundamental structural concepts.

- *Horizontal linearity*

The *horizontal linearity* concept assumes that one index or home page is provided, and it contains links to all or most of the other 'standard' Web pages on the site. The standard Web pages may contain a *back link* to the index page, or links to other related pages. The index and the standard Web pages are on the same or on two different levels of the file system hierarchy. This concept is mostly used at smaller sites (collections), or for the organization of personal Web pages.

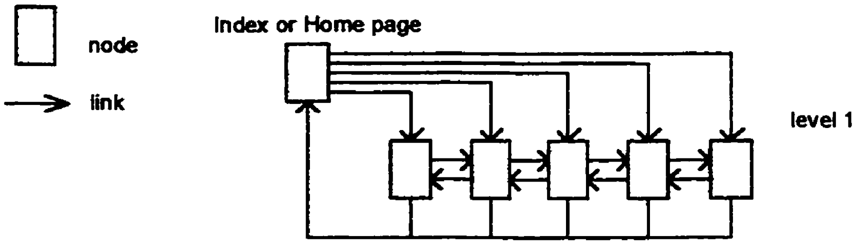


Figure 4.2 Web locality based on *horizontal linearity* concept

- *Vertical linearity*

The *vertical linearity* concept assumes that there are two or more levels down from the main index (home) page. This concept enables the classification and grouping of a large amount of material, and therefore is more frequently used than the previous concept. Evidently in a system that employs vertical linearity the position of a Web page in the file system hierarchy implicitly describes its content and importance, when regarded from the overall site perspective.

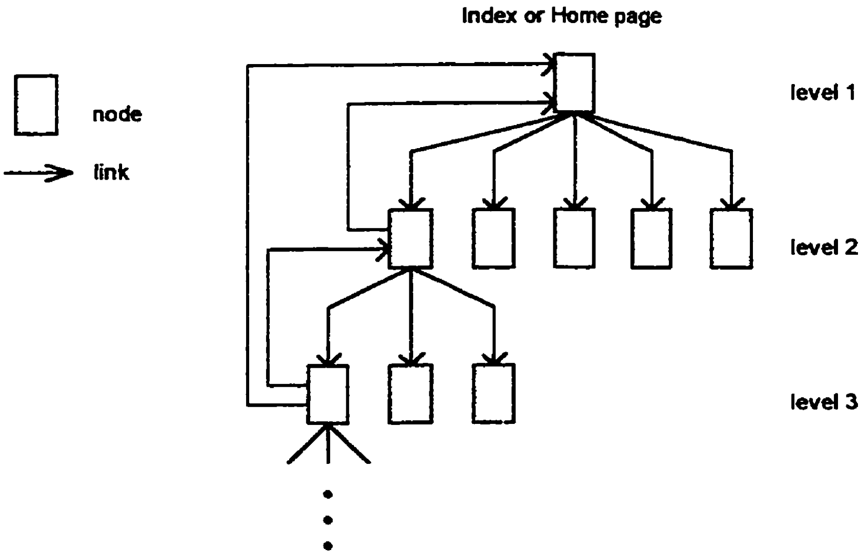


Figure 4.3 Web locality based on *vertical linearity* concept

4.2.2 An overview of existing techniques for hypertext clustering

It is apparent from both linear Web locality structures presented in section 4.2.1, particularly the latter one, that the content and importance of a Web page is a compound function of its *textual* and *hyper* portions, where the hyper part is the information determined by the position and function of the Web page with respect to the surrounding Web space.

A more sophisticated clustering algorithm intended to provide improved results in a hypertext space may have to incorporate all available information. Most current search engines neglect the hyper dimension of the Web. Their information management is entirely based on textual information. Using a limited knowledge of hypertext documents, they do not have the ability to provide refined results. For example, responding to a user's query, most search engines simply select a list of the best matching Web pages, without recognizing that some may be linked, may be mutually dependent, or may form part of a larger composite document. Recently, several advanced techniques for hypertext classification have been proposed. This section presents some examples.

HyPursuit

"HyPursuit is a hierarchical network search engine that clusters hypertext documents to structure a given information space for browsing and search activities" [4.2]. Its document similarity function is a compound function of two mutually independent variables: term similarity and hyperlink similarity. According to the authors, link information may help in grouping together all nodes or Web pages that comprise individual papers (conference on-line proceedings, for example). On the other hand, term or content information may help to cluster thematically related pages. The term similarity measure is based on the word-vectors generated using normalized term frequencies (tf) only. Since this is a significant weakness, some alternatives which would include collection frequencies are being investigated. The hyperlink similarity measure between two documents D_i and D_j is a linear combination of three variables: the length of the shortest path between the documents, the number of ancestor documents that refer to both D_i and D_j , and the number of descendant documents that both D_i and D_j refer to. Clustering is based on the complete link method as in section 3.2.2, and "...although faster clustering algorithms exist, we chose the complete link method because it was easy to implement" [4.2]. Considering the hyperlink similarity measure of HyPursuit, it seems reasonable to be employed for documents entirely placed on a single server together with all their ancestor and descendant documents. However, in the case of standard Web pages, it may be very inefficient to explore every existing path between two documents distributed over the Web server space.

Generalised Similarity Analysis

According to generalized similarity analysis [4.3], the overall similarity between two hypertext documents is a combination of three components: *content similarity*, *hypertext linkage*, and *state-transition patterns*. Word-vectors, necessary for the content similarity computation, are built on the tf-idf model and, as mentioned in the introduction, the word-vector space for a collection is created using the 500 most frequent terms. The similarity between two documents D_i and D_j by hypertext linkage is the ratio of the number of hyperlinks from D_i to D_j to the number of all outgoing links from D_i . "For simplicity, ancestors and descendants are not considered" [4.3]. If all links from D_i directly lead to D_j the ratio is 1, and D_i is either identical or very similar to D_j . If on the other hand there is no link between D_i and D_j the ratio is 0 and they are completely different according to the criterion of hypertext linkage. However, this does not imply that the overall similarity is 0, since it may be that the other two similarities are significant. The similarity by state-transition patterns is based on access logs maintained by a number of web servers, which provide valuable information on how users actually access the information on a server. For example, information on the number of users who followed a hyperlink connecting two documents might indicate the degree of relatedness between the two documents. The step transition probability from D_i to D_j is the ratio of the number of transitions from D_i to D_j to the total number of transitions starting from D_i. The overall similarity by state transition is a function of one step transition probability, and it is consistent with linkage and content similarity models. A difficulty in the implementation of GSA stems from the fact that some servers do not provide state-transition patterns, or they are not publicly available.

Hyper Search Engines

Although this method was originally intended to increase the precision of current search engines, its basic ideas are applicable to hypertext clustering algorithms. According to [4.4], theoretically "...the analysis of the informative content of a web object (page) should involve all the web objects that are reachable from it via hyperlinks ("navigating" in the WWW)". It is not feasible in practice to examine all possible outgoing links including all their children and corresponding objects, since a single Web page could be indirectly connected to an indefinite number of other pages. For that reason, a measure of the depth of pages has been introduced. For a given document D_i the (relative) depth of another document D_j is equal to "...the number of links that have to be activated (clicked) in order to reach D_i from D_j" [4.4]. The examination of the hypertextual surrounding of D_i includes the documents with depth less then $k = 3$. The contribution of a web document at depth k to the overall hyper information of D_i is diminished via a fading factor ($0 < F < 1$) dependent upon k. Knowing that users cannot retrieve all the links at the same time, it is assumed that they first select the most promising (related) links. Accordingly, the hyper information for the best link is multiplied by F, for the second best link by F^2, and so on.

This theory, based on the analysis of hypertextual environments, could solve the problem of index (organizational) or split documents. If a page is a part of a composite document, it contains only a portion of the overall information. Classifications based on incomplete data tend to provide unreliable results. The main disadvantage of this method is that it may require the retrieval of a large number of documents, especially if the Web page of interest is a link-page (a page that exclusively contains links to other pages on a particular topic).

4.3 Hyper Dimension - Based Clustering

Most of the currently known techniques for hypertext clustering that exploit both the topology (hyperlink organization and structural features of Web pages) and textual similarity between items, including the onc presented in section 4.2.2, seem to be impractical for wider use for two main reasons. These techniques either require the knowledge of some publicly unavailable data, such as state-transition patterns, or information that is difficult if not impossible to collect, such as the number of all existing links between two pages. Moreover, they specialize in the problems of hypertext categorization within the same site, or categorization of Web pages that are interlinked to a large extent.

In our research we focused on the creation of a technique for hypertext clustering that could be universally employed. In other words, the technique was not to be limited by the computational expense or accessibility of its input data. It was intended to provide improved clustering for either large scale problems, for example those faced by search engines, or small scale problems, including organizing a bookmark set.

The following sections present the main features of our technique for clustering based on hyper dimensions, and the results obtained. In particular, section 4.3.1 introduces the hyper dimensions employed by the algorithm, and offers the reasons for their utilization. Section 4.3.2 presents the clustering results regarding a collection of 25 Web pages on the same topic (neural networks).

4.3.1 Hyper Dimensions

Experimenting with Web documents of various profiles and sizes we observed that in many cases the meaning and content of a page were not delivered entirely in textual form. Moreover there were situations in which taking the textual contents of some Web pages into consideration for the purpose of their classification was completely misleading. It was evident that some information, including page length, number of images and links, the percentage of upward, downward and outward links, etc. gave valuable information on Web page functionality, and thereby provided for more accurate clustering.

The functionality of a Web page is exclusively based on the features it exhibits, and does not depend on its content. The same functional categories can be found in different thematic domains. Based on our experience the most frequently encountered functional categories are:

- *index or link pages*

These pages contain a number of primarily outward links to other pages on a particular topic. Accordingly, they do not deliver any specific information, but serve as navigational points. They usually have large link density, a relatively small number of images, and are placed deeper in the file system hierarchy.

- *organizational home pages*

Organizational home pages serve as general navigational points within a site. In other words these pages are the entry points for institution and organization Web sites. Usually they are placed at the top of the file system hierarchy (e.g. http://www.*university*.edu/), and contain a number of downward links. They may contain many graphical elements (images) compared to the amount of textual information.

- *departmental home pages*

In contrast to organizational pages, departmental home pages are navigational points for a particular subdomain within a site. They appear deeper in the file system hierarchy, and contain a large number of upward and downward links. However, the downward link density is usually greater than the upward link density.

- *content pages*

These pages are not intended to provide improved navigation, but simply to deliver information. They are quite deep in the file system hierarchy, and mostly do not contain many links. In addition, they may be of relatively large size, with few images compared to the amount of textual information.

- *pages of special content*

Pages of special content in addition to delivering some information, also contain links that cross interprotocol boundaries, or point to files with extensions other than *.html* or *.htm* . For example, certain Web pages on neural networks contain links to files with extensions including *.ps*, *.zip*, *.java* , *.exe* , ... Other thematic domains may contain, for instance, links to audio or video files.

The following figure depicts the functional topology of a typical site. In other words, it illustrates standard positions and connectedness of Web pages that belong to different functional categories.

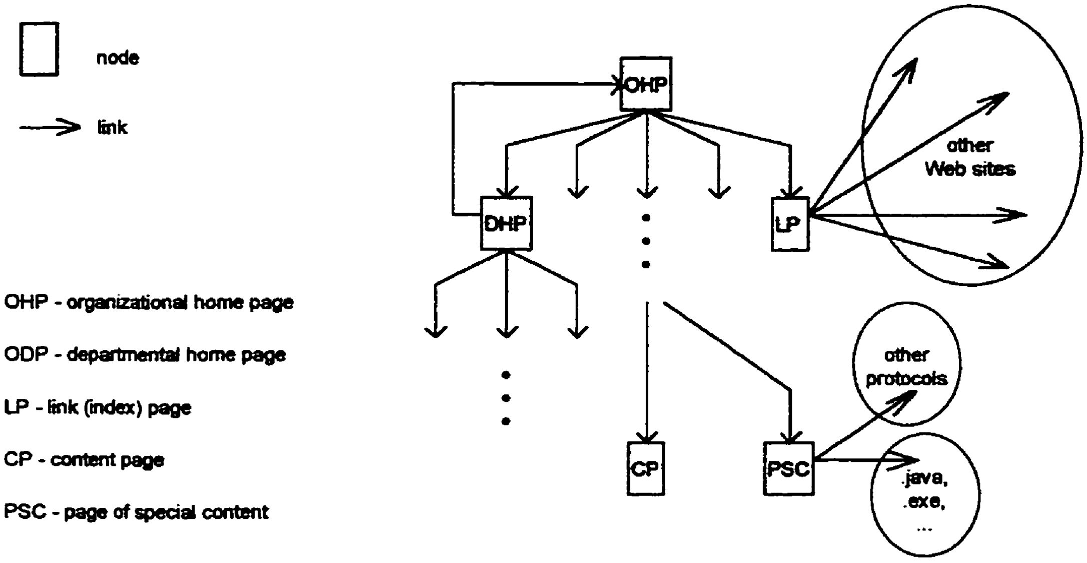


Figure 4.3 Functional topology of a typical Web site

Table 4.1 presents a selection, based on our experience, of features that would be the most helpful in discriminating among the main functional Web page categories. It should be emphasised that numerical (quantitative) values related to these features can be calculated using information retrievable directly from the corresponding html source codes, and therefore are convenient to implement.

node type	depth	size	links	% outward links	% upward links	% crosswise & downward links	anchors	images	e-mail or news	other protocols	.java, .exe, ... files
OHP	Sm	Sm / Me	M / S	S / N	N	M	S / N	M / S	M / S	N	N
DHP	Me	Sm / Me	M / S	S / N	S	S	S / N	S	M / S	N	N
LP	Me	Me	M	M	S	S	N	N	S	S / N	N
CP	Me / Lg	Me / Lg	S / N	N	S / N	S - N	S / N	S / N	S / N	N	N
PSC	Me / Lg	Me / Lg	S / N	N	S / N	S - N	S / N	S / N	S / N	S	S

Sm - small	M - many
Me - medium	S - several
Lg - large	N - none

Table 4.1

In order to enable clustering that would group Web documents into functional categories, we have introduced 12 *hyper dimensions* or *weights*. Each dimension is defined with a function that takes one of the features presented in Table 4.1 as its argument. The functions are intended to ensure normalization, and maximum discrimination power to the corresponding features. The first four dimensions provide normalization of a Web document to the collection, while the remainder provide normalization within the document itself. In general, normalization of a dimension ensures that its numerical value for any vector in the corresponding space is rescaled to (0,1) range. Thereby the vectors of the corresponding space become comparable with each other in terms of the given dimension.

1) *depth weight*

Depth weight is defined using the expression given in (4.1).

$$\text{depth\_weight} = \left(\frac{\dfrac{\text{depth\_max}}{\text{depth}} - 1}{\dfrac{\text{depth\_max}}{\text{depth\_min}} - 1} \right)^{n_1} \qquad (4.1)$$

The depth of a document is measured by the number of '/' in the URL. *Depth_max* and *depth_min* are the maximum and minimum depth, respectively, encountered for a group of Web documents that is to be clustered. Parameter n_1 should provide an exponential increase to the function, and in our case $n_1 = 3$.

According to the shape of the function, only documents that are relatively high in the file system hierarchy, such as organizational home pages, will have a large depth weight, whereas all others will have depth weights close to zero.

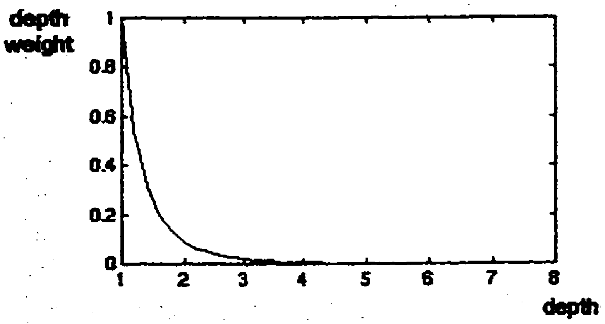


Figure 4.4 (a) Depth weight: depth_max = 8, depth_min = 1, n_1=3

2) *text length weight*

Text length weight is defined by

$$\text{textLength\_weight} = \left(\frac{\text{textLength} - \text{textLength\_min}}{\text{textLength\_max} - \text{textLength\_min}} \right)^{n_2} \qquad (4.2)$$

The text length of a document is measured in bytes. *TextLength_max* and *textLength_min* are the maximum and minimum text length, respectively, encountered for a group of Web documents that is to be clustered. In a similar manner to n_1, n_2 is intended to provide an exponential increase to the function, and in our case $n_2 = 3$.

Large text length weights correspond to documents that contain a great deal of textual information, such as content pages, while documents of less textual information will have a correspondingly lower weight.

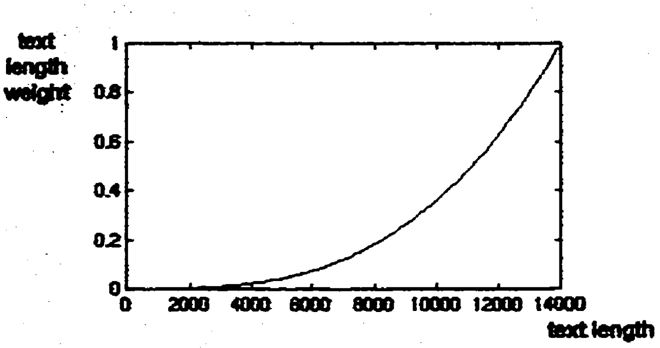


Figure 4.4 (b) Text length weight: textLength_max = 14 000 (bytes), textLength_min = 10 (bytes), n_2=3

3) *link density weight*

This is defined by

$$\text{linkDensity\_weight} = \left(\frac{\text{linkDensity - linkDensity\_min}}{\text{linkDensity\_max - linkDensity\_min}} \right)^{n_3} \tag{4.3}$$

The link density of a document is the *number of links / text length* ratio. *LinkDensity_max* and *linkDensity_min* are the maximum and minimum link density, respectively, found within a group of Web documents that is to be clustered. In this case $n_3 = 9$.

High link density weights will have documents with many links compared to the amount of textual information. This property is common to link pages and, in some cases, to organizational home pages.

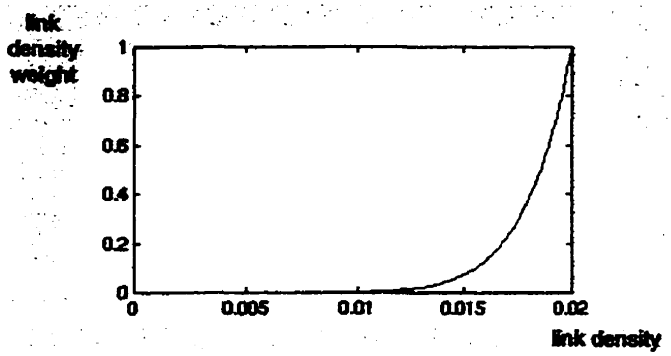


Figure 4.4 (c) Link density weight: linkDensity_max = 0.02, linkDensity_min =0.0 , n_3=9

4) *image density weight*

Image density weight is defined by

$$\text{imageDensity\_weight} = \left(\frac{\text{imageDensity} - \text{imageDensity\_min}}{\text{imageDensity\_max} - \text{imageDensity\_min}} \right)^{n_4} \quad (4.4)$$

The image density of a Web page is the *number of images / text length* ratio. *ImageDensity_max* and *imageDensity_min* are the maximum and minimum image density, respectively, found within a group of Web documents that is to be clustered. Similar to n_3, n_4 = 9.

Although Web pages of all functional categories may contain images, high image density weight is primarily a property of organizational home pages. These pages tend to deliver considerable information in graphical form, compared to the amount of textual information, in order to appear attractive to users.

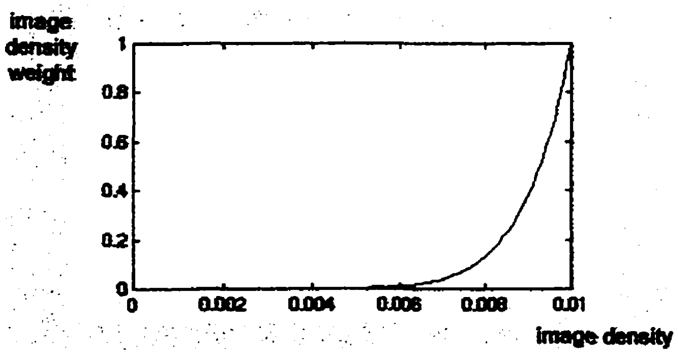


Figure 4.4 (d) Image density weight: imageDensity_max = 0.01, imageDensity_min =0.0 , n_4=9

5) *text under links weight*

This is defined by

$$\text{textUnderLinks\_weight} = \frac{\text{textUnderLinksLength}}{\text{textLength}} \tag{4.5}$$

This dimension is intended to provide discrimination between the pages that contain links within images such as navigational maps (organizational home pages), and those that have links embedded in text (link pages, content pages, ...).

6) *outward link density weight*

This is defined by

$$\text{outwardLinkDensity\_weight} = \begin{cases} 0 & , \text{outwardLinkDensity} \leq \text{a} \\ 0.5 & , \text{a} < \text{outwardLinkDensity} < \text{b} \\ 1 & , \text{b} \leq \text{outwardLinkDensity} \end{cases} \tag{4.6}$$

The outward link density of a Web page is the *total number of links / number of outward links* ratio. *Outward link density weight*, as defined in (4.6), provides discrimination between the pages with very high, average and low percentages of outward links. Accordingly, for most link pages this weight would be 0.5 or 1, while for organizational and departmental pages it would be 0 or 0.5. In this case a = 0.3, and b = 0.7.

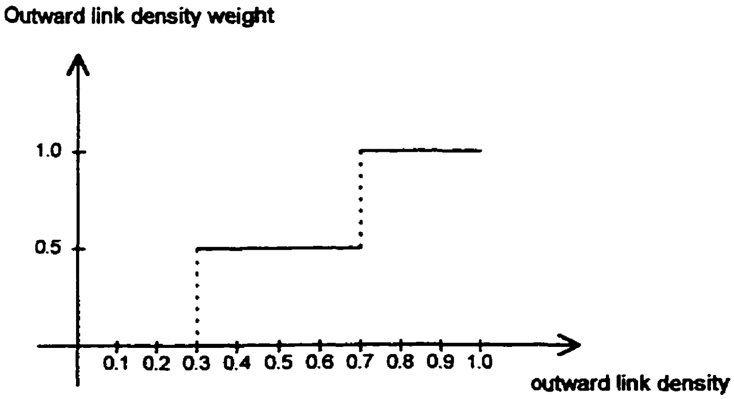


Figure 4.4 (e) Outward link density

7) *upward link density weight*

Upward link density weight is defined using the same concept as outward link density weight (4.6). For the pages with a sufficiently high percentage of upward links this weight would be 1, while a lower percentage of upward links would result in a weight of the value 0.5 or 0.

8) *downward & crosswise link density weight*

Again, *downward & crosswise link density* is defined similarly to (4.6). Accordingly, this weight would have the value 0.5 or 1 for most organizational and departmental home pages.

9) *links to other protocols weight*

$$\text{linksToOtherProtocols\_weight} = \begin{cases} 0 & , \text{numberOfLinksToOtherProtocols} = 0 \\ 0.5 & , \text{numberOfHttpLinks} > \text{numberOfLinksToOtherProtocols} \\ 1 & , \text{numberOfHttpLinks} < \text{numberOfLinksToOtherProtocols} \end{cases} \quad (4.7)$$

Links to other protocols weight, as defined in (4.7), is supposed to have the highest value (0.5 or 1) for content pages, or pages of special content, while for the other functional categories it may be 0 or 0.5.

10) *anchor weight*

$$\text{anchor\_weight} = \begin{cases} 0 & , \text{numberOfAnchors} = 0 \\ 1 & , \text{numberOfAnchors} \neq 0 \end{cases} \qquad (4.8)$$

An *anchor* is a link that originates from and points to the same Web page. Therefore anchors appears within Web documents that contain a great deal of information and provide for their better organization and understanding. Accordingly, *anchor weight* as defined in (4.8) is 1 primarily for content pages.

11) *e-mail and news weight*

E-mail and news weight is based on the same concept as given in (4.8). It means that Web documents with at least one *e-mail* or *news* link (mostly organizational and link pages) will have for this weight the value 1, otherwise this weight will be 0.

12) *links to .java, .exe, ... files weight*

The *links to .java, .exe, ... files weight* is also based on the concept in (4.8). As mentioned earlier, it is intended to discriminate between pages of special content and other functional categories. For thematic domains other than neural networks this weight may, for example, be related to different types of files.

4.3.2 Experimental Results

The results presented in this section are regarding a collection of 25 Web documents that belonged to the same thematic domain (neural networks), but to different functional categories (Table 4.2). The URLs of the Web pages are given in Appendix 10.

topic (category)	number of documents
NN links	5
NN theory	5
NN companies	5
NN labs	5
NN software	5

Table 4.2

The purpose of the experiment was to compare the clustering results obtained for the same set of documents using the textual content, i.e. the corresponding word vectors, against those using the hyper dimensions only.

Initially, text-only clustering based on the modified ART2 algorithm (as presented in Chapter 3) was performed, in a word vector space that was created using the modified TF/IDF model (as presented in Chapter 2). The first twenty-five words (stems) with the greatest discriminatory power, and accordingly the word vector dimensions in this case, were found to be:

byte, java, instal, download, applet, demo, group, genova, network, confer
neural, mixtur, perceptron, univers, lab, research, dib, neurosolut, learn, artifici,
product, scienc, html, theori, recognit

It is evident from Figure 4.5 that the text-only clustering did not provide sensible results. Even for the lowest levels of clustering tolerance some misclassifications occurred, such as the merging of documents D_{13} and D_{22}, D_7 and D_{14}, D_4 and D_8, ... We have observed that the poor clustering was mainly caused by the limited discrimination power of the overall word vector space. In particular, some words that served as the word vector dimensions appeared in documents of different groups, and consequently resulted in partially overlapped clusters. Moreover, the meaning and functionality of the NN link-pages was indicated, not in the words they contained, but almost exclusively in their structure and topology. Therefore they presented noisy data to the text-based clustering. An illustration is the page presented in Figure 4.6. It is apparent that without its hyper structure this link page was simply a set of unrelated statements (words), and as such contributed to the failure of both word vector space creation and text-based clustering.

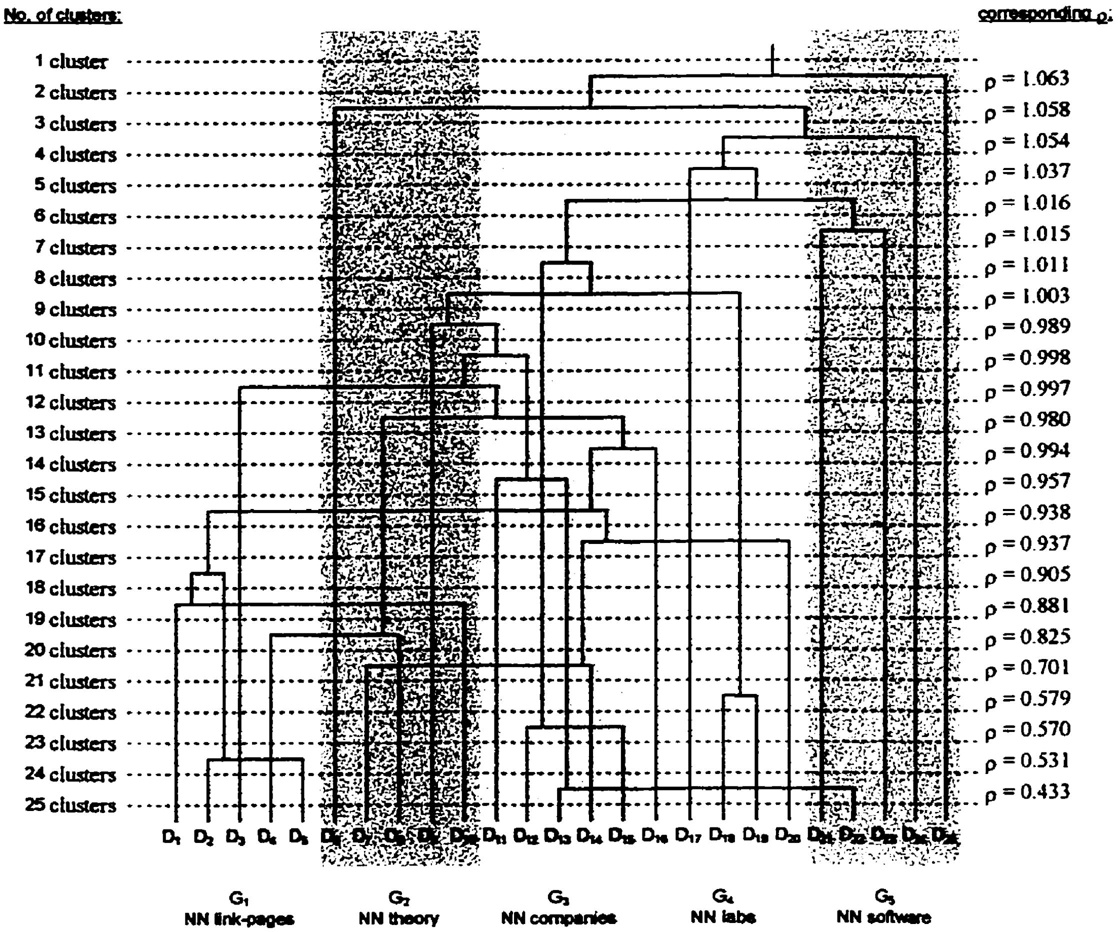
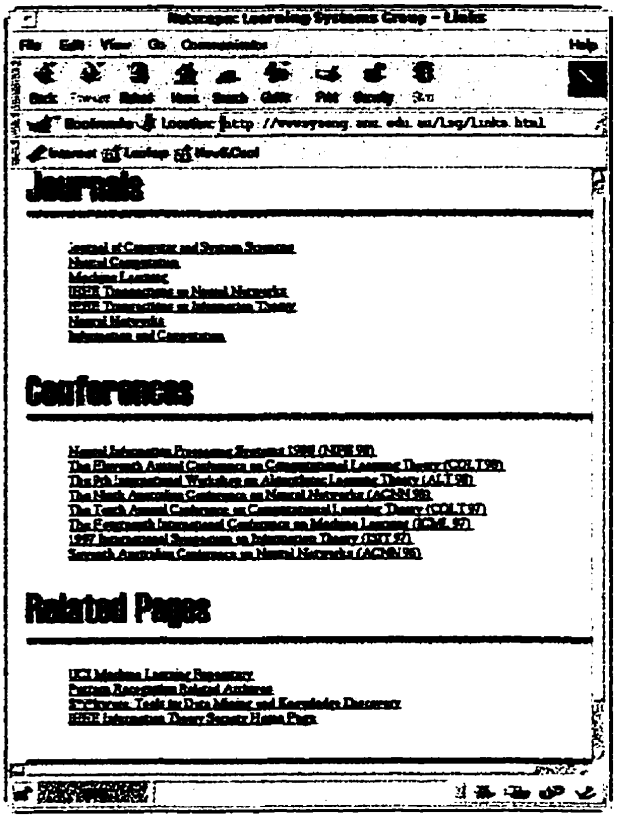


Figure 4.5 Dendogram obtained for text based clustering on the collection from Table 4.2



http://www.syseng.anu.edu.au/lsg/links.html

Journals

Journal of Computer and System Sciences
Neural Computation
Machine Learning
IEEE Transactions on Neural Networks
IEEE Transactions on Information Theory
Neural Networks
Information and Computation

Conferences

Neural Information Processing Systems 1998 (NIPS 98)
The Eleventh Annual Conference on Computational Learning Theory (COLT 98)
The 9th International Workshop on Algorithmic Learning Theory (ALT 98)
The Ninth Australian Conference on Neural Networks (ACNN 98)
The Tenth Annual Conference on Computational Learning Theory (COLT 97)
The Fourteenth International Conference on Machine Learning (ICML 97)
1997 International Symposium on Information Theory (ISIT 97)
Seventh Australian Conference on Neural Networks (ACNN 96)

Related Pages

UCI Machine Learning Repository
Pattern Recognition Related Archives
Software Tools for Data Mining and Knowledge Discovery
IEEE Information Theory Society Home Page

Figure 4.6 A link page of the collection from Table 4.2

In the case of the large collection of 218 Web pages on neural networks presented in Chapter 2, we used the interterm statistics to solve the problem of overlapped clusters and noisy data, and thereby to improve the clustering. However, this time the collection was of relatively small size and it was impossible to gather enough word related statistical information to obtain the same kind of improvement.

In contrast to the text based clustering, the clustering based on the hyper dimensions provided very satisfactory results. The corresponding dendogram is given in Figure 4.7. It is evident that, even for a small number of output nodes, the documents were in most cases properly classified. In particular, the only misclassifications occurred for some documents of the groups NN companies and NN labs. However, those were reasonable misclassifications since in general the documents from both groups had the properties of organizational home pages. Therefore the two groups completely merged in a larger cluster when the number of output clusters was reduced to five. On the other hand, there was no misclassification for the documents of the groups NN link-pages, NN theory, and NN software, since these groups had the properties of distinguishable functional categories: link pages, content pages, and pages of special content, respectively.

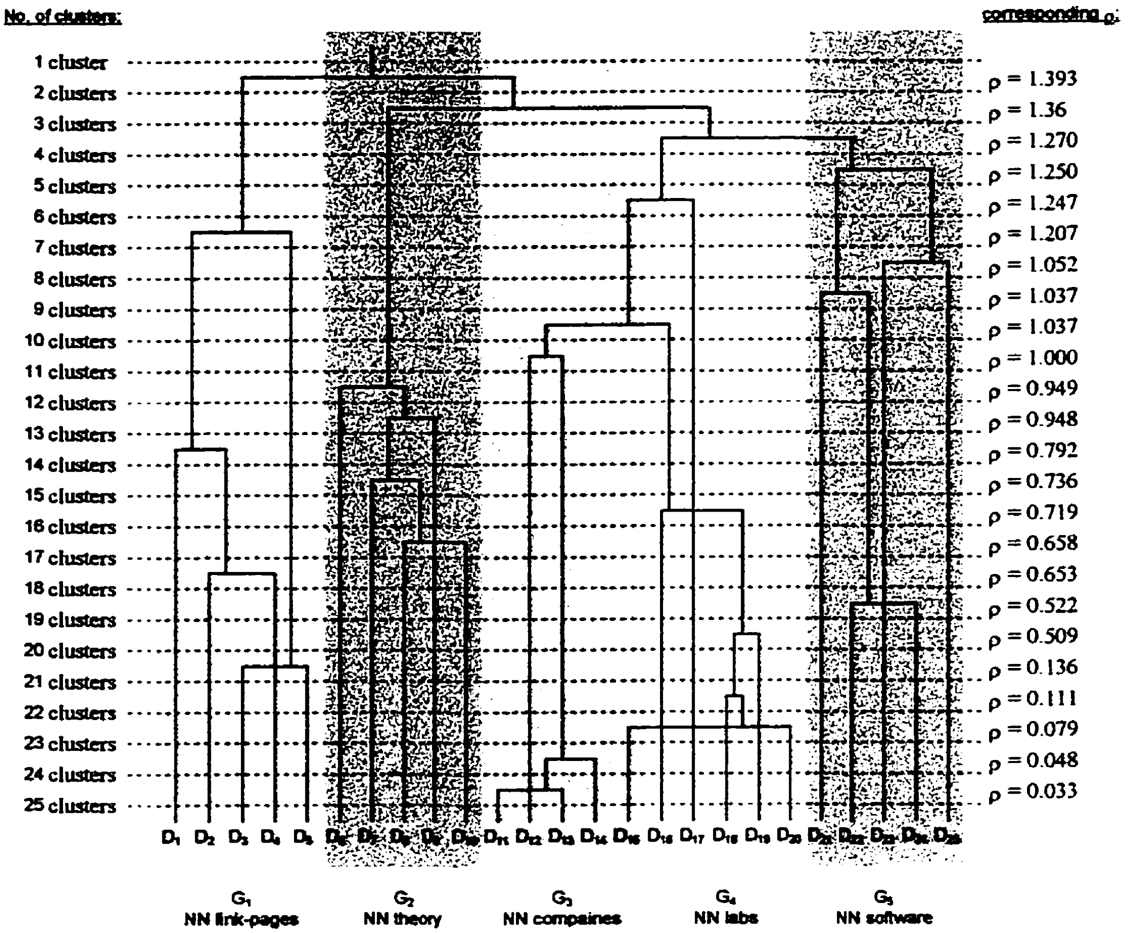


Figure 4.7 Dendogram obtained for hyper-dimensions based clustering on the collection from Table 4.2

4.4 Adaptive Hypertext Clustering

4.4.1 Background

The experiments presented in sections 2.4, 3.6.2, and 4.3.2 have shown the following:

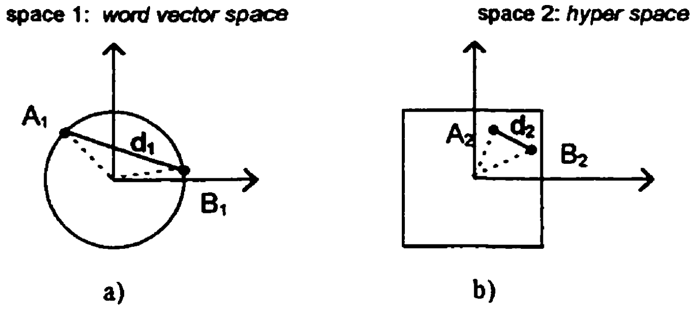
- Text (content) based Web page clustering provides satisfactory results in two main cases. (i) a collection is of relatively small size and its documents belong to different thematic domains, or (ii) a collection is of relatively large size and the interterm statistics gathered from the documents suggest semantically close or distant concepts within the same thematic domain.
- Hyper-dimension based clustering provides satisfactory performance if documents belong to the same thematic domain, but to different functional categories.

However, in a number of cases neither of these two techniques is appropriate. In general, Web collections may contain documents on different topics, and the documents on each topic may belong to different functional categories. This implies that an algorithm for universal Web page clustering may have to incorporate both textual and hyper information.

The principal problem of combined hyper-text clustering is regarding vector representation of Web pages. In particular, Web documents may be of various lengths (sizes), and in order to enable their comparison the text-related dimensions (word vectors) they are required to be normalized according to (4.9).

$$Xnormalized_i = \frac{x_i}{\sqrt{x_1 + x_2 + \ldots + x_n}} \qquad (4.9)$$

Word vector normalization actually means that only the relative information on the content of a Web page is preserved. Accordingly, the numerical value of each word vector dimension describes the percentage to which the corresponding concept is present in the document. If the presence of one concept is significant it may imply that the presence of all the other concepts is negligible. From a mathematical point of view, normalization based on (4.9) means that all word vectors lie on the unit hypersphere in n-dimensional word vector space, as depicted in Figure 4.8 a).



A_i - representation of document A in space i

B_i - representation of document B in space i

d_i - distance between A and B in space i

Figure 4.8 Web page representation in word vector and in hyper space

In contrast to text-related dimensions, hyper dimensions (section 4.3.1) can not be normalized in the same manner, since most of these are not mutually dependent or related. For example, if a Web page has significant depth weight it does not imply that its outward link density weight is insignificant, etc. However, hyper dimensions require a different type of normalization, as explained earlier, in order to maintain their numerical values within the limits 0 to 1. Consequently, hyper vectors may take any position within the unit hypercube in the 12-dimensional hyper space, as illustrated in Figure 4.8 b).

It is apparent that due to the different nature of word vector and hyper dimensions it would be inappropriate to simply combine them in a unique hyper-text space. Therefore, a Web page clustering algorithm intended to utilize all available information has to incorporate simultaneous performances in separate word vector and hyper space. In addition, the overall distance between two Web pages has to be defined as a compound function of the distances in the two spaces.

The modified ART2 algorithm provides perfectly stable learning (clustering) for a sufficiently small value of the dynamic parameter $\Delta\rho$ (section 3.6.1). Stable learning means that for a given level of clustering tolerance two items (clusters) will merge into a larger cluster only if the level of their mismatch is less than ρ, and for any higher level of ρ the two items will remain within the same cluster. This property makes modified ART2 very convenient for the problem of multi-space clustering. In particular, an algorithm can ensure stable multi-space clustering only if it guarantees stable clustering within each individual space.

The following section describes the adaptive hypertext clustering (AHC) algorithm, which is based on modified ART2. A significant property of the AHC algorithm is its ability to switch between text-based and hyper dimension - based clustering. The algorithm is proven to provide very satisfactory results regarding the clustering of Web documents on various topics and in various functional categories.

4.4.2 An Overview of the Adaptive Hypertext Clustering (AHC) Algorithm

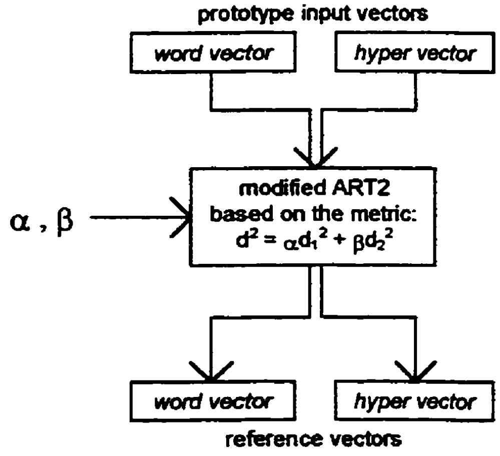


Figure 4.9 System for adaptive hypertext clustering

The adaptive hypertext clustering algorithm may be defined as modified ART2 adjusted to the problem of multi-space vector representation (Figure 4.9).

It can be observed form Figure 4.9 that the main difference between the standard modified ART2 and AHC is in their respective measures for the distance between training vectors. While the standard modified ART2 uses a Euclidean metric, the AHC algorithm employs the formula given in (4.9) as its distance measure.

$$d^2 = \alpha d_1{}^2 + \beta d_2{}^2 \tag{4.10}$$

d is the overall distance between two data points according to the AHC algorithm, d_1 is the Euclidean distance calculated in one (word vector) space, and d_2 in the other (hyper) space.

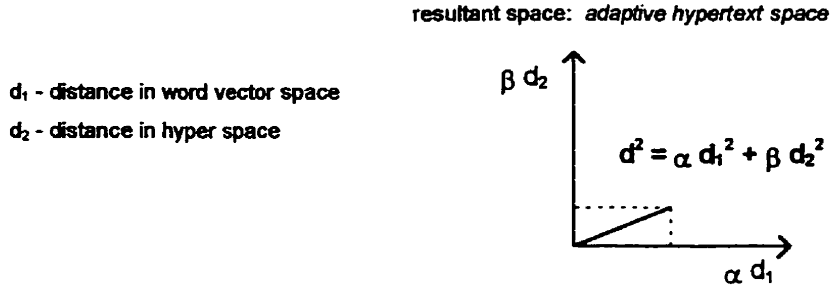

resultant space: adaptive hypertext space
d₁ - distance in word vector space
d₂ - distance in hyper space
β d₂
d² = α d₁² + β d₂²
α d₁

Figure 4.10 Distance measure employed by AHC

Recalling the procedure of standard ART2 (section 3.6.1), and particularly step 5 of its learning routine, it has to be emphasised that AHC applies (3.16) separately to word and hyper parts of the reference vector. In other words, whenever two prototype input vectors are found to be sufficiently close to each other according to (4.9), the merging will occur in word vector and hyper space independently from each other.

Parameters α, $\beta \in [0,1]$ are adjustable, and they determine the nature of clustering. For example, if $\alpha = 1$ and $\beta = 0$ then AHC produces pure text-based clustering, and Web pages are categorized according to their content. On the other hand, if $\alpha = 0$ and $\beta = 1$ the resultant clustering depends exclusively on hyper dimensions, and each cluster formed represents a distinguishable functional category. However, if α and $\beta \neq 0$, both word vector and hyper dimensions influence the grouping of Web pages. In that case the ratio α/β determines if textual or structural information is more decisive.

It has been shown through a number of examples that the α/β ratio should be relatively large when AHC is applied to a collection with a significant percentage of documents that originate from different thematic domains. It also should be large if a collection contains documents on the same topic and most of them belong to the same functional category. On the other hand, if many documents belong to the same thematic domain but to different functional categories the α/β ratio should be small in order to provide sensible clustering.

4.4.3 Experimental Results

Experiment 1

The first experiment performed involved the data collection of 25 Web documents presented in Table 4.3. The corresponding URLs are given in Appendix 11.

topic (category)	number of documents
tennis	5
jazz	5
NN index-pages	5
NN theory	5
NN companies	5

Table 4.3

It is apparent that the collection contained Web documents on three main topics: tennis, jazz, and neural networks. However, the group on neural networks consisted of three subcategories: NN index (link) pages, NN theoretical pages, and home pages of companies dealing with NN.

The first twenty-five words with the highest discrimination power according to the modified TF/IDF model, and thereby the word vector dimensions in this case were:

tenni, neural, jazz, icon, byte, network, worldwid, research, tabl, genova,
learn, product, artifici, group, lab, recognit, confer, system, educ, link,
dib, download, theori, univers, scienc

Figure 4.11 presents the clustering results obtained with the AHC algorithm for α=1, β=0. It is apparent that pure text-based clustering provided perfect separation among the main thematic categories. However, it failed to accurately recognize the subgroups within the group on neural networks.

The clustering obtained with AHC for α=0, β=1 is presented in Figure 4.12. The hyper dimension based clustering was more successful in discriminating among the subgroups of the group on NN compared to the previous case. On the other hand, for the given values of α and β, the algorithm failed to recognize the main thematic categories.

The most accurate clustering results were obtained for α=1.0, β=0.776 (Figure 4.13). This time, combining the information on the content and functionality of Web pages, the algorithm properly identified both the main groups and the subgroups.

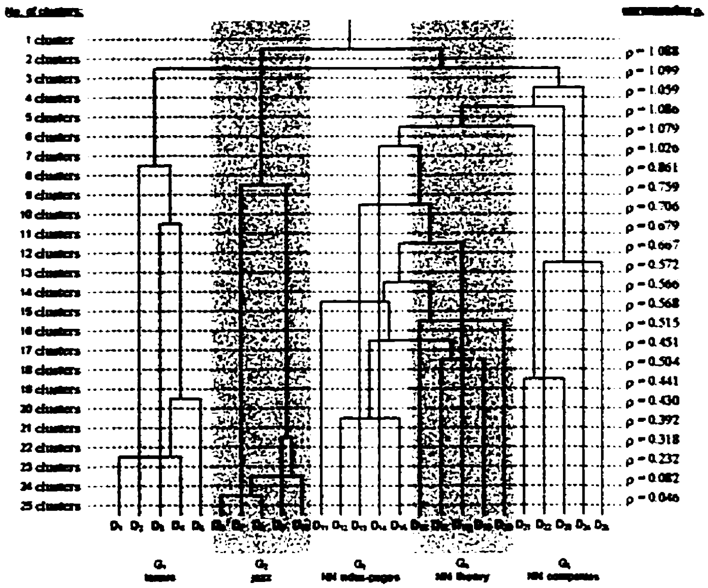


Figure 4.11 Dendogram obtained with AHC (α=1, β=0)
on the collection from Table 4.3

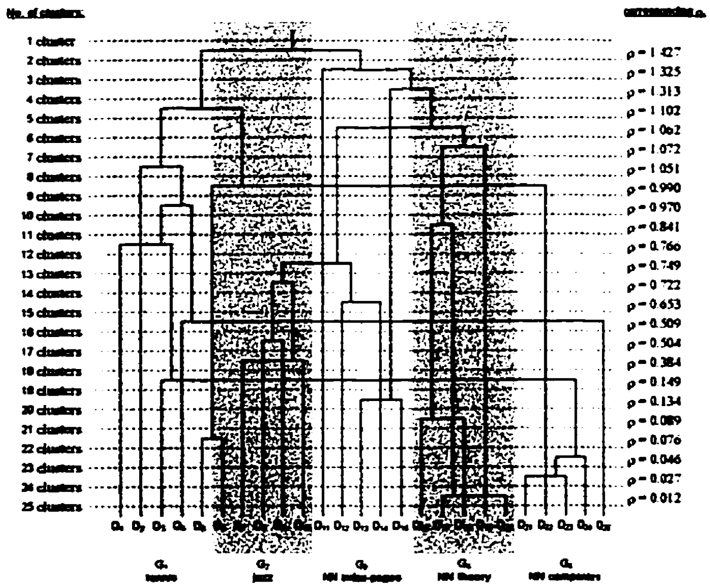


Figure 4.12 Dendogram obtained with AHC (α=0, β=1)
on the collection from Table 4.3

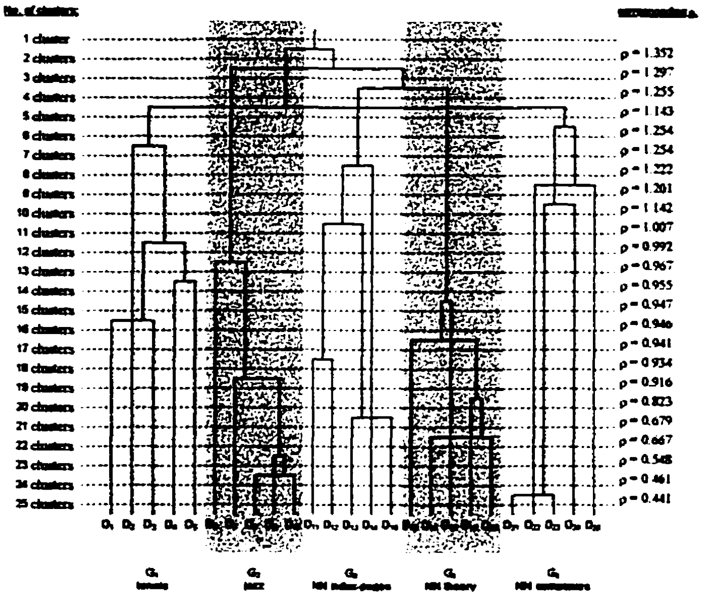


Figure 4.13 Dendogram obtained with AHC (α=1.0, β=0.776) on the collection from Table 4.3

Experiment 2

The second experiment concerned the data collection presented in Table 4.4. The corresponding URLs are given in Appendix 12. The first and last two groups in this case were thematically identical to the corresponding groups from Experiment 1. However, the third group was replaced with the group on volleyball. Therefore, the collection consisted of documents on four different topics (tennis, jazz, volleyball, and neural networks), and the group on neural networks consisted of two functional subcategories. The purpose of the experiment was to investigate how a variation in the number of distinguishable thematic domains would effect the ideal α/β ratio.

topic (category)	number of documents
tennis	5
jazz	5
volleyball	5
NN theory	5
NN companies	5

Table 4.4

Words with the highest discrimination power in this case were:

tenni, http, jazz, volleybal, icon, byte, ball, neural, network, magazin, artifici, donwload, worldwide, educ, gam, product, speech, technologi world, search, neuron, org, sub, australian, musicianship

According to Figures 4.14, 4.15 and 4.16 it is evident that the AHC algorithm for α=1.0, β=0.6 produced the best results. It should be noticed that the parameter β was of lower value as compared to the previous case. This implies that the increase in the number of different topics required more emphasis to be placed on text (content) related information in order to obtain a satisfactory categorization.

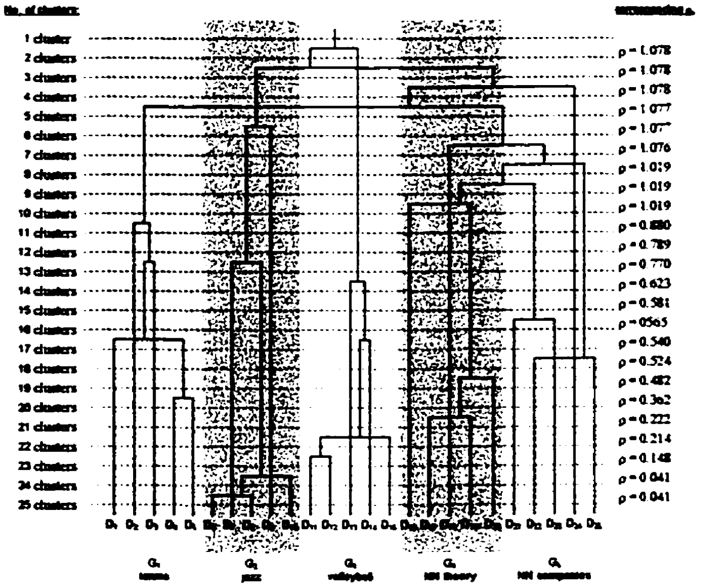


Figure 4.14 Dendogram obtained with AHC (α=1, β=0)
on the collection from Table 4.4

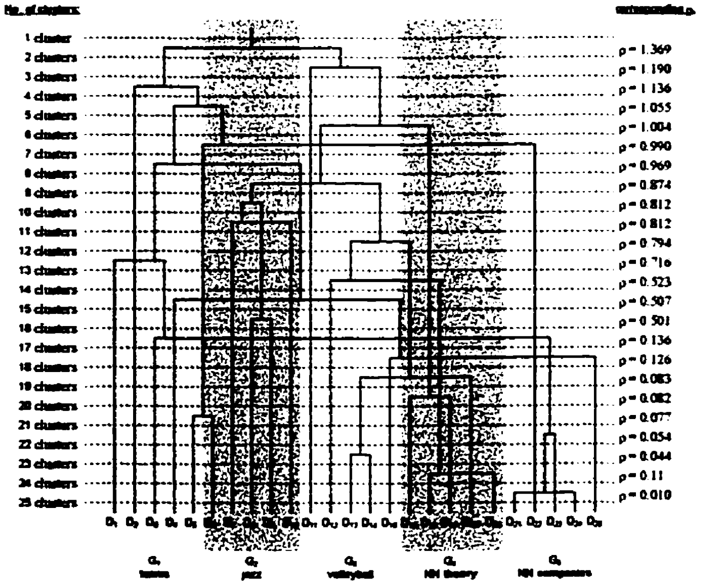


Figure 4.15 Dendogram obtained with AHC (α=0, β=1)
on the collection from Table 4.4

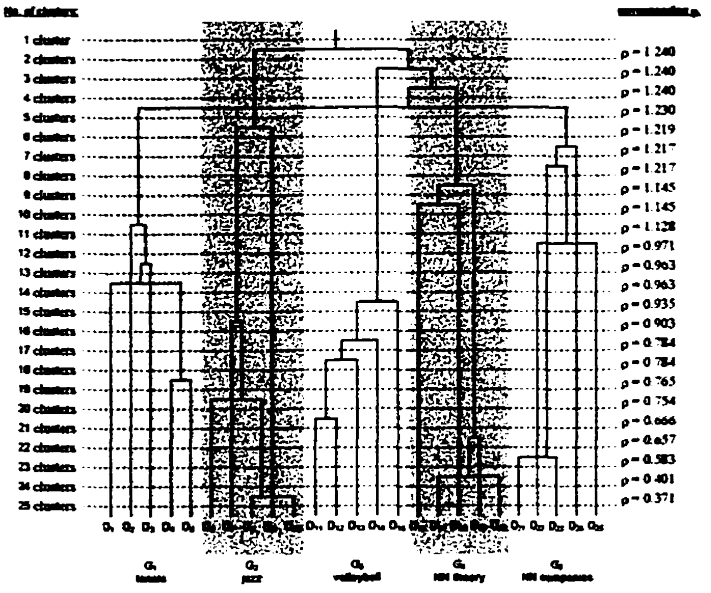


Figure 4.16 Dendogram obtained with AHC (α=1.0, β=0.6) on the collection from Table 4.4

Experiment 3

In the third experiment all documents were on the same main topic (neural networks), but of different functionality (Table 4.6). In particular, there were five groups: NN companies, NN links, NN software, theory on fuzzy-NN, and theory on reinforcement learning. However, the five groups belonged to four structural categories, since the last two consisted of documents that were of *content page* type. The corresponding URLs are given in Appendix 13. The purpose of the experiment was to test how the AHC algorithm would cope with the problem of a Web page collection from the same thematic domain, and partially from the same functional category.

topic (category)	number of documents
NN companies	5
NN link-pages	5
NN software	5
theory on fuzzy-NN	5
theory on reinforcement learning	5

Table 4.5

The word vector space created for the collection from Table 4.5 consisted of the following stems:

fuzzi, byte, reinforc, java, download, instal, applet, learn, demo, network, perceptron, system, product, confer, neurosolut, multi-lay, group, content, neural, algorithm, recognit, comput, hom, html, link

According to Figures 4.17 and 4.18, neither the pure text-based nor the hyper dimension - based clustering provided satisfactory results. The text-based clustering accurately recognized only the two groups that consisted of documents of particular content (*theory on fuzzy-NN* and *theory on reinforcement learning*). On the other hand, the clustering based on hyper dimensions was better at recognizing the main functional categories. Evidently the clustering presented in Figure 4.19, obtained for α=1.0, β=0.7, was the most accurate.

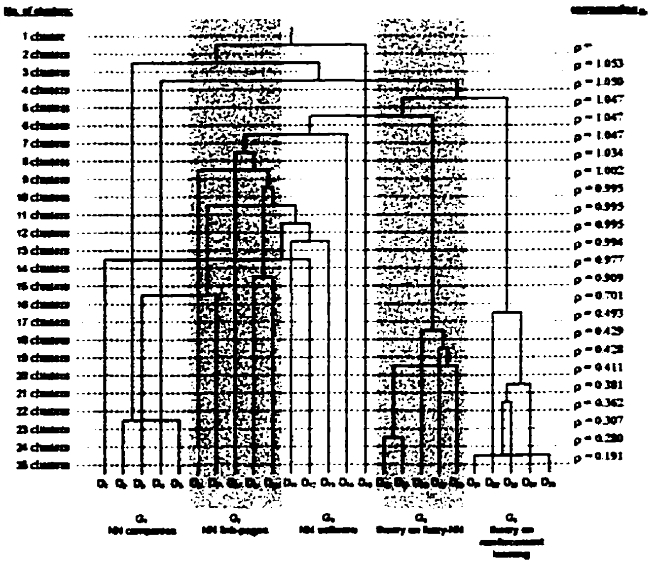


Figure 4.17 Dendogram obtained with AHC (α=1.0, β=0.0) on the collection from Table 4.5

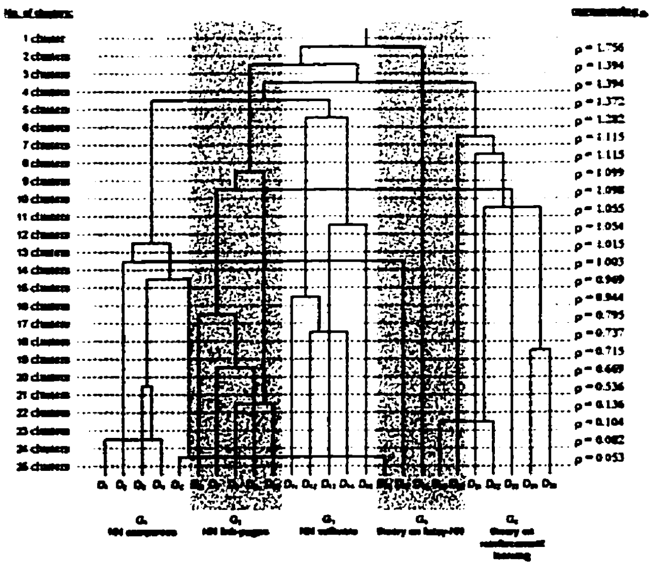


Figure 4.18 Dendogram obtained with AHC (α=0.0, β=1.0) on the collection from Table 4.5

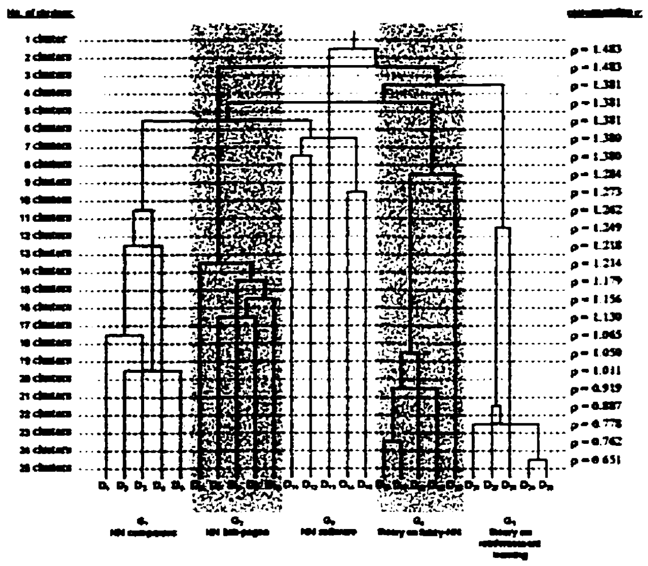


Figure 4.19 Dendogram obtained with AHC (α=1.0, β=0.7) on the collection from Table 4.5

3.5 Conclusions

The WWW is an on-line hypertextual collection, and text-only based clustering techniques when applied to Web pages may not always produce satisfactory results. It is observed that some valuable information on the importance and functionality of a Web page can be extracted from its hyper structure: length, number of links, etc... The results obtained by experimenting with the adaptive hypertext algorithm have shown that, combining hyper and text similarity measures, it is possible to provide for significantly improved categorization in a number of situations. However, it has also been shown that the complexity of a collection determines the actual importance of text and hyper related data for Web page clustering purposes. Thus, if a collection consists of documents on various topics, or if most of documents belong to the same functional category, content based information should be dominant. On the other hand, clustering within a collection that consists of documents from the same thematic domain may require the domination of hyper over textual information. At the present stage the AHC algorithm allows the user to control the relative contribution of content and hyper similarity measures, and thereby to determine the nature of the clustering. However, further work regarding AHC may incorporate methods for automatic discovery of the most convenient mode of operation, in order to obtain the best possible results.

References

[4.1] Robert Orfali, Dan Harkey, Jeri Edwards 1996. *The Essential Client/Server Survival Guide*. New York, NY: Wiley Computer Publishing.

[4.2] Weiss R., Velez B., Sheldon M. A., Namprempre C., Szilagyi P., Duda A., Gifford D. K. 1996. *HyPursuit: A Hierarchical Network Search Engine that Exploits Content-Link Hypertext Clustering*. In Proceedings of Hypertext '96, Washington DC, USA, March 16-20, 1996.
http://www.psrg.lcs.mit.edu/ftpdir/papers/hypertext96.ps

[4.3] Chaomei Chen 1997. *Structuring and Visualising the WWW by Generalised Similarity Analysis*. Proceedings of the Eighth ACM Conference on Hypertext: Hypertext 97, Southampton, UK, 1997.
http://www.brunel.ac.uk/~cssrccc2/papers/ht97.pdf

[4.4] Massimo Marchiori 1996. *The Quest for Correct Information on the Web: Hyper Search Engines*. In Hyperproceedings of WWW6 Conference, Santa Clara, California, USA, April 7-11, 1997.
http://www6.nttlabs.com/HyperNews/get/PAPER222.html

[4.5] Bryan Pfaffenberger 1997. *The Elements of Hypertext Style*. Boston: AP Professional.

[4.6] Ian S. Graham 1997. *HTML Sourcebook: A complete Guide to HTML 3.2 and HTML Extensions*. New York: Wiley Computer Publishing.

[4.7] Andrew Casson 1994. *PLINTH, HTML and the World Wide Web*. Artificial Intelligence Applications Institute, University of Edinburgh.
http://www.aiai.ed.ac.uk/~andrewc/plinth/www/contents.html

[4.8] Nikos Drakos 1994. *From Text to Hypertext: A Post-Hoc Rationalisation of LaTeX2HTML*. Computer Based Learning Unit, University of Leeds.
http://cbl.leeds.ac.uk/~nikos/doc/www94/www94.html

CHAPTER 5

Software

5.1 The Software Implementation

The methods described in this thesis have been entirely embedded in an autonomous agent, implemented in Java, and built for the purpose of Web page classification employing neural networks. For a given collection of Web pages, the agent is able to perform text-based (Figure 5.1), hyper dimension - based (Figure 5.2), and combined hyper-text clustering (Figure 5.3). The first two types of clustering may be considered as the special cases of the final system.

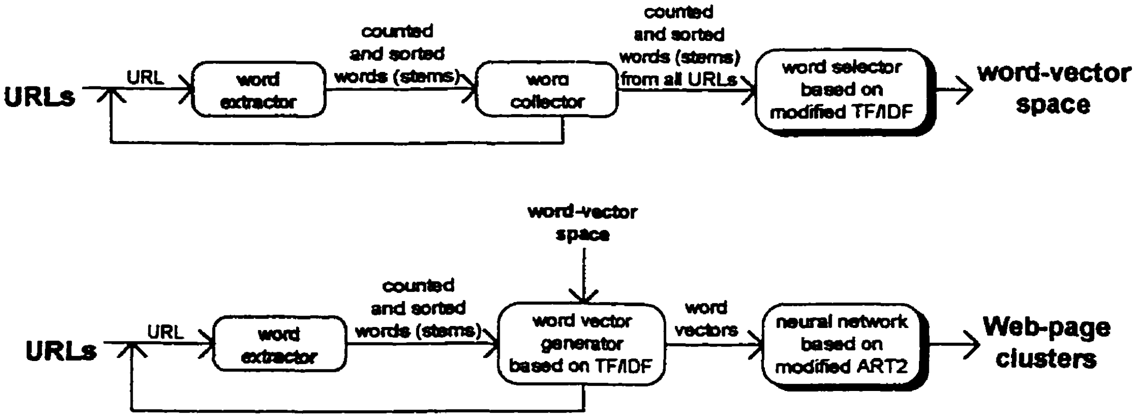

URLs
URL
word extractor
counted and sorted words (stems)
word collector
counted and sorted words (stems) from all URLs
word selector based on modified TF/IDF
word-vector space
word-vector space
URLs
URL
word extractor
counted and sorted words (stems)
word vector generator based on TF/IDF
word vectors
neural network based on modified ART2
Web-page clusters

Figure 5.1 System for text-based clustering

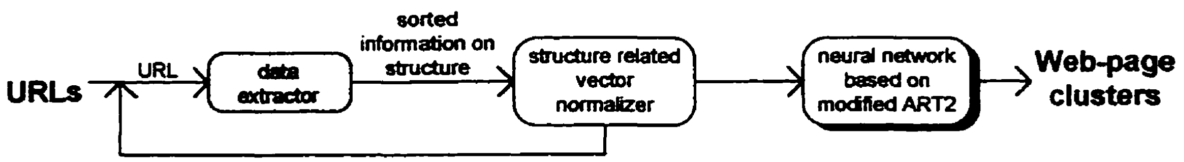

URLs
URL
data extractor
sorted information on structure
structure related vector normalizer
neural network based on modified ART2
Web-page clusters

Figure 5.2 System for hyper dimension -based clustering

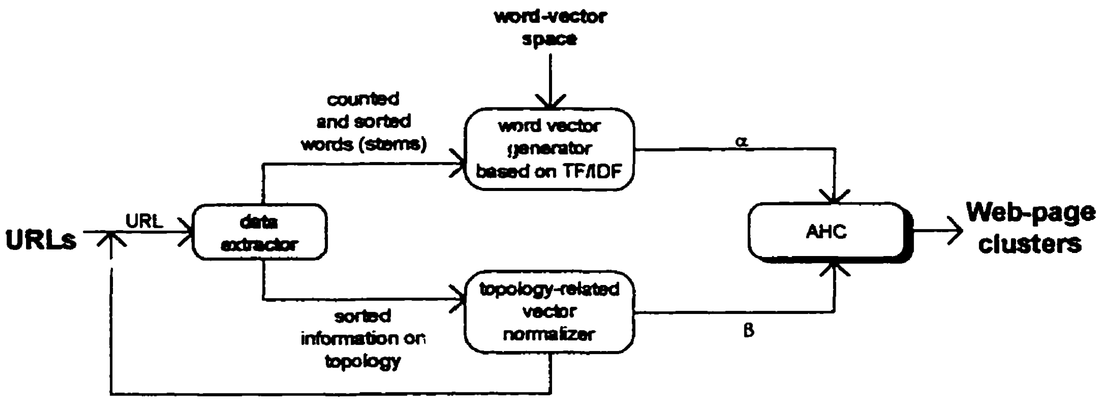

word-vector space
counted and sorted words (stems)
word vector generator based on TF/IDF
α
URLs
URL
data extractor
AHC
Web-page clusters
topology-related vector normalizer
sorted information on topology
B

Figure 5.3 System for adaptive hyper-text clustering

The program begins with a list of URLs. For each URL, the agent retrieves its HTML source code, recognizes if the Web page uses frames, in which case it detects URLs for the frames and retrieves them replacing the original HTML source code, then separates the text from HTML tags. The agent further parses both the text and the HTML tags in order to extract information on the content and hyper structure of the corresponding Web page. (Figure 5.4)

The extraction of text related data is obtained by deleting non-letters around words, eliminating stop words, and performing stemming operations. Furthermore, attempting to extract as much textual information as possible, the agent enters into the HTML portions and looks for ALT tags, i.e. words that ALT refers to. This partly solves the problem of not being able to read images. Also, it gives larger weights, ranging from 1 to 5, to the words that appear in special HTML markups such as *title*, *h1*, *h2*, *a*,.... HyPursuit [4.2] recommends those weights being larger, ranging from 1 to 10. This has been shown however to give misleading results, especially in short documents, where words from the ordinary text portions do not have a chance to achieve a comparable TF. The agent creates a vector of all identified words or stems, calculating their normalized term frequencies TF, and adds those results to the vectors that store stems, corresponding TF values, and the numbers of documents containing stems for the entire collection.

In order to extract hyper structure related data the agent performs a number of operations. For example, based on the URL of a given page the agent determines its depth in the file system hierarchy. Furthermore, it looks for markups such as *img*, and *a* within HTML tags, in order to count images and links. For each link the agent analyses its URL, and recognizes if it points to a Web page placed on the same site or elsewhere. For links which point to the same site, the agent examines if they are of upward, downward or crosswise type. Also, it counts e-mail and news links, links with extensions other than .html or .htm, and links to other protocols. Finally, the agent recognizes text placed under links.

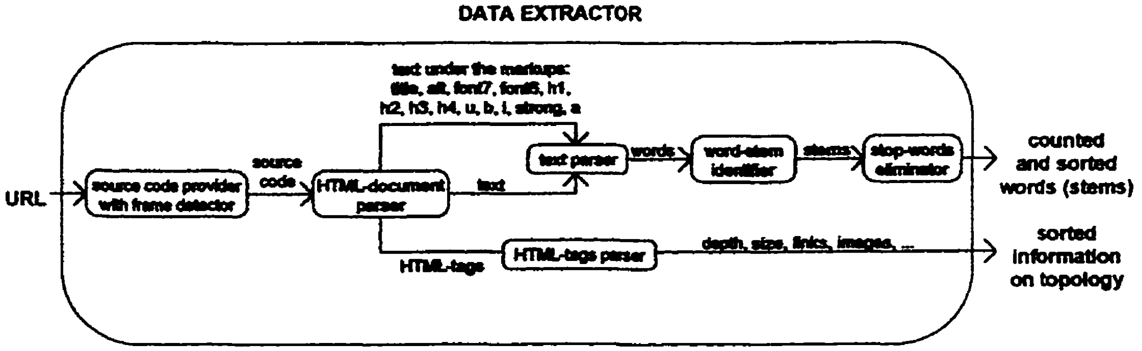

DATA EXTRACTOR
URL
source code provider with frame detector
source code
HTML-document parser
text under the markups: title, alt, font7, font6, h1, h2, h3, h4, u, b, i, strong, a
text
text parser
words
word-stem identifier
stems
stop-words eliminator
counted and sorted words (stems)
HTML-tags
HTML-tags parser
depth, size, links, images, ...
sorted information on topology

Figure 5.4 Data extraction from a Web page

If the user requires text-based clustering (Figure 5.1), after all pages have been visited, the agent calculates inverse document frequencies for all identified words, and makes a new list based on the modified TF/IDF of Eqn (2.2). Taking a number of the most highly ranked terms, it forms a term-vector space. Then the agent revisits the pages, calculates their normalized word vector representations for the given space using the standard TF/IDF model, and finally classifies the vectors employing a neural network based on modified ART2.

On the other hand, if hyper dimension - based clustering is required (5.2), after all pages have been visited the agent normalizes the corresponding vectors, and again performs a clustering based on modified ART2.

However, if the user is interested in adaptive hypertext clustering (5.3), then combining the above mentioned procedures the agent calculates normalized word vector and hyper dimension - based vectors, and classifies them using the modified ART2 algorithm adapted for multi-space clustering.

CHAPTER 6

Conclusions and Recommendations

6.1 Conclusions

In this thesis we have presented several techniques intended to provide for rapid and appropriate page classification on the Web. The techniques are not limited by the accessibility of input data, and could be equally useful when applied to larger collections, including on-line catalogs and search engines, or to smaller ones, such as bookmark sets. Some of our methods are text (content) related, and improve the selection of the most salient words for a given collection. Moreover, by determining term and inter-term statistics (correlations) they enable the compression of words that are irrelevant or less salient, and lead to subsequent analysis of the corresponding group of documents in a lower dimensional term-vector space. Lower dimensionality implies substantially reduced computation requirements for a given set of document vectors. Another significant improvement regarding Web page classification has been obtained by employing the modified adaptive resonance (ART2) algorithm. In contrast to other unsupervised ANN learning methods, including the SOM algorithm and hard competitive learning, modified ART2 is proven to successfully deal with nonstationary collections of non-uniform document distributions. Moreover, it provides stable hierarchical clustering, and therefore is very convenient for highly efficient multi-level document retrieval. The main limitation of the ART2 algorithm is the absence of information on input data topology. However, this limitation has been overcome by combining modified ART2 with competitive Hebbian learning. Moreover, modified ART2 with CHL is shown to be very useful in the discovery of related or relevant groups of Web documents. Finally, in our research we have developed an algorithm for adaptive hypertext clustering that could be universally employed. In other words, this algorithm uses only the data that is retrievable from HTML source codes and, in contrast to the existing techniques for hypertext clustering, does not require the knowledge of publicly unavailable information. A significant property of the AHC algorithm is its ability to switch between text-based and hyper dimension - based clustering. Therefore, the algorithm can be easily adapted to provide the most appropriate Web page classification within collections of various thematic and functional profiles.

6.2 Recommendations for Future Work

The techniques presented in this thesis were proven, through a number of experiments, to provide satisfactory results. However, there are still some segments which could be improved. For example, at the present stage all the steps, from Web page retrieval to hypertext clustering, are automatic, except thesaurus creation for the purpose of word vector space dimensionality reduction (Chapter 2). Although thesaurus creation slows down the overall performance, the improvements obtained verify its importance, and justify the presence of human intervention. One way to improve the algorithm would be to make this step completely automatic, and yet comparable to the case when human assistance is involved.

Another issue which should be addressed is regarding the AHC algorithm. It has already been mentioned that AHC allows the user to control the contribution of content and hyper similarity measures, and thereby to determine the nature of clustering. However, if the user does not make a proper assumption about the relative importance of text and hyper related information for clustering purposes, the classification may not be satisfactory. Therefore, further work regarding AHC could incorporate methods for automatic discovery of the most convenient mode of operation. There are indications that the discrimination power of individual word vector dimensions determines the discrimination power of the overall word vector space. If the space is of low discrimination power, that could be an indicator that for the given case more emphasis should be placed on the hyper-dimension based similarity measure.

It has been stressed that in the thematic domain of neural networks, upon which we mainly focused, documents from the functional category known as *pages of special content* primarily contain links to files with extensions: *.ps*, *.zip*, *.java* , and *.exe*. However, pages of special content within other thematic domains could contain, for example, links to audio or video files. Therefore, more general application of the algorithm requires certain modifications in this area.

APPENDIX 1

URLs of 218 Web Pages on Neural Networks

1	http://www.nd.com/new/index.htm
2	http://www.neural.com/pressreleases/1997/April/Microsoft.html
3	http://www.zsolutions.com/backpack.htm
4	http://ourworld.compuserve.com/homepages/attg/attgprdf.html
5	http://www.irnet.com/pages/press/050597merger.htm
6	http://www.neural.com/NetProphet/NetProphet.html
7	http://www.bgif.no/neureka/about.html
8	http://www.neuralware.com/_private/company.htm
9	http://www.camneuro.stjohns.co.uk/frameadt.htm
10	http://www.neural.com/pressreleases/1996/Apr/NetProphet.html
11	http://www.shef.ac.uk/~signalbox/page3.htm
12	http://www.globalweb.co.uk/gwl/
13	http://ProfitTaker.com/technical_mendelsohn_small.htm
14	http://www.ppgsoft.com/fuzneu.html
15	http://www.neuralfusion.com/order.html
16	http://www.inc.com/incmagazine/archives/18961091.html
17	http://www.alliedtech.com/sbir/ss/88text.html
18	http://www.biocompsystems.com/
19	http://www.biocompsystems.com/pages/financia.htm
20	http://www.fonix.com/corporate.html
21	http://www.cam.org/~fsalvat/
22	http://www.ann-pro.com/
23	http://www.progno.com/neural.htm
24	http://www.wardsystems.com/consum.htm
25	http://promland.com/
26	http://www.bioele.nuee.nagoya-u.ac.jp/wsc1/papers/p031.html
27	http://carlassoc.com/
28	http://www.pfr.com/index.html
29	http://www.neurotec.com/techno/nn/nn5.html
30	http://ourworld.compuserve.com/homepages/attg/attgnrlt.html
31	http://rfhs8012.fh-regensburg.de/~sauer/jfroehl/diplom/e-g.html#FeedforwardType
32	http://www.neuralt.com/Whitepaper/NTL_theory.htm
33	http://www.msci.memphis.edu/~garzonm/nnets/nnpubs.html
34	http://www.calsci.com/whatare.htm
35	http://rfhs8012.fh-regensburg.de/~sauer/jfroehl/diplom/e-11-text.html
36	http://www.cs.utexas.edu/users/nn/web-pubs/htmlbook96/wiskott/node3.html#figrunBlobsarch
37	http://www.cs.utexas.edu/users/nn/web-pubs/htmlbook96/dong/node3.html#SECTION00021000000000000000
38	http://robotics.eecs.berkeley.edu/MURI/muriproject7.html
39	http://rfhs8012.fh-regensburg.de/~sauer/jfroehl/diplom/e-13-text.html
40	http://www-dse.doc.ic.ac.uk/~nd/surprise_96/journal/vol2/cs11/article2.html
41	http://www.dacs.dtic.mil/techs/neural/neural10.html
42	http://www.cs.iastate.edu/~cs474/spring97/lecturenotes.html
43	http://www.cs.utk.edu/~mclennan/Classes/594-MNN/
44	http://www.shef.ac.uk/psychology/gurney/notes/index.html
45	http://www.eng.fsu.edu/feeds/foo.html
46	http://adam.fiz.huji.ac.il/orens/NNB.html
47	http://www.abo.fi/~abulsari/courses.html
48	http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/DemoGNG/GNG.html
49	http://psy.uq.edu.au/~mav/java/Necker.html

50 http://www.aist.go.jp/ETL/etl/suri/motomura/BN/bn-java.html
51 http://suhep.phy.syr.edu/courses/modules/MM/SIM/Hopfield/
52 http://home.cc.umanitoba.ca/~umcorbe9/anns.html
53 http://rfhs8012.fh-regensburg.de/~saj39122/jfroehl/diplom/e-index.html
54 http://www.cl.ais.net/uthed/neural.htm
55 http://www.informatik.uni-stuttgart.de/ipvr/bv/pns/
56 http://www.aist.go.jp/ETL/~akaho/MixtureEM.html
57 http://www.netkonect.net/n/neural/nd_contents.html
58 http://www.nada.kth.se/nada/sans/compneuro.html
59 http://www.ai.univie.ac.at/oefai/nn/tool.html
60 http://www.ifsys.co.uk/Showcase1/index.html
61 http://www.mathworks.com/products/neuralnet/
62 http://www.aist.go.jp/NIBH/~b0616/Lab/BSOM1/index.html
63 http://www.informatik.uni-stuttgart.de/ipvr/bv/projekte/snns/announce.html
64 http://neuron.duke.edu/
65 http://cns-web.bu.edu/pub/laliden/WWW/nnet.frame.html
66 http://www.mcs.com/~drt/svbp.html#nearest
67 http://www.rpl.com/features.html
68 http://www.industry.net/c/mn/_swnet/
69 http://polimage.polito.it/groups/daniela/daniela.html
70 http://diwww.epfl.ch/w3mantra/mantra_machine.html
71 http://www.eln.utovrm.it/~cirlab/cnn_chip.html#6x6DPCNN
72 http://www.webcom.com/~ictech/nram.html
73 http://www.ece.uc.edu/~ansl/hardware.html
74 http://www.dhp.nl/~heiniw/thesis/
75 http://cns-web.bu.edu/muri/year1-report/4f.html
76 http://msia02.msi.se/~lindsey/elba2html/section3_5.html
77 http://msia02.msi.se/~lindsey/elba2html/subsubsection3_3_1_4.html#SECTION00031400000000000000
78 http://www.cbu.edu/~pong/624lsb2.htm
79 http://www.ibm.fr/france/cdlab/zicope.htm
80 http://www1.physik.unibas.ch/~leimgrub/nntrack/node5.html#SECTION00050000000000000000
81 http://petrus.upc.es/~microele/neuronal/US/projectes.html
82 http://www.amazon.com/exec/obidos/ISBN%3D0879422890/5019-1394227-027506
83 http://www.emsl.pnl.gov:2080/proj/neuron/papers/keller.aist94.html
84 http://www.emsl.pnl.gov:2080/proj/neuron/papers/keller.wcnn95.html
85 http://www.nada.kth.se/nada/sans/walking.html
86 http://www.ai.univie.ac.at/oefai/nn/ctg-cu96.html
87 http://ic-www.arc.nasa.gov/ic/projects/neuro/sc_nav/sc_nav.html
88 http://www.brunel.ac.uk/~hssrjis/issue/j714.html
89 http://accurate-automation.com/projects/ncrobot.htm#Robotics
90 http://www.emsl.pnl.gov:2080/proj/neuron/briefs/cardio.html
91 http://web.egr.msu.edu/CSANNLAB/bss.html
92 http://diwww.epfl.ch/lami/team/cornu/ametis2/ametis2.html
93 http://www.sensoryInc.com/neural.shtml
94 http://www.cs.cmu.edu/afs/cs.cmu.edu/project/alv/member/www/projects/ALVINN.html
95 http://www.ai.univie.ac.at/oefai/nn/neufodi.html
96 http://tag-www.larc.nasa.gov/tops/tops95/exhibits/emt/emt-174-95/emt17495.html
97 http://www.emsl.pnl.gov:2080/docs/cie/neural/newsgroups.html
98 http://ciips.ee.uwa.edu.au/Papers/Conference_Papers/title.html
99 http://www.lehigh.edu/~ay00/nn.html
100 http://www.ai.univie.ac.at/oefai/nn/conn_biblio.html
101 http://www.loria.fr/exterieur/equipe/rfia/cortex/servers.html
102 http://jnns-www.okabe.rcast.u-tokyo.ac.jp/jnns/home.html
103 http://cns-web.bu.edu/inns/
104 http://www.ewh.ieee.org/tc/nnc/
105 http://www-dsi.ing.unifi.it/neural/siren/sirenEN.html
106 http://www.neuronet.ph.kcl.ac.uk/neuronet/organisations/enns.html
107 http://cscsi.sfu.ca/

108 http://www.wins.uva.nl/research/neuro/ias-ras/ias.html
109 http://http.hq.eso.org/~fmurtagh/clustering.html
110 http://cns-web.bu.edu/inns/nn/N103.html
111 http://www.msci.memphis.edu:80/~jagota/JANN/index.html
112 http://www.cs.cmu.edu/afs/cs/project/cnbc/nips/NIPS.html
113 http://www.cnel.ufl.edu:80/nnsp97/
114 http://www.cs.caltech.edu/~learn/nncm.html
115 http://www.emsl.pnl.gov:2080/proj/neuron/workshops/WEEANN95/
116 http://www.arc.unm.edu/wcci-98/ijcnn.html
117 http://diwww.epfl.ch/w3mantra/mn96.html
118 http://www.abo.fi/~abulsari/EANN96.html
119 http://www.lpac.ac.uk/SEL-HPC/Events/NeuralWorkshop/index.html
120 http://shay.ecn.purdue.edu/~csglee/icra-98/
121 http://www.clients.globalweb.co.uk/nctt/newsletter/01/
122 http://www.brunel.ac.uk/~hssrjis/issue/index.html
123 http://ourworld.compuserve.com/homepages/FTPub/jcif.htm
124 http://www.csu.edu.au/ci/aboutci.html
125 http://www-mitpress.mit.edu/journal-home.tcl?issn=08997667
126 http://neural-server.aston.ac.uk/NCAF/journal/index.html
127 http://www.eeb.ele.tue.nl/neural/contents/neural_computation.html
128 http://cns-web.bu.edu/inns/nn.html
129 http://www.oup-usa.org/acadref/honc.html
130 http://www.oup-usa.org/acadref/nc_toc.html
131 http://www.amazon.com/exec/obidos/ISBN=0201539217/5019-1394227-027506
132 http://www.ececs.uc.edu/~ansl/
133 http://polimage.polito.it/groups/nngroup.html
134 http://dibe.unige.it/neuronet/DIBE/SPUG.html
135 http://dibe.unige.it/department/ncas/group.html
136 http://www.cs.rulimburg.nl/~antal/nn.html
137 http://www.iic.umanitoba.ca/projects-hc.htm
138 http://www.msci.memphis.edu/~garzonm/#pubs
139 http://www.emsl.pnl.gov:2080/docs/cie/neural/
140 http://www.fit.qut.edu.au/~robert/rulex.html
141 http://www.iscs.nus.edu.sg/~leowwk/isl/nn.html
142 http://web.egr.msu.edu/CSANNLAB/ann.html
143 http://dsi.ing.unifi.it/neural/
144 http://www.cnl.salk.edu/CNL/
145 http://bbf-www.uia.ac.be/
146 http://www.eeb.ele.tue.nl/neural/neuron.html
147 http://www.voicenet.com/~rybak/resp.html
148 http://www.nnc.yale.edu/papers/NIPS94/nipsfin.html
149 http://www.nnc.yale.edu/papers/ebench/ebench.html
150 http://suhep.phy.syr.edu/MM/Biology/bio_org.html#work
151 http://www.ice.ge.cnr.it/Biology.html
152 http://www.uivt.cas.cz/ics/nn-biol.html
153 http://www.vxm.com/21R.7.html
154 http://www.cns.ed.ac.uk/people/mark/intro/node3.html#SECTION000300000000000000000
155 http://www.wspc.com.sg/journals/ijns/82/zhou.html
156 http://www.wspc.com.sg/journals/ijns/82/hung.html
157 http://www.cs.utexas.edu/users/nn/web-pubs/htmlbook96/sirosh/
158 http://ciips.ee.uwa.edu.au/Papers/Conference_Papers//1992/02//Index.html
159 http://websom.hut.fi/websom/
160 http://iserv.iki.kfki.hu/pub/icnn94/rozgonyi.scaling.abs.html
161 http://www.nd.com/models/sofm.htm
162 http://www.neuroinformatik.ruhr-uni-bochum.de/ini/PEOPLE/fritzke/research/new.html
163 http://www-karc.crl.go.jp/avis/maekawa/study.html
164 http://gepasi.dbs.aber.ac.uk/roy/koho/kohonen.htm
165 http://web.egr.msu.edu/CSANNLAB/pca.html

166 http://www.nd.com/models/pca.htm
167 http://i3a.dtic.ua.es:8080/local/neural-196/0320.html
168 http://www.neuroinformatik.ruhr-uni-bochum.de/icann96/IC96/PROGRAM/aapo@nucleus.html
169 http://ftpserver.nus.sg/docs/csbiblio/Neural/pca.html
170 http://www.cs.colostate.edu/~anderson/matlab-papers/rl/rl.html
171 http://www.cs.cmu.edu/Groups/reinforcement/web/homepage.html
172 http://www.neuralware.com/products/proplus/proii5.html
173 http://iserv.iki.kfki.hu/pub/icml96/szepes.genreinf.abs.html
174 http://www.oup-usa.org:9200/9205/abs/ncc3.htm
175 http://www.cs.cmu.edu/~reinf/web/cmu-rl-papers.html
176 http://Neuron.derby.ac.uk/neuron/resneuralnet.html
177 http://www2.shef.ac.uk/psychology/gurney/notes/l6/subsection3_1_2.html#SECTION00012000000000000000
178 http://iserv.iki.kfki.hu/pub/fomin.robhah.abs.html
179 http://www.nd.com/learning/hebbian.htm
180 http://www.santafe.edu/~jmerelo/neurolib/node5.html
181 http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/JavaPaper/node17.html
182 http://hebb.cis.uoguelph.ca/~deb/27642/LectureNotes/rbf/node2.html#SECTION00020000000000000000
183 http://www.cns.ed.ac.uk/people/mark/intro/node8.html
184 http://www.cns.ed.ac.uk/people/mark/intro/intro.html
185 http://i3a.dtic.ua.es:8080/local/neural-196/0335.html
186 http://www.oup-usa.org:9200/9205/abs/ncg2_6.htm
187 http://neural-server.aston.ac.uk/postdocs/on_line_learning.html
188 http://www.nd.com/models/rbf.htm
189 http://www.ncs.co.uk/
190 http://www.flextool.com/
191 http://cs.uregina.ca/~narate/Links/nn+fuzzy.html
192 http://www.forwiss.uni-erlangen.de/aknn/index.eng.html
193 http://peacock.pse.che.tohoku.ac.jp/~yyama/work/ann/section3.2.html
194 http://pcim.com/arc/conf/mexpo/11.html
195 http://www.tech.plym.ac.uk/soc/sameer/abstract/msc.htm
196 http://www.dreamlabs.com/~webland/c_p/icnn95/nn95s608.htm
197 http://i80s25.ira.uka.de/neurofuzzy/homepage.html
198 http://www.isd.uni-stuttgart.de/~rudolph/geneticalg/genetic_abstract_01.html
199 http://arti.vub.ac.be/www/brochure/section1.3.0.3.html
200 http://www.ntu.edu/1/atmp/1997Courses/mc97043001.htm
201 http://www.epcc.ed.ac.uk/tracs/tracs.seminars/abstracts/giulio.html
202 http://set.gmd.de/AS/nn/publi/nips-94.html
203 http://www.faqs.org/faqs/ai-faq/neural-nets/part3/section-7.html
204 http://www.cs.utoronto.ca/~radford/thesis.abstract.html
205 http://www-speech.sri.com/projects/hybrid.html
206 http://www-laforia.ibp.fr/CONNEX/Articles/artieres-eurospeech95.abs.txt
207 http://www-karc.crl.go.jp/avis/zhang/study.html
208 http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/JavaPaper/
209 http://www.nd.com/learning/compete.htm
210 http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/JavaPaper/
node9.html#SECTION00500000000000000000
211 http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/JavaPaper/
node15.html#SECTION00600000000000000000
212 http://www.cs.utoronto.ca/~zoubin/zoubin/linear-gaussian.abstract.html
213 http://www.wspc.com.sg/journals/ijns/81/firmin.html
214 http://www.nd.com/learning/backprop.htm
215 http://herens.idiap.ch/~perry/moerland-94.1.bib.abs.html
216 http://sunflower.singnet.com.sg/~wspclib/Books/compsci/3094.html
217 http://outside.gsfc.nasa.gov/ESS/exchange/contrib/schowengerdt/cm5backprop.html
218 http://www.aic.nrl.navy.mil/~kamgar/frame3.html

APPENDIX 2

Stop Words

A
ABOUT
ABOVE
ACROSS
AFTER
AFTERWARDS
AGAIN
AGAINST
ALL
ALMOST
ALONE
ALONG
ALREADY
ALSO
ALTHOUGH
ALWAYS
AMONG
AMONGST
AN
AND
ANOTHER
ANY
ANYHOW
ANYONE
ANYTHING
ANYWHERE
ARE
AROUND
AS
AT
BE
BECAME
BECAUSE
BECOME
BECOMES
BECOMING
BEEN
BEFORE
BEFOREHAND
BEHIND
BEING
BELOW
BESIDE
BESIDES
BETWEEN
BEYOND
BOTH
BUT
BY
CAN
CANNOT
CO
COULD
DONW
DURING
EACH
EG
EITHER
ELSE
ELSEWHERE
ENOUGH
ETC
EVEN
EVER
EVERY
EVERYONE
EVERYTHING
EVERYWHERE
EXCEPT
FEW
FIRST
FOR
FORMER
FORMERLY
FROM
FURTHER
HAD
HAS
HAVE
HE
HENCE
HER
HERE
HEREAFTER
HEREBY
HEREIN
HEREUPON
HERS
HERSELF
HIM
HIMSELF
HIS
HOW
HOWEVER
I
IE
IF
IN
INC
INDEED
INTO
IS
IT
ITS
ITSELF
LAST
LATTER
LATTERLY
LEAST
LESS
LTD
MANY
MAY
ME
MEANWHILE
MIGHT
MORE
MOREOVER
MOST
MOSTLY
MUCH
MUST
MY
MYSELF
NAMELY
NEITHER
NEVER
NEVERTHELESS
NEXT
NO
NOBODY
NONE
NOONE
NOR
NOT
NOTHING
NOW
NOWHERE
OF
OFF
OFTEN
ON
ONCE
ONE
ONLY
ONTO
OR
OTHER
OHTERS
OTHERWISE
OUR
OURS
OURSELVES
OUT
OVER
OWN
PER
PERHAPS
RATHER
SAME
SEEM
SEEMED
SEEMING
SEEMS
SEVERAL
SHE
SHOULD
SINCE
SO
SOME
SOMEHOW
SOMEONE
SOMETHING
SOMETIME
SOMETIMES
SOMEWHERE
STILL
SUCH
THAN
THAT
THE
THEIR
THEM
THEMSELVS
THEN
THENCE
THERE
THEREAFTER
TEREBY
THEREFORE
THEREIN
THEREUPON
THESE
THEY
THIS
THOSE
THOUGH
THROUGH
THROUGHOUT
THRU
THUS
TO
TOGETHER
TOO
TOWARD
TOWARDS
UNDER
UNTIL
UP
UPON
US
VERY
VIA
WAS
WE
WELL
WERE
WHAT
WHATEVER
WHEN
WHENCE
WHENEVER
WHERE
WHEREAFTER
WHEREAS
WHEREBY
WHEREIN
WHEREUPON
WHEREVER
WHETHER
WHITHER
WHICH
WHILE
WHO
WHOEVER
WHOLE
WHOM
WHOSE
WHY
WILL
WITH
WITHIN
WITHOUT
WOULD
YET
YOU
YOUR
YOURS
YOUSELF
YOUSELVES

APPENDIX 3

The Porter Algorithm for Affix Removal

Affix removal algorithms remove suffixes and/or prefixes from terms leaving a *stem*. These algorithms sometimes also transform the resultant stem.

The Porter algorithm for affix removal consists of a set of condition/action rules. The conditions fall into three classes: conditions on the stem, conditions on the suffix, and conditions on the rules.

There are several types of stem conditions.

1. The *measure*, denoted m, of a stem is based on its alternate vowel-consonant sequences. Vowels are a, e, i, o, u, and y if preceded by a consonant. Consonants are all letters that are not vowels. Let C stand for a sequence of consonants, and V for a sequence of vowels. The measure m, then, is defined as

$$[C](VC)^{m}[V]$$

The superscript m in the equation, which is the measure, indicates the number of VC sequences. Square brackets indicate an optional occurrence. Some examples of measures for terms follow.

Measure	Examples
m=0	TR, EE, TREE, Y, BY
m=1	TROUBLE, OATS, TREE, IVY
m=2	TROUBLES, PRIVATE, OATEN

2. *<X> - the stem ends with a given letter X
3. *v* - the stem contains a vowel
4. *d - the stem ends in a double consonant
5. *o - the stem ends with a consonant-vowel-consonant, sequence, where the final consonant is not w, x, or y.

The rules are divided into steps. The rule in a step are examined in sequence, and only one rule from a step can apply. The longest possible suffix is always removed because of the ordering of the rules within a step.

The algorithm is as follows.

```
{       step1a (word);
        step1b (stem);
        if (second or third rule of step 1b was used)
                step1b1 (stem);
        step1c (stem);
        step2 (stem);
        step3 (stem);
        step4 (stem);
        step5a (stem);
        step5b (stem);        }
```

The rules for the steps of the stemmer are as follows.

Step 1a Rules

Conditions	Suffix	Replacement	Examples
NULL	sses	ss	caresses → caress
NULL	ies	i	ponies → poni ties → tie
NULL	ss	ss	carress → carress
NULL	s	NULL	cats → cat

Step 1b Rules

Conditions	Suffix	Replacement	Examples
(m>0)	eed	ee	feed → feed agreed → agree
(*v*)	ed	NULL	plastered → plaster bled → bled
(*v*)	ing	NULL	sing → sing

Step 1b1 Rules

Conditions	Suffix	Replacement	Examples
NULL	at	ate	conflate(ed) → conflate
NULL	bl	ble	troubl(ing) → trouble
NULL	iz	ize	siz(ed) → size
(*d and not (*<L>or*<S>or*<Z>))	NULL	single letter	hopp(ing) → hop tann(ed) → tan fall(ing) → fall hiss(ing) → hiss fizz(ed) → fizz
(m=1 and *o)	NULL	e	fail(ing) → fail fil(ing) → file

Step 1c Rules

Conditions	Suffix	Replacement	Examples
(*v*)	y	i	happy → happi
			sky → sky

Step 2 Rules

Conditions	Suffix	Replacement	Examples
(m>0)	ational	ate	relational → relate
(m>0)	tional	tion	conditional → condition
			rational → rational
(m>0)	enci	ence	valenci → valence
(m>0)	anci	ance	hesitanci → hesitance
(m>0)	izer	ize	digitizer → digitize
(m>0)	abli	able	conformabli → conformable
(m>0)	alli	al	radicalli → radical
(m>0)	entli	ent	differentli → different
(m>0)	eli	e	vileli → vile
(m>0)	ousli	ous	analogousli → analogous
(m>0)	ization	ize	vietnamization → vietnamize
(m>0)	ation	ate	predication → predicate
(m>0)	ator	ate	operator → operate
(m>0)	alism	al	feudalism → feudal
(m>0)	iveness	ive	decisiveness → decisive
(m>0)	fulness	ful	hopefulness → hopeful
(m>0)	ousness	ous	callousness → callous
(m>0)	aliti	al	formaliti → formal
(m>0)	iviti	ive	sensitiviti → sensitive
(m>0)	biliti	ble	sensibiliti→ sensible

Step 3 Rules

Conditions	Suffix	Replacement	Examples
(m>0)	icate	ic	triplicate → triplic
(m>0)	ative	NULL	formative → form
(m>0)	alize	al	formalize → formal
(m>0)	iciti	ic	electriciti → electric
(m>0)	ical	ic	electrical → electric
(m>0)	ful	NULL	hopeful → hope
(m>0)	ness	NULL	goodness → good

Step 4 Rules

Conditions	Suffix	Replacement	Examples
(m>1)	al	NULL	revival → reviv
(m>1)	ance	NULL	allowance → allow
(m>1)	ence	NULL	inference → infer
(m>1)	er	NULL	airliner → airlin
(m>1)	ic	NULL	gyroscopic → gyroscop
(m>1)	able	NULL	adjustable → adjust
(m>1)	ible	NULL	defensible → defens
(m>1)	ant	NULL	irritant → irrit
(m>1)	ement	NULL	replacement → replac
(m>1)	ment	NULL	adjustment → adjust
(m>1)	ent	NULL	dependent → depend
(m>1) and (*<S>or*<T>)	ion	NULL	adoption → adopt
(m>1)	ou	NULL	homologou → homolog
(m>1)	ism	NULL	communism → commun
(m>1)	ate	NULL	activate → activ
(m>1)	iti	NULL	angulariti → angular
(m>1)	ous	NULL	homologous → homolog
(m>1)	ive	NULL	effective → effect
(m>1)	ize	NULL	bowdlerize → bowdler

Step 5a Rules

Conditions	Suffix	Replacement	Examples
(m>1)	e	NULL	probate → probat rate → rate
(m=1 and not *o)	e	NULL	cease → ceas

Step 5b Rules

Conditions	Suffix	Replacement	Examples
(m>1 and *d and *<L>)	NULL	single letter	controll → control roll → roll

APPENDIX 4

192-dimensional Word-Vector Space

welcom
group
team
research
scienc
lab
laboratori
center
staf
peopl
member
organ
societi
council
committe
institut
univers
school
edu
depart
student
professor
confer
workshop
event
registr
speaker
schedul
talk
speech
implement
model
prototyp
structur
control
signal
digit
analog
robot
architectur

sensor
machin
devic
mechan
filter
technic
manufactur
industri
technologi
product
project
design
develop
servic
fuzzi
reinforc
competit
hebbian
heb
radial
rbf
self-organ
kohonen
som
backpropag
genet
markov
princip
pca
nlpca
gaussian
bayesian
hopfield
supervis
unsupervis
theoret
rul
algorithm
method
techniqu

approach
function
compon
basi
connect
analysi
pattern
cluster
equat
recognit
vector
mathemat
volum
vol
journal
issu
proceed
report
publish
press
author
book
handbook
chapter
paper
articl
previou
index
list
content
abstract
introduct
document
overview
librari
archiv
lectur
tutori
not
cours

cellular
cell
biologi
biolog
brain
neurobiologi
molecular
dna
synaptic
synaps
axon
dendrit
neuron
layer
weight
input
output
hidden
nod
multilay
circuit
chip
hardwar
electron
electr
board
processor
java
applet
simul
demonstr
demo
softwar
program
run
cod
window
unix
postscript
fil

onlin
on-lin
newslett
e-mail
email
newsgroup
imag
figur
click
button
contact
messag
download
packag
memori
user
tool
profession
manag
market
financi
stock
financ
compani
corpor
com
trad
price
busi
object
engin
vlsi

APPENDIX 5

125-dimensional Word-Vector Space

welcom
group
team, staf, peopl
research
scienc
lab, laboratori, institut
center
member
organ
societi, council, committe
univers, school, edu, depart
student, professor
confer, workshop
event
registr
speaker, talk
schedul
speech
implement
model, prototyp
structur
control
signal
digit
analog
robot
architectur
sensor
machin, mechan
devic
filter
technic
manufactur, industri
technologi
product
project, design
develop
servic
fuzzi
reinforc
competit
hebbian, heb
radial, rbf
self-organ, kohonen, som
backpropag

genet
markov
princip, pca, nlpca
gaussian
bayesian
hopfield
supervis
unsupervis
theoret
rul
algorithm, method, techniqu
approach
function
compon
basi
connect
analysi
pattern
cluster
equat
recognit
vector
mathemat
journal, issu, volum, vol
proceed
report
publish, press
author
book, handbook, chapter
paper, articl, document
previou
index, list
abstract, content
introduct
overview
librari, archiv
cours, lectur, tutori
not
cell, cellular
biologi, biolog, neurobiologi
brain
molecular
synaps, synaptic, dna, axon, dendrit
neuron
layer, multilay

weight
input, output
hidden
nod
circuit
chip, vlsi
hardwar, electron, electr
board
processor
java, applet
demo, demonstr, simul
softwar
program, cod
run
unix, window
postscript, fil
onlin, on-lin
newslett, newsgroup
e-mail, email
imag, figur
click, button
contact, messag
download, packag
memori
user
tool
profession
manag
market, trad
financi, financ
stock, price
compani, corpor, com
busi
object
engin

APPENDIX 6

111-dimensional Word-Vector Space

welcom
group
team, staf, peopl
research, devic
lab, laboratori, institut
center
member
organ
societi, council, committe
univers, school, edu, depart, scienc, schedul
student, professor
confer, workshop, registr, proceed, schedul
event
speaker, talk
speech
implement, analog, devic, circuit, processor
model, prototyp, processor
structur
control, devic, processor
signal
digit
robot
architectur
sensor
machin, mechan
filter
technic
technologi, manufactur, industri, busi
product
project, design, devic, circuit
develop, manufactur, industri, circuit, processor
fuzzi
reinforc
competit
hebbian, heb
radial, rbf
self-organ, kohonen, som
backpropag
genet
markov
princip, pca, nlpca
gaussian
bayesian
hopfield
supervis
unsupervis
theoret
rul
algorithm, method, techniqu
approach
function
compon
basi
connect
analysi
pattern
cluster
equat
recognit
vector
mathemat
journal, issu, volum, vol
publish, press
author
book, handbook, chapter
paper, articl, document, proceed, report
previou
index, list
abstract, content
introduct
overview
librari, archiv
cours, lectur, tutori
not
cell, cellular
biologi, biolog, neurobiologi
brain
molecular
synaps, synaptic, dna, axon, dendrit
neuron
layer, multilay
weight
input, output, analog
hidden
nod
chip, vlsi, analog, circuit
hardwar, electron, electr, circuit
board
java, applet
demo, demonstr, simul
softwar
program, cod, processor, run
unix, window
postscript, fil

onlin, on-lin
newslett, newsgroup
e-mail, email
imag, figur
click, button
contact, messag
download, packag
memori
user
tool
profession
manag
market, trad, stock, price
financi, financ
compani, corpor, com, manufactur, industri, servic, busi, stock, price
object
engin

APPENDIX 7

Clusters formed in 192-dimensional Word-Vector Space

PRODUCT:
http://www.nd.com/new/index.htm
http://www.ann-pro.com/
http://www.neuralware.com/_private/company.htm
http://www.wardsystems.com/consum.htm
http://www.inc.com/incmagazine/archives/18961091.html
http://www.shef.ac.uk/~signalbox/page3.htm
http://www.biocompsystems.com/
http://www.ncs.co.uk/
http://www.globalweb.co.uk/gwl/
http://www.irnet.com/pages/press/050597merger.htm
http://www.biocompsystems.com/pages/financia.htm
http://www.pfr.com/index.html
http://ProfitTaker.com/technical_mendelsohn_small.htm
http://carlassoc.com/
http://www.progno.com/neural.htm
http://www.calsci.com/whatare.htm
http://www.neural.com/pressreleases/1997/April/Microsoft.html
http://www.fonix.com/corporate.html
http://www.neural.com/NetProphet/NetProphet.html
http://www.neural.com/pressreleases/1996/Apr/NetProphet.html
http://promland.com/
http://www.alliedtech.com/sbir/ss/88text.html
http://www.cam.org/~fsalvat/
http://ourworld.compuserve.com/homepages/FTPub/jcif.htm
http://www.bioele.nuee.nagoya-u.ac.jp/wsc1/papers/p031.html
http://www.bgif.no/neureka/about.html
http://www.neuralfusion.com/order.html
http://www.flextool.com/

SIMULATION:
http://www.zsolutions.com/backpack.htm
http://www.informatik.uni-stuttgart.de/ipvr/bv/pns/
http://cns-web.bu.edu/pub/laliden/WWW/nnet.frame.html
http://www.informatik.uni-stuttgart.de/ipvr/bv/projekte/snns/announce.html
http://www.industry.net/c/mn/_swnet/
http://www.ai.univie.ac.at/oefai/nn/tool.html
http://www.nada.kth.se/nada/sans/compneuro.html
http://www.nada.kth.se/nada/sans/walking.html
http://tag-www.larc.nasa.gov/tops/tops95/exhibits/emt/emt-174-95/emt17495.html
http://ic-www.arc.nasa.gov/ic/projects/neuro/sc_nav/sc_nav.html
http://accurate-automation.com/projects/ncrobot.htm#Robotics

DOCUMENT:
http://ourworld.compuserve.com/homepages/attg/attgprdf.html
http://websom.hut.fi/websom/

NEURON:
http://www.canneuro.stjohns.co.uk/frameadt.htm
http://www.neurotec.com/techno/nn/nn5.html
http://ourworld.compuserve.com/homepages/attg/attgnrlt.html
http://www.rpl.com/features.html
http://www.neuralt.com/Whitepaper/NTL_theory.htm
http://rfhs8012.fh-regensburg.de/~sauer/jfroehl/diplom/e-g.html#FeedforwardType
http://rfhs8012.fh-regensburg.de/~sauer/jfroehl/diplom/e-13-text.html
http://www-dse.doc.ic.ac.uk/~nd/surprise_96/journal/vol2/cs11/article2.html
http://rfhs8012.fh-regensburg.de/~sauer/jfroehl/diplom/e-11-text.html
http://neuron.duke.edu/
http://www.voicenet.com/~rybak/resp.html
http://www.uivt.cas.cz/ics/nn-biol.html
http://www.eeb.ele.tue.nl/neural/neuron.html
http://www.nnc.yale.edu/papers/NIPS94/nipsfin.html
http://www.nnc.yale.edu/papers/ebench/ebench.html
http://suhep.phy.syr.edu/MM/Biology/bio_org.html#work
http://www.cs.utexas.edu/users/nn/web-pubs/htmlbook96/wiskott/node3.html#figrunBlobsarch
http://www.cs.utexas.edu/users/nn/web-pubs/htmlbook96/dong/node3.html#SECTION00021000000000000000
http://gepasi.dbs.aber.ac.uk/roy/koho/kohonen.htm

FUZZY:
http://www.ppgsoft.com/fuzneu.html
http://cs.uregina.ca/~narate/Links/nn+fuzzy.html
http://peacock.pse.che.tohoku.ac.jp/~yyama/work/ann/section3.2.html
http://www.tech.plym.ac.uk/soc/sameer/abstract/msc.htm
http://www.forwiss.uni-erlangen.de/aknn/index.eng.html
http://www.dreamlabs.com/~webland/c_p/icnn95/nn95s608.htm
http://pcim.com/arc/conf/mexpo/11.html
http://i80s25.ira.uka.de/neurofuzzy/homepage.html

GROUP:
http://www.msci.memphis.edu/~garzonm/nnets/nnpubs.html
http://www.ewh.ieee.org/tc/nnc/
http://www.arc.unm.edu/wcci-98/ijcnn.html
http://www.cs.cmu.edu/afs/cs/project/cnbc/nips/NIPS.html
http://shay.ecn.purdue.edu/~csglee/icra-98/
http://diwww.epfl.ch/w3mantra/mn96.html
http://www.abo.fi/~abulsari/EANN96.html
http://www.cs.caltech.edu/~learn/nncm.html
http://robotics.eecs.berkeley.edu/MURI/muriproject7.html
http://www.ai.univie.ac.at/oefai/nn/ctg-cu96.html
http://www.cs.cmu.edu/afs/cs.cmu.edu/project/alv/member/www/projects/ALVINN.html
http://www.ai.univie.ac.at/oefai/nn/neufodi.html
http://diwww.epfl.ch/lami/team/cornu/ametis2/ametis2.html
http://www.iic.umanitoba.ca/projects-hc.htm
http://www.emsl.pnl.gov:2080/docs/cie/neural/newsgroups.html
http://www.loria.fr/exterieur/equipe/rfia/cortex/servers.html
http://http.hq.eso.org/~fmurtagh/clustering.html
http://cns-web.bu.edu/inns/nn/N103.html
http://www.fit.qut.edu.au/~robert/rulex.html
http://polimage.polito.it/groups/nngroup.html
http://dibe.unige.it/department/ncas/group.html
http://www.cs.rulimburg.nl/~antal/nn.html
http://dsi.ing.unifi.it/neural/
http://neural-server.aston.ac.uk/postdocs/on_line_learning.html
http://dibe.unige.it/neuronet/DIBE/SPUG.html
http://www.emsl.pnl.gov:2080/docs/cie/neural/

http://web.egr.msu.edu/CSANNLAB/ann.html
http://jnns-www.okabe.rcast.u-tokyo.ac.jp/jnns/home.html
http://cns-web.bu.edu/inns/
http://www.neuronet.ph.kcl.ac.uk/neuronet/organisations/enns.html
http://www.wins.uva.nl/research/neuro/ias-ras/ias.html
http://cscsi.sfu.ca/
http://www-dsi.ing.unifi.it/neural/siren/sirenEN.html

SENSOR:
http://www.dacs.dtic.mil/techs/neural/neural10.html
http://www.emsl.pnl.gov:2080/proj/neuron/papers/keller.aist94.html
http://www.emsl.pnl.gov:2080/proj/neuron/papers/keller.wcnn95.html
http://www.emsl.pnl.gov:2080/proj/neuron/briefs/cardio.html

LECTURE:
http://www.cs.iastate.edu/~cs474/spring97/lecturenotes.html
http://www.cs.utk.edu/~mclennan/Classes/594-MNN/

BOOK:
http://www.shef.ac.uk/psychology/gurney/notes/index.html
http://www.amazon.com/exec/obidos/ISBN=0201539217/5019-1394227-027506
http://www.amazon.com/exec/obidos/ISBN%3D0879422890/5019-1394227-027506

COURSE:
http://www.eng.fsu.edu/feeds/foo.html
http://adam.fiz.huji.ac.il/orens/NNB.html
http://www.abo.fi/~abulsari/courses.html

COMPETITIVE:
http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/DemoGNG/GNG.html
http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/JavaPaper/
http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/JavaPaper/node15.html#SECTION00600000000000000000
http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/JavaPaper/node9.html#SECTION00500000000000000000
http://www.nd.com/learning/compete.htm
http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/JavaPaper/node17.html
http://Neuron.derby.ac.uk/neuron/resneuralnet.html
http://iserv.iki.kfki.hu/pub/fomin.robhah.abs.html
http://www.nd.com/learning/hebbian.htm
http://www.santafe.edu/~jmerelo/neurolib/node5.html

MACHINE:
http://psy.uq.edu.au/~mav/java/Necker.html
http://diwww.epfl.ch/w3mantra/mantra_machine.html

BAYESIAN:
http://www.aist.go.jp/ETL/etl/suri/motomura/BN/bn-java.html
http://set.gmd.de/AS/nn/publi/nips-94.html
http://www.cs.utoronto.ca/~radford/thesis.abstract.html
http://www.faqs.org/faqs/ai-faq/neural-nets/part3/section-7.html
http://www.aist.go.jp/NIBH/~b0616/Lab/BSOM1/index.html

HOPFIELD:
http://suhep.phy.syr.edu/courses/modules/MM/SIM/Hopfield/
http://www.aic.nrl.navy.mil/~kamgar/frame3.html

JAVA:
http://home.cc.umanitoba.ca/~umcorbe9/anns.html
http://rfhs8012.fh-regensburg.de/~saj39122/jfroehl/diplom/e-index.html
http://www.ifsys.co.uk/Showcase1/index.html
http://www.cl.ais.net/uthed/neural.htm
http://www.aist.go.jp/ETL/~akaho/MixtureEM.html

DEMONSTRATION:
http://www.netkonect.net/n/neural/nd_contents.html

RADIAL:
http://www.mathworks.com/products/neuralnet/
http://www.cns.ed.ac.uk/people/mark/intro/node3.html#SECTION00030000000000000000
http://www.cns.ed.ac.uk/people/mark/intro/node8.html
http://www.cns.ed.ac.uk/people/mark/intro/intro.html
http://hebb.cis.uoguelph.ca/~deb/27642/LectureNotes/rbf/node2.html#SECTION00020000000000000000
http://www.nd.com/models/rbf.htm
http://i3a.dtic.ua.es:8080/local/neural-l96/0335.html
http://www.oup-usa.org:9200/9205/abs/ncg2_6.htm

HARDWARE:
http://www.mcs.com/~drt/svbp.html#nearest
http://www.webcom.com/~ictech/nram.html
http://www.ece.uc.edu/~ansl/hardware.html
http://msia02.msi.se/~lindsey/elba2html/section3_5.htm
http://www1.physik.unibas.ch/~leimgrub/nntrack/node5.html#SECTION00050000000000000000
http://polimage.polito.it/groups/daniela/daniela.html
http://www.eln.utovrm.it/~cirlab/cnn_chip.html#6x6DPCNN
http://cns-web.bu.edu/muri/year1-report/4f.html
http://petrus.upc.es/~microele/neuronal/US/projectes.html
http://www.dhp.nl/~heiniw/thesis/
http://msia02.msi.se/~lindsey/elba2html/subsubsection3_3_1_4.html#SECTION00031400000000000000
http://www.cbu.edu/~pong/624lsb2.htm

CORPORATION:
http://www.ibm.fr/france/cdlab/zicope.htm

VOLUME:
http://www.brunel.ac.uk/~hssrjis/issue/j714.html
http://www.brunel.ac.uk/~hssrjis/issue/index.html
http://cns-web.bu.edu/inns/nn.html
http://www.eeb.ele.tue.nl/neural/contents/neural_computation.html
http://neural-server.aston.ac.uk/NCAF/journal/index.html

SIGNAL:
http://web.egr.msu.edu/CSANNLAB/bss.html

SPEECH:
http://www.sensoryinc.com/neural.shtml
http://ciips.ee.uwa.edu.au/Papers/Conference_Papers/title.html
http://www-speech.sri.com/projects/hybrid.html
http://www-laforia.ibp.fr/CONNEX/Articles/artieres-eurospeech95.abs.txt
http://www-karc.crl.go.jp/avis/zhang/study.html

ARCHIVE:
http://www.lehigh.edu/~ay00/nn.html

LIBRARY:
http://www.ai.univie.ac.at/oefai/nn/conn_biblio.html
http://www.clients.globalweb.co.uk/nctt/newsletter/01/
http://ciips.ee.uwa.edu.au/Papers/Conference_Papers//1992/02//Index.html

ISSUE:
http://www.msci.memphis.edu:80/~jagota/JANN/index.html

WORKSHOP:
http://www.cnel.ufl.edu:80/nnsp97/
http://www.emsl.pnl.gov:2080/proj/neuron/workshops/WEEANN95/
http://www.lpac.ac.uk/SEL-HPC/Events/NeuralWorkshop/index.html

JOURNAL:
http://www.csu.edu.au/ci/aboutci.html
http://www-mitpress.mit.edu/journal-home.tcl?issn=08997667

HANDBOOK:
http://www.oup-usa.org/acadref/honc.html
http://www.oup-usa.org/acadref/nc_toc.html

LAB:
http://www.ececs.uc.edu/~ansl/
http://www.iscs.nus.edu.sg/~leowwk/isl/nn.html

MOLECULAR:
http://www.msci.memphis.edu/~garzonm/#pubs
http://www.ice.ge.cnr.it/Biology.html

NEUROBIOLOGY:
http://www.cnl.salk.edu/CNL/
http://bbf-www.uia.ac.be/

BRAIN:
http://www.vxm.com/21R.7.html

SUPERVISED:
http://www.wspc.com.sg/journals/ijns/82/zhou.html
http://www.wspc.com.sg/journals/ijns/82/hung.html

SELF-ORGANIZED:
http://www.cs.utexas.edu/users/nn/web-pubs/htmlbook96/sirosh/
http://www.nd.com/models/sofm.htm
http://www-karc.crl.go.jp/avis/maekawa/study.html
http://www.neuroinformatik.ruhr-uni-bochum.de/ini/PEOPLE/fritzke/research/new.html
http://iserv.iki.kfki.hu/pub/icnn94/rozgonyi.scaling.abs.html

PRINCIPAL:
http://web.egr.msu.edu/CSANNLAB/pca.html
http://www.nd.com/models/pca.htm
http://ftpserver.nus.sg/docs/csbiblio/Neural/pca.html
http://www.neuroinformatik.ruhr-uni-bochum.de/icann96/IC96/PROGRAM/aapo@nucleus.html
http://i3a.dtic.ua.es:8080/local/neural-l96/0320.html

REINFORCEMENT:
http://www.cs.colostate.edu/~anderson/matlab-papers/rl/rl.html
http://iserv.iki.kfki.hu/pub/icml96/szepes.genreinf.abs.html
http://www.cs.cmu.edu/~reinf/web/cmu-rl-papers.html

http://www.cs.cmu.edu/Groups/reinforcement/web/homepage.html
http://www.oup-usa.org:9200/9205/abs/ncc3.htm

GENETIC:
http://www.neuralware.com/products/proplus/proii5.html
http://www.isd.uni-stuttgart.de/~rudolph/geneticalg/genetic_abstract_01.html
http://arti.vub.ac.be/www/brochure/section1.3.0.3.html
http://www.ntu.edu/1/atmp/1997Courses/mc97043001.htm
http://www.epcc.ed.ac.uk/tracs/tracs.seminars/abstracts/giulio.html

HEBB:
http://www2.shef.ac.uk/psychology/gurney/notes/l6/subsection3_1_2.html#SECTION0001200000000000000

GAUSSIAN:
http://www.cs.utoronto.ca/~zoubin/zoubin/linear-gaussian.abstract.html
http://www.wspc.com.sg/journals/ijns/81/firmin.html

BACKPROPAGATION:
http://www.nd.com/learning/backprop.htm
http://herens.idiap.ch/~perry/moerland-94.1.bib.abs.html
http://outside.gsfc.nasa.gov/ESS/exchange/contrib/schowengerdt/cm5backprop.html
http://sunflower.singnet.com.sg/~wspclib/Books/compsci/3094.html

APPENDIX 8

Clusters formed in 111-dimensional Word-Vector Space

DOWNLOAD:
http://www.nd.com/new/index.htm
http://www.neuralfusion.com/order.html
http://www.mcs.com/~drt/svbp.html#nearest

COMPANY:
http://www.neural.com/pressreleases/1997/April/Microsoft.html
http://www.imet.com/pages/press/050597merger.htm
http://www.neural.com/NetProphet/NetProphet.html
http://www.neural.com/pressreleases/1996/Apr/NetProphet.html
http://www.inc.com/incmagazine/archives/18961091.html
http://www.fonix.com/corporate.html
http://www.wardsystems.com/consum.htm
http://www.neuralware.com/_private/company.htm
http://www.biocompsystems.com/
http://www.globalweb.co.uk/gwl/
http://www.shef.ac.uk/~signalbox/page3.htm
http://ProfitTaker.com/technical_mendelsohn_small.htm
http://www.biocompsystems.com/pages/financia.htm
http://carlassoc.com/
http://www.pfr.com/index.html
http://www.progno.com/neural.htm
http://promland.com/
http://www.ncs.co.uk/
http://www.ann-pro.com/
http://www.alliedtech.com/sbir/ss/88text.html
http://www.camneuro.stjohns.co.uk/frameadt.htm
http://www.cam.org/~fsalvat/
http://ourworld.compuserve.com/homepages/FTPub/jcif.htm
http://www.bioele.nuee.nagoya-u.ac.jp/wsc1/papers/p031.html

DEMO:
http://www.zsolutions.com/backpack.htm
http://www.ai.univie.ac.at/oefai/nn/tool.html
http://www.informatik.uni-stuttgart.de/ipvr/bv/pns/
http://www.netkonect.net/n/neural/nd_contents.html
http://cns-web.bu.edu/pub/laliden/WWW/nnet.frame.html
http://www.industry.net/c/mn/_swnet/
http://www.informatik.uni-stuttgart.de/ipvr/bv/projekte/snns/announce.html
http://www.rpl.com/features.html
http://www.nada.kth.se/nada/sans/compneuro.html
http://www.mathworks.com/products/neuralnet/

PAPER:
http://websom.hut.fi/websom/
http://ourworld.compuserve.com/homepages/attg/attgprdf.html

TEAM:
http://www.bgif.no/neureka/about.html

FUZZY:
http://www.ppgsoft.com/fuzneu.html
http://cs.uregina.ca/~narate/Links/nn+fuzzy.html
http://peacock.pse.che.tohoku.ac.jp/~yyama/work/ann/section3.2.html
http://www.tech.plym.ac.uk/soc/sameer/abstract/msc.htm
http://www.dreamlabs.com/~webland/c_p/icnn95/nn95s608.htm
http://www.forwiss.uni-erlangen.de/aknn/index.eng.html
http://pcim.com/arc/conf/mexpo/11.html
http://i80s25.ira.uka.de/neurofuzzy/homepage.html
http://www.flextool.com/

NEURON:
http://www.neurotec.com/techno/nn/nn5.html
http://ourworld.compuserve.com/homepages/attg/attgnrlt.html
http://rfhs8012.fh-regensburg.de/~sauer/jfroehl/diplom/e-g.html#FeedforwardType
http://rfhs8012.fh-regensburg.de/~sauer/jfroehl/diplom/e-13-text.html
http://www-dse.doc.ic.ac.uk/~nd/surprise_96/journal/vol2/cs11/article2.html
http://rfhs8012.fh-regensburg.de/~sauer/jfroehl/diplom/e-11-text.html
http://www.neuralt.com/Whitepaper/NTL_theory.htm
http://www.cs.utexas.edu/users/nn/web-pubs/htmlbook96/wiskott/node3.html#figrunBlobsarch
http://neuron.duke.edu/
http://www.voicenet.com/~rybak/resp.html
http://www.uivt.cas.cz/ics/nn-biol.html
http://www.msci.memphis.edu/~garzonm/#pubs
http://www.eeb.ele.tue.nl/neural/neuron.html
http://www.nnc.yale.edu/papers/NIPS94/nipsfin.html
http://www.nnc.yale.edu/papers/ebench/ebench.html
http://suhep.phy.syr.edu/MM/Biology/bio_org.html#work
http://www.calsci.com/whatare.htm

JOURNAL:
http://www.msci.memphis.edu/~garzonm/nnets/nnpubs.html
http://www-mitpress.mit.edu/journal-home.tcl?issn=08997667
http://www.brunel.ac.uk/~hssrjis/issue/j714.html
http://www.brunel.ac.uk/~hssrjis/issue/index.html
http://neural-server.aston.ac.uk/NCAF/journal/index.html
http://www.eeb.ele.tue.nl/neural/contents/neural_computation.html
http://cns-web.bu.edu/inns/nn.html
http://www.csu.edu.au/ci/aboutci.html
http://www.clients.globalweb.co.uk/nctt/newsletter/01/
http://www.msci.memphis.edu:80/~jagota/JANN/index.html

EQUATION:
http://www.cs.utexas.edu/users/nn/web-pubs/htmlbook96/dong/node3.html#SECTION00021000000000000000

CHIP:
http://polimage.polito.it/groups/daniela/daniela.html
http://www.eln.utovrm.it/~cirlab/cnn_chip.html#6x6DPCNN
http://cns-web.bu.edu/muri/year1-report/4f.html
http://petrus.upc.es/~microele/neuronal/US/projectes.html
http://www.dhp.nl/~heiniw/thesis/
http://www.cbu.edu/~pong/624lsb2.htm
http://www.webcom.com/~ictech/nram.html
http://msia02.msi.se/~lindsey/elba2html/subsubsection3_3_1_4.html#SECTION00031400000000000000
http://diwww.epfl.ch/w3mantra/mantra_machine.html

http://www.ibm.fr/france/cdlab/zicope.htm
http://www.ece.uc.edu/~ansl/hardware.html
http://www.amazon.com/exec/obidos/ISBN%3D0879422890/5019-1394227-027506
http://msia02.msi.se/~lindsey/elba2html/section3_5.html
http://www1.physik.unibas.ch/~leimgrub/nntrack/node5.html#SECTION00050000000000000000
http://robotics.eecs.berkeley.edu/MURI/muriproject7.html
http://www.ai.univie.ac.at/oefai/nn/neufodi.html
http://www.ai.univie.ac.at/oefai/nn/ctg-cu96.html
http://www.cs.cmu.edu/afs/cs.cmu.edu/project/alv/member/www/projects/ALVINN.html
http://diwww.epfl.ch/lami/team/cornu/ametis2/ametis2.html

SENSOR:
http://www.dacs.dtic.mil/techs/neural/neural10.html
http://www.emsl.pnl.gov:2080/proj/neuron/papers/keller.aist94.html
http://www.emsl.pnl.gov:2080/proj/neuron/papers/keller.wcnn95.html
http://www.emsl.pnl.gov:2080/proj/neuron/briefs/cardio.html

COURSE:
http://www.cs.iastate.edu/~cs474/spring97/lecturenotes.html
http://adam.fiz.huji.ac.il/orens/NNB.html
http://www.abo.fi/~abulsari/courses.html
http://www.eng.fsu.edu/feeds/foo.html
http://www.iic.umanitoba.ca/projects-hc.htm

MATHEMATICS:
http://www.cs.utk.edu/~mclennan/Classes/594-MNN/

BOOK:
http://www.shef.ac.uk/psychology/gurney/notes/index.html
http://www.oup-usa.org/acadref/honc.html
http://www.amazon.com/exec/obidos/ISBN=0201539217/5019-1394227-027506
http://www.oup-usa.org/acadref/nc_toc.html

COMPETITIVE:
http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/DemoGNG/GNG.html
http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/JavaPaper/node17.html
http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/JavaPaper/
http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/JavaPaper/node15.html#SECTION00060000000000000000
http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/JavaPaper/node9.html#SECTION00050000000000000000
http://www.nd.com/learning/compete.htm

MACHINE:
http://psy.uq.edu.au/~mav/java/Necker.html

BAYESIAN:
http://www.aist.go.jp/ETL/etl/suri/motomura/BN/bn-java.html
http://set.gmd.de/AS/nn/publi/nips-94.html
http://www.cs.utoronto.ca/~radford/thesis.abstract.html
http://www.faqs.org/faqs/ai-faq/neural-nets/part3/section-7.html
http://www.aist.go.jp/NIBH/~b0616/Lab/BSOM1/index.html

HOPFIELD:
http://suhep.phy.syr.edu/courses/modules/MM/SIM/Hopfield/
http://www.aic.nrl.navy.mil/~kamgar/frame3.html

JAVA:

http://home.cc.umanitoba.ca/~umcorbe9/anns.html
http://rfhs8012.fh-regensburg.de/~saj39122/jfroehl/diplom/e-index.html
http://www.ifsys.co.uk/Showcase1/index.html
http://www.cl.ais.net/uthed/neural.htm
http://www.aist.go.jp/ETL/~akaho/MixtureEM.html

CONTROL:
http://www.nada.kth.se/nada/sans/walking.html
http://tag-www.larc.nasa.gov/tops/tops95/exhibits/emt/emt-174-95/emt17495.html
http://ic-www.arc.nasa.gov/ic/projects/neuro/sc_nav/sc_nav.html
http://accurate-automation.com/projects/ncrobot.htm#Robotics

SIGNAL:
http://web.egr.msu.edu/CSANNLAB/bss.html

SPEECH:
http://www.sensoryInc.com/neural.shtml
http://ciips.ee.uwa.edu.au/Papers/Conference_Papers/title.html
http://www-speech.sri.com/projects/hybrid.html
http://www-laforia.ibp.fr/CONNEX/Articles/artieres-eurospeech95.abs.txt

INDEX:
http://www.emsl.pnl.gov:2080/docs/cie/neural/newsgroups.html

LIBRARY:
http://www.lehigh.edu/~ay00/nn.html
http://www.ai.univie.ac.at/oefai/nn/conn_biblio.html

CONFERENCE:
http://www.loria.fr/exterieur/equipe/rfia/cortex/servers.html
http://http.hq.eso.org/~fmurtagh/clustering.html
http://cns-web.bu.edu/inns/nn/N103.html
http://www.cs.caltech.edu/~learn/nncm.html
http://www.fit.qut.edu.au/~robert/rulex.html
http://www.cs.cmu.edu/afs/cs/project/cnbc/nips/NIPS.html
http://www.emsl.pnl.gov:2080/proj/neuron/workshops/WEEANN95/
http://shay.ecn.purdue.edu/~csglee/icra-98/
http://www.abo.fi/~abulsari/EANN96.html
http://diwww.epfl.ch/w3mantra/mn96.html
http://www.cnel.ufl.edu:80/nnsp97/
http://www.lpac.ac.uk/SEL-HPC/Events/NeuralWorkshop/index.html

SOCIETY:
http://jnns-www.okabe.rcast.u-tokyo.ac.jp/jnns/home.html
http://cns-web.bu.edu/inns/
http://www.neuronet.ph.kcl.ac.uk/neuronet/organisations/enns.html
http://www.wins.uva.nl/research/neuro/ias-ras/ias.html
http://cscsi.sfu.ca/
http://www.ewh.ieee.org/tc/nnc/
http://www.arc.unm.edu/wcci-98/ijcnn.html
http://www-dsi.ing.unifi.it/neural/siren/sirenEN.html

LAB:
http://www.ececs.uc.edu/~ansl/
http://www.emsl.pnl.gov:2080/docs/cie/neural/
http://www.iscs.nus.edu.sg/~leowwk/isl/nn.html
http://www.cnl.salk.edu/CNL/
http://web.egr.msu.edu/CSANNLAB/ann.html

GROUP:
http://polimage.polito.it/groups/nngroup.html
http://dibe.unige.it/department/ncas/group.html
http://www.cs.rulimburg.nl/~antal/nn.html
http://dsi.ing.unifi.it/neural/
http://neural-server.aston.ac.uk/postdocs/on_line_learning.html
http://dibe.unige.it/neuronet/DIBE/SPUG.html

BIOLOGY:
http://bbf-www.uia.ac.be/
http://www.ice.ge.cnr.it/Biology.html

BRAIN:
http://www.vxm.com/21R.7.html

RADIAL:
http://www.cns.ed.ac.uk/people/mark/intro/node3.html#SECTION00030000000000000000
http://hebb.cis.uoguelph.ca/~deb/27642/LectureNotes/rbf/node2.html#SECTION00020000000000000000
http://www.cns.ed.ac.uk/people/mark/intro/node8.html
http://www.cns.ed.ac.uk/people/mark/intro/intro.html
http://www.nd.com/models/rbf.htm
http://www.oup-usa.org:9200/9205/abs/ncg2_6.htm
http://i3a.dtic.ua.es:8080/local/neural-l96/0335.html

SUPERVISED:
http://www.wspc.com.sg/journals/ijns/82/zhou.html
http://www.wspc.com.sg/journals/ijns/82/hung.html

SELF-ORGANIZED:
http://www.cs.utexas.edu/users/nn/web-pubs/htmlbook96/sirosh/
http://www.nd.com/models/sofm.htm
http://www-karc.crl.go.jp/avis/maekawa/study.html
http://www.neuroinformatik.ruhr-uni-bochum.de/ini/PEOPLE/fritzke/research/new.html
http://gepasi.dbs.aber.ac.uk/roy/koho/kohonen.htm
http://ciips.ee.uwa.edu.au/Papers/Conference_Papers//1992/02//Index.html
http://iserv.iki.kfki.hu/pub/icnn94/rozgonyi.scaling.abs.html

PRINCIPAL:
http://web.egr.msu.edu/CSANNLAB/pca.html
http://www.nd.com/models/pca.htm
http://ftpserver.nus.sg/docs/csbiblio/Neural/pca.html
http://i3a.dtic.ua.es:8080/local/neural-l96/0320.html
http://www.neuroinformatik.ruhr-uni-bochum.de/icann96/IC96/PROGRAM/aapo@nucleus.html

REINFORCEMENT:
http://www.cs.colostate.edu/~anderson/matlab-papers/rl/rl.html
http://iserv.iki.kfki.hu/pub/icml96/szepes.genreinf.abs.html
http://www.cs.cmu.edu/~reinf/web/cmu-rl-papers.html
http://www.cs.cmu.edu/Groups/reinforcement/web/homepage.html
http://www.oup-usa.org:9200/9205/abs/ncc3.htm

GENETIC:
http://www.neuralware.com/products/proplus/proii5.html
http://www.isd.uni-stuttgart.de/~rudolph/geneticalg/genetic_abstract_01.html
http://arti.vub.ac.be/www/brochure/section1.3.0.3.html
http://www.ntu.edu/1/atmp/1997Courses/mc97043001.htm
http://www.epcc.ed.ac.uk/tracs/tracs.seminars/abstracts/giulio.html

HEBBIAN:
http://Neuron.derby.ac.uk/neuron/resneuralnet.html
http://www2.shef.ac.uk/psychology/gurney/notes/l6/subsection3_1_2.html#SECTION00012000000000000000
http://www.nd.com/learning/hebbian.htm
http://iserv.iki.kfki.hu/pub/fomin.robhah.abs.html
http://www.santafe.edu/~jmerelo/neurolib/node5.html

MARKOV:
http://www-karc.crl.go.jp/avis/zhang/study.html

GAUSSIAN:
http://www.cs.utoronto.ca/~zoubin/zoubin/linear-gaussian.abstract.html
http://www.wspc.com.sg/journals/ijns/81/firmin.html

BACKPROPAGATION:
http://www.nd.com/learning/backprop.htm
http://herens.idiap.ch/~perry/moerland-94.1.bib.abs.html
http://outside.gsfc.nasa.gov/ESS/exchange/contrib/schowengerdt/cm5backprop.html
http://sunflower.singnet.com.sg/~wspclib/Books/compsci/3094.html

APPENDIX 9

URLs of Web Pages from Table 1.1

1 http://peacock.tnjc.edu.tw/ADD/sport/faq.html
2 http://www.cs.utoronto.ca/~cwong/tennis/tennis.html
3 http://www.tennismag.com.au/
4 http://www.tennisw.com/
5 http://tennis.org.uk/tennistoday/
6 http://alexia.lis.uiuc.edu/~kieraldo/tennis/rules.htm
7 http://www.volleyball.org/Index.html
8 http://www.volleyball.org/history.html
9 http://www.volleyball.org/whatis.html
10 http://www.volleyball.org/magazines.html
11 http://www.teleport.com/~lindac/accord.htm
12 http://www-th.phys.rug.nl/~nijhof/accordions.html
13 http://www.srv.net/~cathyw/accord.html
14 http://www.cs.cmu.edu/afs/cs/user/phoebe/accordion/web-pointers.html
15 http://www2.idsonline.com/jeff/jazz.html
16 http://www.acns.nwu.edu/jazz/education.html
17 http://www.jazzcentralstation.com/jcs/station/jazzdest/chicago/history.html
18 http://www.mc.maricopa.edu/its/lib/academic/Ibt/Ibt204/projects/jazz/jazz.html
19 http://www.yahoo.com/Entertainment/Music/Genres/Jazz/
20 http://www.bris.ac.uk/Depts/Philosophy/VL/
21 http://www.ed.ac.uk/~ejua35/phillink.htm
22 http://people.delphi.com/gkemerling/index.htm#lin
23 http://www.clients.globalweb.co.uk/nctt/newsletter/01/
24 http://cns-web.bu.edu/pub/snorrason/professional/neural.html
25 http://java.sun.com/docs/index.html

APPENDIX 10

Voronoi Region and Delaunay Triangulation

Voronoi Region

Given a set of vectors $w_1, \ldots, w_N$ in R^n, the Voronoi Region V_i of a particular vector w_i is defined as the set of all points in R^n for which w_i is the nearest vector.

$$V_i = \{ \xi \in R^n \mid i = \arg\min_{j \in \{1, \ldots, N\}} \|\xi - w_j\| \}$$

The partition of R^n formed by all Voronoi polygons is called Voronoi Tessellation or Dirichlet Tessellation. (Figure I and Figure II)

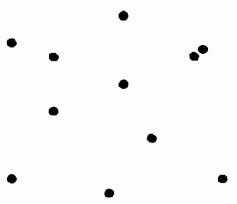


Figure I Set of reference vectors

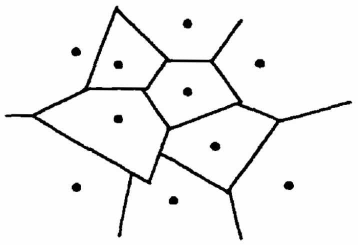


Figure II Corresponding Voronoi tessellation

Delaunay Triangulation

If one connects all pairs of points for which the respective Voronoi regions share an edge (an (n-1)-dimensional hypersurface for spaces of dimension n) one gets the Delaunay Triangulation. (Figure III)

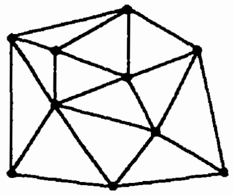


Figure III Corresponding Delaunay triangulation

APPENDIX 11

URLs of Web Pages from Table 4.2

1	http://www.lehigh.edu/~ay00/nn.html
2	http://www.loria.fr/exterieur/equipe/rfia/cortex/servers.html
3	http://www.cis.hut.fi/~jari/research.html
4	http://wwwsyseng.anu.edu.au/lsg/links.html
5	http://ai.iit.nrc.ca/subjects/Neural.html
6	http://www-dse.doc.ic.ac.uk/~nd/surprise_96/journal/vol2/cs11/article2.html
7	http://www.dacs.dtic.mil/techs/neural/neural10.html
8	http://www.cs.colostate.edu/~anderson/matlab-papers/rl/rl.html
9	http://www.ntu.edu/1/atmp/1997Courses/mc97043001.htm
10	http://www.springer-ny.com/nst/reviews/irwin_rev.html
11	http://www.nd.com/
12	http://www.ann-pro.com/
13	http://www.neuralfusion.com/
14	http://www.sensoryinc.com/
15	http://www.progno.com/default.htm
16	http://www.nd.com/demo/demo.htm
17	http://www.scriptsoftware.com/index.html
18	http://www.unica-usa.com/download.htm
19	http://www.aist.go.jp/ETL/~akaho/MixtureEM.html
20	http://home.cc.umanitoba.ca/~umcorbe9/mlp.html
21	http://polimage.polito.it/groups/nngroup.html
22	http://www.dc.fi.udc.es/Welcome.html
23	http://dibe.unige.it/neuronet/DIBE/
24	http://dibe.unige.it/department/ncas/group.html
25	http://www.brunel.ac.uk:8080/depts/ee/Research_Programme/NN/

APPENDIX 12

URLs of Web Pages from Table 4.3

1 http://www.tennis.org.uk/
2 http://www.tennismag.com.au/
3 http://www.tennisw.com/
4 http://tennis.org.uk/tennistoday/
5 http://alexia.lis.uiuc.edu/~kieraldo/tennis/rules.htm
6 http://www2.idsonline.com/jeff/jazz.html
7 http://www.acns.nwu.edu/jazz/education.html
8 http://www.jazzcentralstation.com/jcs/station/jazzdest/chicago/history.html
9 http://www.mc.maricopa.edu/its/lib/academic/lbt/lbt204/projects/jazz/jazz.html
10 http://www.yahoo.com/Entertainment/Music/Genres/Jazz/
11 http://www.lehigh.edu/~ay00/nn.html
12 http://www.loria.fr/exterieur/equipe/rfia/cortex/servers.html
13 http://www.cis.hut.fi/~jari/research.html
14 http://wwwsyseng.anu.edu.au/lsg/links.html
15 http://ai.iit.nrc.ca/subjects/Neural.html
16 http://www-dse.doc.ic.ac.uk/~nd/surprise_96/journal/vol2/cs11/article2.html
17 http://www.dacs.dtic.mil/techs/neural/neural10.html
18 http://www.teleweb.net/neuralnet/introtonn/index.htm
19 http://www.nd.com/welcome/whatisnn.htm
20 http://blizzard.gis.uiuc.edu/htmldocs/Neural/neural.html
21 http://www.nd.com/
22 http://www.ann-pro.com/
23 http://www.neuralfusion.com/
24 http://www.sensoryinc.com/
25 http://www.progno.com/default.htm

APPENDIX 13

URLs of Web Pages from Table 4.4

1 http://www.tennis.org.uk/
2 http://www.tennismag.com.au/
3 http://www.tennisw.com/
4 http://tennis.org.uk/tennistoday/
5 http://alexia.lis.uiuc.edu/~kieraldo/tennis/rules.htm
6 http://www2.idsonline.com/jeff/jazz.html
7 http://www.acns.nwu.edu/jazz/education.html
8 http://www.jazzcentralstation.com/jcs/station/jazzdest/chicago/history.html
9 http://www.mc.maricopa.edu/its/lib/academic/lbt/lbt204/projects/jazz/jazz.html
10 http://www.yahoo.com/Entertainment/Music/Genres/Jazz/
11 http://www.volleyball.org/Index.html
12 http://www.volleyball.org/history.html
13 http://www.volleyball.org/whatis.html
14 http://www.volleyball.org/magazines.html
15 http://www.volleyball.ca/tablec/vcdir.html
16 http://www-dse.doc.ic.ac.uk/~nd/surprise_96/journal/vol2/cs11/article2.html
17 http://www.dacs.dtic.mil/techs/neural/neural10.html
18 http://www.teleweb.net/neuralnet/introtonn/index.htm
19 http://www.nd.com/welcome/whatisnn.htm
20 http://blizzard.gis.uiuc.edu/htmldocs/Neural/neural.html
21 http://www.nd.com/
22 http://www.ann-pro.com/
23 http://www.neuralfusion.com/
24 http://www.sensoryinc.com/
25 http://www.progno.com/default.htm

APPENDIX 14

URLs of Web Pages from Table 4.5

1 http://www.nd.com/

2 http://www.ann-pro.com/

3 http://www.neuralfusion.com/

4 http://www.sensoryinc.com/

5 http://www.progno.com/default.htm

6 http://www.lehigh.edu/~ay00/nn.html

7 http://www.loria.fr/exterieur/equipe/rfia/cortex/servers.html

8 http://www.cis.hut.fi/~jari/research.html

9 http://wwwsyseng.anu.edu.au/lsg/links.html

10 http://ai.iit.nrc.ca/subjects/Neural.html

11 http://www.nd.com/demo/demo.htm

12 http://www.scriptsoftware.com/index.html

13 http://www.unica-usa.com/download.htm

14 http://www.aist.go.jp/ETL/~akaho/MixtureEM.html

15 http://home.cc.umanitoba.ca/~umcorbe9/mlp.html

16 http://fuzzy.cs.uni-magdeburg.de/nfdef.html

17 http://peacock.pse.che.tohoku.ac.jp/~yyama/work/ann/section3.2.html

18 http://pcim.com/arc/conf/mexpo/11.html

19 http://www.tech.plym.ac.uk/soc/sameer/abstract/msc.htm

20 http://www.dreamlabs.com/~webland/c_p/icnn95/nn95s608.htm

21 http://www.cs.colostate.edu/~anderson/matlab-papers/rl/rl.html

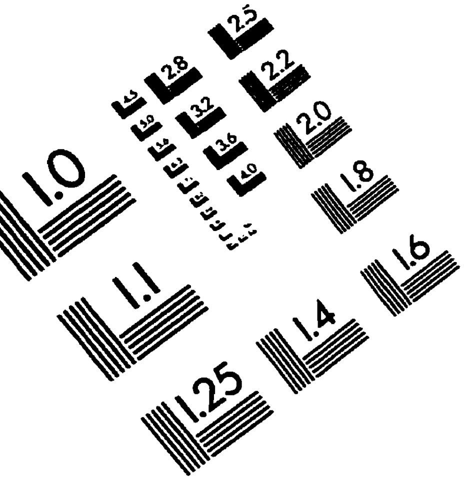
22 http://www.neuralware.com/products/proplus/proii5.html

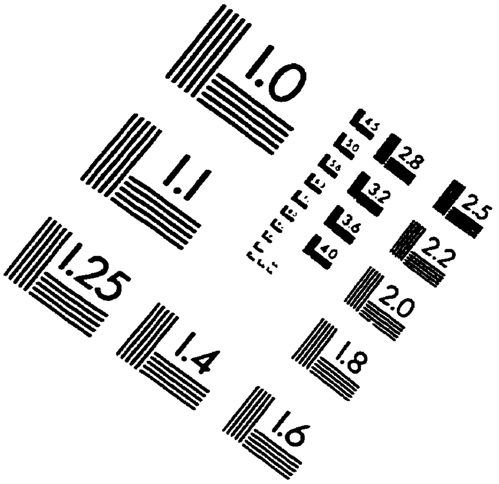
23 http://iserv.iki.kfki.hu/pub/icml96/szepes.genreinf.abs.html

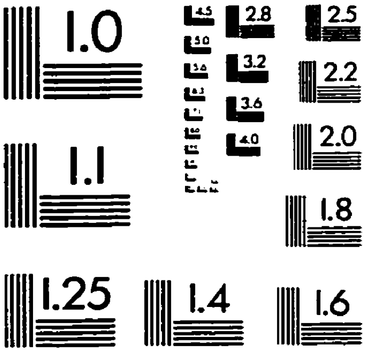
24 http://www.oup-usa.org:9200/9205/abs/ncc3.htm

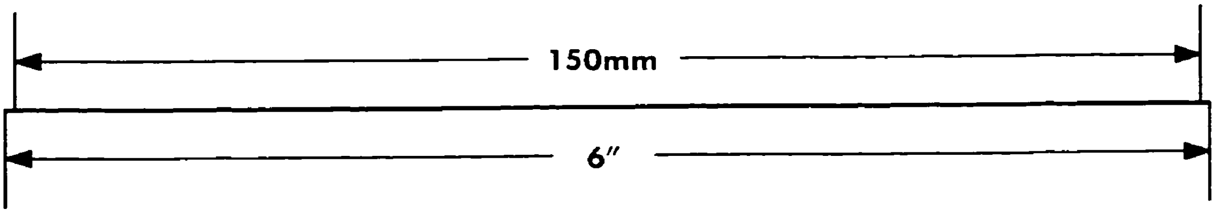
25 http://www.cs.cmu.edu/~reinf/web/cmu-rl-papers.html

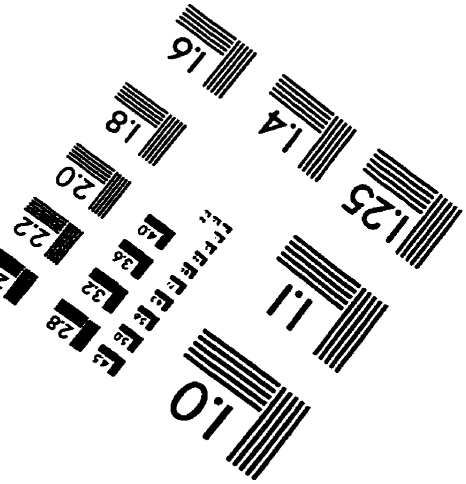
IMAGE EVALUATION TEST TARGET (QA-3)

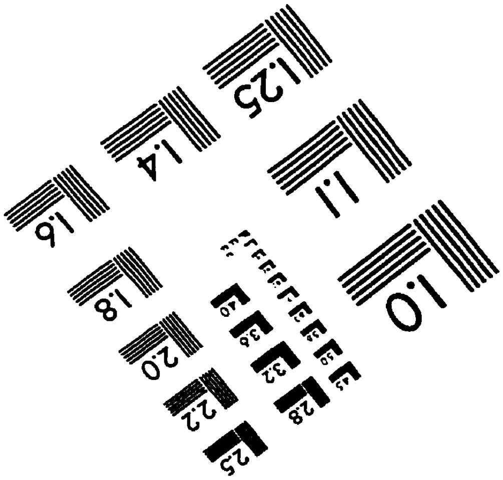












APPLIED IMAGE . Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989