

Similarity Measure Building for Website Recommendation within an Artificial Immune System

Tom Morrison

01973028

Dissertation submitted towards the award of
MSc in Statistics and Management Science

Please contact:

Uwe Aickelin
School of Computer Science
University of Nottingham
NG8 1BB UK
uxa@cs.nott.ac.uk

Abstract

A successful application of Artificial Immune Systems to the problem of recommending films to new users of a film database is used as the inspiration for an attempt to apply similar techniques to the problem of recommending web sites. Similarities and differences in the two situations are discussed as well as other approaches to recommendation problems such as collaborative filtering. The particular nature of the collected data necessitates several stages of processing followed by mapping it onto a web site classification database called DMOZ. The tree like structure of this database provokes the formulation of a tree similarity measure. The aim of this measure is to provide some measure of the similarity of two users that will provide answers when the Pearson correlation coefficient can not and that will take account of information categories that are 'close' to each other.

An attempt is made to adapt the Java code that runs the Artificial Immune System for the film data to process the web site data but it is unsuccessful. Instead the proposed similarity measure is tested using spreadsheet simulation leading to several amendments. The core of the problem lies in the interaction between the various components of the similarity measure. Whilst it seems impossible to include both 'vote' information and tree 'closeness' in the measure without getting contradictory results, a compromise is achieved by limiting the effect of the 'votes' on the overall measure. Further discussion and research into the proposed tree similarity measure, and, in particular, the examination of the effect of using the

proposed measure instead of Pearson's correlation coefficient on the web site data within an Artificial Immune System is needed.

Acknowledgments

Firstly I'd like to thank Steve Cayzer for his code and his patience and help in my efforts to understand it. What was attempted was over ambitious and it is no reflection on either of us that it was not possible in such a short time.

Secondly I'd like to thank Uwe Aickelin, for his teaching, for the original idea and for getting me to my first ever academic conference. I have thoroughly enjoyed the challenge of this research and discussing the ideas involved in it. I hope that there is some 'honey' in this work, and that he is able to find someone to pursue it further. It has been a pleasure working with both of these men.

Lastly I'd like to thank my family, Reuben and Elliot for putting up with my disappearing upstairs to work, and my wife Hyacinth. Her support and encouragement were critical to the completion of the research, and now that it is finished, she will finally get her partner back.

Table of Contents

CHAPTER 1 INTRODUCTION	6
1.1 Introduction	6
1.2 Similarities and differences	7
1.3 The structure of the report	8
CHAPTER 2 LITERATURE REVIEW	10
2.1 Introduction	10
2.2 The Biological Immune System	10
2.3 Artificial Immune Systems	15
2.4 Artificial Immune Systems Applied to Recommendation Problems	16
2.5 Collaborative Filtering and Similarity Measures	19
2.6 The DMOZ database	22
2.7 Summary	26
CHAPTER 3 ACQUIRING THE DATA AND THE AIS CODE	27
3.1 Collecting the data	27
3.2 Processing the data	28
3.3 Building the Artificial Immune System	32
CHAPTER 4 THE SIMILARITY MEASURE	34
4.1 Preliminary discussion	34
4.2 Principles for tree similarity measure construction	35
4.3 Examples	36
4.4 The effect of the edge distance	37
4.5 The effect of the level	38
4.6 The initial version of the similarity measure	39
CHAPTER 5 THE PROCESS OF THE INVESTIGATION	41

5.1	Processing the data	41
5.2	Refining the similarity measure	44
5.3	The amended measure	46
5.4	Change of direction	46
5.5	Using a spreadsheet simulation to test the similarity measure	47
5.6	Adjusting the balance of the measure components	50
5.7	A radical redesign	56
5.8	Summary	61
CHAPTER 6 CONCLUSION		63
6.1	The results in context	63
6.2	Further work	63
6.3	Conclusion	64

CHAPTER 1 INTRODUCTION

1.1 Introduction

The idea for this research was generated from a piece of research by Cayzer and Aickelin (2002) that used an artificial immune system (AIS, also used for the plural) to recommend films to new users of a film database. Each user had rated a number of films and the research showed that once a new user had rated just a few films it was possible to predict the new users rating of other films with similar accuracy to other techniques, as well as to recommend new films to the user with greater accuracy than comparable techniques. Having experienced this success with films the interest was to see if the technique could be generalised to other examples, in particular to web site recommendation.

Recommendation is the process of suggesting a new item to somebody who has not experienced the item before. This is a different sort of problem than predicting the rating a person might give to an item. There are, obviously, a number of web sites that will search for new web sites according to criteria set up by the user. However, these methods will only find web sites that are within the boundaries of the search that was set up. The idea of recommendation is that a new web site would be suggested that the user would not have searched for but that the user might like. One of the particular strengths of the research by Cayzer and Aickelin (2002) was that the group of people who were chosen to generate the recommendation for the new user were similar to this user but diverse, thus

leading to a greater potential to suggest something likeable but different for the user. Could this approach locate new websites that a user would particularly enjoy but that they would not have specifically searched for?

1.2 Similarities and differences

Both films and websites generate differing responses for different people. They have content and design, and cover a wide range of subject matter. However, it is possible to place the majority of films into a fairly small number of genres (action, romantic comedy etc), whilst any classification of web sites would struggle to categorize the majority of its members into such a small number of groups in any sensible way. Also, the rating of films is a familiar and frequent event, leading to the raw data for the film research mentioned above. However, there does not seem to be a collection of people's ratings, on a numerical scale, of web sites. This would mean that the data would have to be collected from scratch to be able to perform this research.

Films, once released, do not normally change appreciably over time (apart from a Director's cut or other revision). Films are remade, but then often have different names, or it is clear which version is being discussed. The world wide web, however, is constantly evolving. Web sites appear and disappear, their content is updated, and their connections to other sites change. Any system that attempted to use such information would need to be dynamic in itself.

The last difference is that the world wide web is more concerned with information whereas films are generally concerned with artistic expression. Obviously documentaries are meant to deal with information but these are a small fraction of the number of films whilst the majority of web sites deal with knowledge. This means that web sites can be classified using some epistemological system. Thus there is a potential structure of the web sites that might be useful in producing recommendations.

1.3 The structure of the report

In order to examine whether the film recommendation research is generalizable to web sites a number of steps need to be taken. In chapter 2, the review of literature, an examination of artificial immune systems is conducted to ensure full understanding. The nature of the data and its potential structure leads to the examination of the DMOZ (Directory Mozilla – <http://dmoz.org/>) web site classification database. Other approaches to recommendation and the possibility of a new similarity measure that utilises the information structure are also considered in this chapter.

Chapter 3 is concerned with the problems of acquiring and pre-processing the data, as well as building the artificial immune system, whilst chapter 4 looks at the building of a new similarity measure from scratch.

In chapter 5 the actual progress of the investigation is described. The new challenges thrown up by the data are described and the decision to abandon the attempt to build the artificial immune system. In its place further examination of the proposed similarity measure is attempted by the use of spreadsheet simulations. Then in chapter 6 there is a discussion of the results produced from the simulations followed by summary and conclusions.

CHAPTER 2 LITERATURE REVIEW

2.1 Introduction

This review of literature starts by describing key aspects of the biological immune system in order to explain artificial immune systems. Then the application of artificial immune systems to recommender problems is considered. An alternative approach, known as collaborative filtering is discussed in section 2.4 along with an examination of typical similarity measures. Finally the DMOZ database is briefly described in order to make clear some of the issues and questions that will need to be addressed in chapter 3.

2.2 The Biological Immune System

In order to describe an artificial immune system it is necessary to have some understanding of our immune system. Much of the following description of the human immune system is based on *Artificial Immune Systems: Part 1 – Basic Theory and Applications* by de Castro and Von Zuben, 1999.

Protection of the human body against foreign invaders is achieved by a multi layered system composed of physical barriers such as the skin and respiratory system; physiological barriers such as destructive enzymes and stomach acids; and the immune system which has two complementary parts, the innate and

adaptive immune systems. The innate immune system is an unchanging mechanism that detects and destroys certain invading organisms, whilst the adaptive immune system responds to previously un-met foreign cells and builds a response to them that can remain in the body over time. The immune system is composed of a number of different types of agent performing different functions at a number of different locations in the body. The precise interaction of these agents is still a topic for debate. In order to present the important aspects of the system from a mathematical viewpoint it is necessary to present a simplified and selective description.

The immune system's job is to detect antigens, which are foreign molecules from a bacterium or similar invader. The innate immune system helps in the detection process but the main response is through the adaptive immune system. Two of the most important cells in this process are white blood cells called T cells and B cells. Both of these originate in the bone marrow but T cells pass on to the Thymus to develop before, as with B cells, they circulate the body in the blood and lymphatic vessels.

The T cells are of three types; T helper cells which are essential to the activation of B cells, Killer T cells which bind to foreign invaders and inject poisonous chemicals into them causing their destruction, and suppressor T cells which inhibit the action of other immune cells thus preventing allergic reactions and autoimmune diseases.

B cells are responsible for the production and secretion of antibodies, which are specific proteins that bind to the antigen. Each B cell can only produce one particular antibody. The antigen is found on a site on the surface of the invading organism and binding of antibody to antigen is a signal to destroy the invading cell. A diagram (from de Castro and Von Zuben, 1999) of this process is shown in figure 1 below.

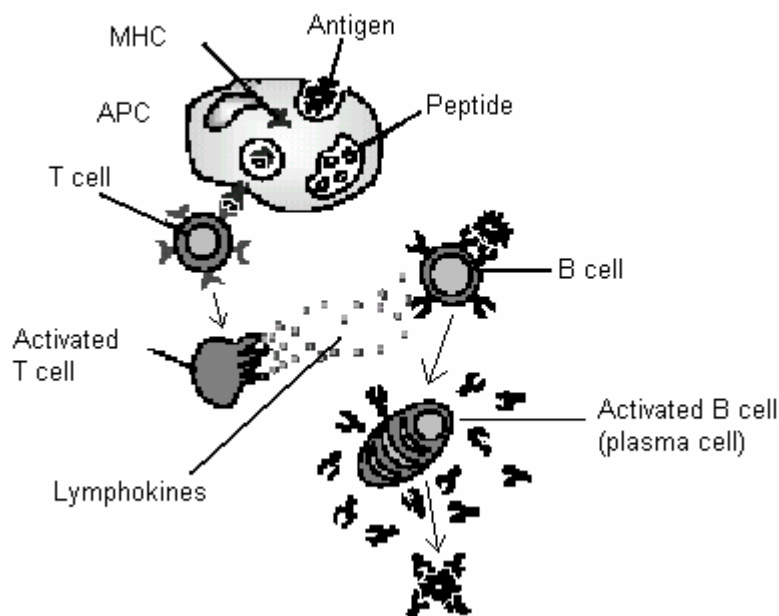


Figure 1 The processes involved in the immune response

The top of the diagram shows a cell known as a macrophage ('big eater' or antigen presenting cell, (APC)) taking in the antigen, breaking it down into peptides, combining the fragments with major histocompatibility complex molecules (MHC) before displaying it on the surface of the cell. When a T cell comes into contact with this peptide-MHC combination it is activated and it releases lymphokines (a signal) which activate a B cell. Note that the B cell can also be activated directly by antigens in solution. When a B cell is activated it

becomes a plasma cell that manufactures the specific antibodies for that antigen. The antibodies bind to the antigens triggering the destruction of the original invader. Then, as de Castro and Von Zuben (1999) comment, “some T and B cells become memory cells that persist in the circulation and boost the immune system’s readiness to eliminate the same antigen if it presents itself in future.” This is the basis of immunisation.

The biological immune system is very complex, containing a number of different agents and processes, not all of which are fully understood. Artificial Immune systems attempt to model or use some elements of the biological system. In particular two processes, known as the ‘clonal selection theory’ (Burnet, 1959) and the ‘idiotypic network theory’ (Jerne, 1974), are particularly important in this research.

When an antibody strongly matches an antigen the corresponding B cell is stimulated to produce clones of itself that then produce more antibodies. This selection of B cells for cloning on the basis of the antibody match is called the ‘clonal selection principle’ and will result in increasing concentrations of that antibody in the body.

However, when the B cell clones itself it does not do so exactly, it will mutate slightly. B cells may be stimulated when the antibody-antigen match is not perfect. By allowing mutation the match could become better. However, a large number of worse matches will be created, and further, some of the newly produced antibodies could be harmful to our own cells. Such cells will die out

under what is known as the 'negative selection principle'. Forrest et al (1994) proposed that this principle of "self-nonsel self discrimination" could be used in virus detection in computer networks. This is an example of using just a small part of the biological immune system that was then extended further by Hofmeyr and Forrest (1999) and also Kim and Bentley (2001). However, even in these more complex systems only a fraction of the functionality of the biological immune system is exploited.

The mutation, mentioned above, is quite rapid, often as much as "one mutation per cell division" (de Castro and Von Zuben, 1999). This allows a very quick response to the antigens. This rapid mutation, known as 'somatic hypermutation', may be linked to the 'fitness' of the antibody, ie those B cells producing antibodies that are a good match would be subject to less mutation than ones that were not such a good match. The antibody-antigen interaction coupled with somatic hypermutation, form the basis for much of the AIS applications and research. Examples are Timmis et al (2000) who used an AIS for clustering multivariate data, and Hajela and Yoo (1999) who combined a genetic algorithm and an AIS to optimise the design of a 10 bar truss.

The idiotypic network theory, Jerne (1973), maintains that interactions in the immune system do not just occur between antibodies and antigens, but that agents of our system may interact with each other. This theory could help to explain how the memory of past infections is maintained and could result in the suppression of similar antibodies thus encouraging diversity in the antibody pool.

2.3 Artificial Immune Systems

This last possibility was used in the research by Cayzer and Aickelin (2002) in order to preserve diversity. The AIS in their research produced a pool of users who were similar to the new entrant to the database, but dissimilar to each other. Whilst this method produced similar performance in predicting film ratings to a k-nearest neighbour approach, the advantage of diversity in the pool of recommenders was found to be significantly improved recommendation success. Given the sparseness of the web site search space it may be that suppression of antibodies on similarity grounds might be unnecessary or counterproductive but this can be assessed by appropriate tests on the parameters involved.

There are a number of successful Artificial Immune System implementations. However, even in the most complex artificial systems only a fraction of the functionality of the biological immune system is exploited. Typically, the antibody-antigen interaction coupled with somatic hypermutation, form the basis for many Artificial Immune System applications. The research by Timmis et al (2000) also applied the idiotypic network theory and was successful both in classifying data and “generalises to cover a larger region of the input space”. However, the article does not comment on the effect of modelling a suppression factor between antibodies as well as a reinforcement factor. Some of the most promising research to date has been conducted in the area of computer security, for instance by Hofmeyr and Forrest (2000) in computer network security and by Kim and Bentley (2001) for fraud detection.

2.4 Artificial Immune Systems Applied to Recommendation Problems

Whilst most of the applications described above involve somatic hypermutation, Cayzer and Aickelin (2002) had only identical cloning, not mutation, in their algorithm. This was because the potential antibodies were actual users of the film database (Eachmovie database provided by the Compaq Research Centre). There the task was to find users that were similar to new entrants to the database. Somatic Hypermutation was not used, since it is not immediately obvious how to mutate users sensibly such that these artificial entities still represent plausible profiles.

For the same reasons, cloning in the intended Artificial Immune System will make exact copies, too. Future work might include making inexact copies to create novel profiles once appropriate rules for doing so have been established. This could be particularly beneficial when data gathering is expensive or data is otherwise sparse, perhaps due to its sensitive nature, leading to few users being willing to share their information with others.

The main loop of the recommender algorithm is shown below and is the core of the proposed Artificial Immune System. The aim of this algorithm is to increase the concentrations of those antibodies (database users) that are similar to the antigen (target user). This process is subject to the suppression of similar antibodies following Jerne's (1973) idiotypic ideas mentioned above. Thus, over

time the Artificial Immune System contains high concentrations of a diverse set of users who have similar film preferences to the target user.

```
Initialise AIS
Encode user for whom to make predictions as antigen Ag
WHILE (AIS not stabilised) & (More data available)
DO
  Add next user as an antibody Ab
  Calculate matching score between Ab and Ag
  Calculate matching scores between Ab and other antibodies
  WHILE (AIS at full size) & (AIS not stable)
  DO
    Iterate AIS
  OD
OD
```

Figure 2 AIS pseudo code

The diagrams in figure 3 below show the idiotypic effect. In the top diagram antibodies Ab1 and Ab3 are very similar and they would have their concentrations reduced in the 'Iterate AIS' stage of the algorithm above. However, in the lower diagram, the four antibodies are well separated from each other as well as being close to the antigen and so would have their concentrations increased.

At each iteration of the film recommendation AIS the concentration of the antibodies is changed according to the formula given on the next page. This will increase the concentration of antibodies who are similar to the antigen and can allow either the stimulation, suppression, or both, of antibody-antibody interactions to have an effect on the antibody concentration. More detailed discussion of these effects on recommendation problems are contained within Cayzer and Aickelin's (2002) paper.

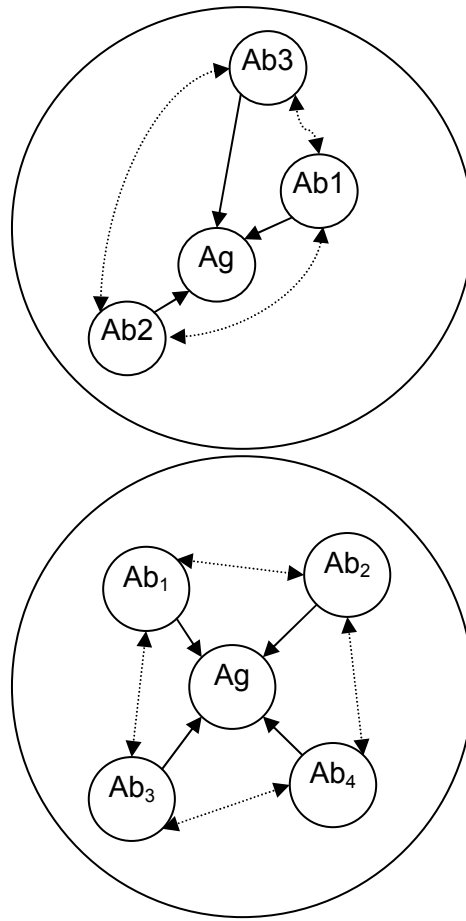


Figure 3 :Illustration of the idiotypic effect.

The formal equation for the idiotypic effect adapted from eqn 3 in Farmer (1986):

$$\frac{dx_i}{dt} = c \left[\left(\frac{\text{antibodies}}{\text{recognised}} \right) - \left(\frac{I \text{ am}}{\text{recognised}} \right) + \left(\frac{\text{antigens}}{\text{recognised}} \right) \right] - \left(\frac{\text{death}}{\text{rate}} \right)$$

$$= c \left[\sum_{j=1}^N m_{ji} x_j x_j - k_1 \sum_{j=1}^N m_{ij} x_i x_j + \sum_{j=1}^n m_{ji} x_i y_j \right] - k_2 x_i$$

Where:

N is the number of antibodies

x_i is the concentration of antibody i

m_i is the correlation between antibody i and the (sole) antigen

m_{ij} is the correlation between antibodies i and j

y is the concentration of the (sole) antigen

k_1 is stimulation, k_2 suppression and k_3 death rate

The algorithm is terminated when the Artificial Immune System is said to have stabilised, i.e. if it has not changed in consistency for more than ten iterations. The concentrations and correlations of the users in the final neighbourhood, i.e. final immune system iteration, are then used to calculate a weighted sum of the ratings of web sites. This would be either a specific unseen web site by the target user in order to predict its ratings, or general top 10 recommendations of new web sites that the target user might enjoy.

2.5 Collaborative Filtering and Similarity Measures

There are a number of algorithms that recommend items to users. One of the best known examples is Amazon.com's book recommender based on similar items bought (<http://www.amazon.com>). Generally, these recommenders use what is termed "collaborative filtering" or "social filtering" by Billsus and Pazzani (1998). With the exponential growth of available information on the internet, the need for automated techniques to winnow down the possibilities has also grown but "only a few different algorithms have been proposed in the literature thus far" (ibid).

Many of the current collaborative filtering techniques use the Pearson correlation coefficient to compare the item ratings of different users. This suffers from several limitations. For example, due to the extremely large amount of information to be rated, two users may only have a very small number of items in

common causing the correlation measure to be unduly influenced by those items. Further, there is potentially no difference between the correlation between two users with 3 items in common and the measure for two users with 30 items in common, in terms of their “influence on the final prediction” (ibid 1998).

The sparseness of the information space also implies that two users might have no items in common. Can we therefore conclude that they have completely dissimilar tastes, or does the fact that they have not rated particular items imply a similar view of the importance of those items? For these reasons, alternative approaches to both current collaborative filtering algorithms and to the use of the Pearson correlation coefficient should be investigated. In particular, similarity measures that exploit the tree like structure of web site classification systems would be useful. More information about traditional collaborative filtering and how to improve it is provided by Gokhale (1999).

In both library and internet searches I was unsuccessful in finding any similarity measures relating to tree structures. I was hopeful that in areas like biology or graph theory that there would be some use for such a measure and that there would, therefore, be some work that I could refer to. The graph theory concept of minimum spanning distance is used in gene expression data clustering, for example the ‘EXCAVATOR’ software and associated article by Ying et al (In press) which can be found at the URL

<http://compbio.ornl.gov/structure/excavator/command.html> .

They define a number of potential similarity measures for use with their application as shown in the list in figure 4

- "-dist 1" or no "-dist" flag (default): $(1 - \text{correlation coefficient})$.
- "-dist 11": $(1 - \text{square of correlation coefficient})$.
- "-dist 12": $(1 - \text{absolute value of correlation coefficient})$.
- "-dist 2": Euclidean distance.
- "-dist 21": square of Euclidean distance.
- "-dist 3": sine square of the angle between two vectors.

Figure 4 Ying et al's list of possible similarity measures

However, all of these measures are familiar ones that make no use of the shape of the associated tree structure.

Noel et al (2002) note that minimum spanning trees have been used as a way of visualising document collections using co-citations to quantify the distance between documents. In the visualisation “branches in the tree correspond to bifurcations of ideas in the evolution of science, with highly influential documents appearing near the center of the network, and the emerging research front represented as documents on the fringes”. Whilst there are some similarities between this work and what I am proposing, one of the key differences is that they are dealing with an “undirected graph” whilst a structure of knowledge that starts with a root and then breaks down into topics and subtopics is hierarchical and therefore inherently directional. They compare the co-citation count and correlation coefficient as measures within algorithms used to represent the document relationships in two dimensions. So they are trying to infer a structure from a measure, rather than trying to measure ‘distances’ in a previously defined structure.

2.6 The DMOZ database

In our problem of web site recommendation, the original data consists of sets of web site addresses or uniform resource locators (URLs). It is extremely unlikely that a set of people will have many exact addresses in common within their web profiles. Because of this, it is necessary to transform or translate the addresses into a different form. To do this a number of steps are necessary and a widely used web site classification database, called DMOZ, will be used.

Let us look at the issues involved in the classification of URLs systematically.

Typically, an individual web profile in raw form might consist of a list of bookmarks as shown in Figure 5 (in this case taken from the Opera browser – only a small section is shown).

```
#URL
NAME=ODP - Open Directory Project
URL=http://dmoz.org/
CREATED=1017158736
VISITED=1023875733
ORDER=2147483647
DESCRIPTION=
SHORT NAME=
```

```
#URL
NAME=Open Directory RDF Dump
URL=http://dmoz.org/rdf.html
CREATED=1017159133
VISITED=1023875759
ORDER=2147483647
DESCRIPTION=
SHORT NAME=
```

Figure 5: Part of a raw web profile taken from the Opera browser.

This data has to be pre-processed in order to remove superfluous characters. This also includes removing any categories the user might have assigned to some of the bookmarks. Unfortunately, such categorisation of information cannot be kept as it is arbitrary and individual to the person that owns the bookmarks. For instance, www.bbc.co.uk could be classified under 'media' by one person and under 'news' by another. Hence, this filtering typically yields a file such as the one partially shown in Figure 4.

1. www.bbc.co.uk/weather/
2. www.bbc.co.uk/
3. www.bbc.co.uk/sport/hi/english/football/default.stm
4. www.guardian.co.uk/
5. football.guardian.co.uk/

Figure 6 Part processed data with superfluous information deleted.

As can be seen from the third line in Figure 6, some of the URLs will have quite long addresses. Another web profile might contain a very similar address such as www.bbc.co.uk/sport/hi/english/football/eng/default.stm. If we were to use the raw addresses within the Artificial Immune System, these two would be considered different. However, it is clear that the two users have bookmarked different pages within the same part of the same site, i.e. 'BBC online - football', and thus have very similar interests.

Therefore, it is still necessary, although difficult, to process the data before it can be used. A program will need to be devised which will process the URLs in such a way so that the two addresses discussed above would be considered the same. However, looking again at Figure 6, a simple truncation would lead to the first

three items occupying the same category. At the same time, it might not lead to the last two being picked together despite the fact that both the addresses refer to pages from the same site. Furthermore, it might not put 3 and 5 together despite the fact that they are both about football.

To overcome problems of misclassification and to have a common standard we decided to use the DMOZ open directory (<http://dmoz.org>) as a classification system. Figure 7 shows part of the structure of this directory.

```
<Topic r:id="Top/Arts">
<tag catid="2"/>
<d:Title>Arts</d:Title>
<narrow r:resource="Top/Arts/Books"/>
<narrow r:resource="Top/Arts/Music"/>
<narrow r:resource="Top/Arts/Television"/>
<narrow r:resource="Top/Arts/Writing"/>
<narrow r:resource="Top/Arts/Animation"/>
<narrow r:resource="Top/Arts/Anime"/>
[...]
<Topic r:id="Top/Kids_and_Teens/Pre-School">
<catid>468769</catid>
<link r:resource="http://www.coolplays.com/">
<link r:resource="http://kayleigh.tierranet.com/">
<link r:resource="http://www.megafire.com.br/">
<link r:resource="http://www.123child.com/act/">
<link r:resource="http://www.peterrabbit.com/">
<ExternalPage about="http://www.coolplays.com/">
<d:Title>Coolplay's Cool for Kids</d:Title>
<d:Description>Includes animated nursery rhymes, crafts, alphabet and
spelling games, and colouring book.
```

Figure 7 Part of the DMOZ open directory structure.

The first half of Figure 7 shows part of the 'Arts' category, which is immediately below the root of the tree (called Top). Each category has a unique identifier number (2 in this case). This category has a number of sub categories that in turn have several sub categories of their own. In total, there are some 5 million URLs

in 428,590 categories spread over 16 levels in the directory. Categories can also be referred to using an address showing the parent categories in a way that preserves the tree structure information. For example, a category address might read '1.3.9' meaning that it is the ninth sub category of category 3 which is the third sub category of category 1.

The second half of Figure 7 shows how URLs are represented in DMOZ and gives an example of a more detailed description of one URL as provided by an anonymous referee. As can be seen each category will contain a number of URLs and it is by comparison with these that a URL will be mapped on to the database. The complete DMOZ database is roughly one Gigabyte in size and updated regularly. All specifications in this paper refer to DMOZ as of 1 June 2002. Overall, the version of DMOZ that we use has the tree structure shown in figure 8 below with the deepest branch being 16 levels below the top:

Number of categories at the level	Level
1	0
18	1
621	2
6675	3
30754	4
61042	5
68901	6
101567	7
82802	8
51454	9
20592	10
3467	11
616	12
69	13
8	14
2	15
1	16

Figure 8 Full DMOZ structural tree.

2.7 Summary

The literature reviewed here illustrates that AIS can be used with great effect to find a group of matching users to a new user within a film database and hence to recommend films that the new user might enjoy. Alternative recommendation techniques such as collaborative filtering are discussed as well as the need for a new similarity measure that might exploit the hierarchical tree structure of the database chosen to represent the data. The nature of the data and some of the problems in preparation are considered as well as the chosen representation, the DMOZ database. The questions raised by the review are whether a using an AIS will allow web site recommendation in a similar manner despite the different nature of the data, and whether using a different similarity measure than Pearson's correlation coefficient would help in the working of the AIS.

CHAPTER 3 ACQUIRING THE DATA AND THE AIS CODE

3.1 Collecting the data

As this research is rooted in the positivistic paradigm it is necessary to use a large sample of data. The data is a set of website addresses for each person (web profiles) as mentioned in the last chapter. There was no way of knowing in advance how many people's web profiles would be needed but, in discussion with my supervisor we estimated that at least 1000 web profiles were necessary. A sample of this size would allow the holding back of a subset to use for testing. An alternative way of testing would be to split some web profiles in half and use one 'half' to try to find the other.

The data was collected by a number of means. The principle one was to send an email requesting that individuals forward their web profiles to a specially set up web site. The email was sent to all staff and students at the University of the West of England and to a number of other mail lists. It was also possible to access a number of personal web pages which contain collections of bookmarks. In all cases, as soon as the web profile was collected, all connections with the person donating the profile were erased and a unique number was substituted. A copy of the email used is contained in appendix A.

The danger with collecting data in this way is that it has either been volunteered or removed from the world wide web. By seeking volunteers one is prevented from getting a random sample since the volunteering process introduces bias.

However, in this case since the research is about proof of concept then this should not be too much of a problem, although it may mean that we receive slightly 'fuller' web profiles than is the case in general. One criticism might be that I have 'taken' peoples lists of bookmarks from the web, but since placing them there is akin to publishing them without copyright, I believe that this should be acceptable.

3.2 Processing the data

Some description of the data is contained in chapter 2. As was stated it was necessary to pre-process the data to remove superfluous characters and user classifications. The file of web site addresses was then mapped onto the DMOZ database so that each URL was replaced with a category identification number (or category address) and a 'vote'. This was not always possible but two strategies were used within the DMOZ ontology to achieve this. They were, normalisation, and reverse partial look-up. First, all URLs undergo a kind of normalisation when pre-formatting the data, as well as when doing look-ups. The protocol and host part are mapped to lowercase characters and host only URLs are always terminated with a "/". During the actual look-up to gain the category information from DMOZ a reverse truncation search is employed. That is, at first we try to match the full URL, then we try to match up to the last "/", then to the last but one

“/” etc. For instance, we would first try to match ‘www.bbc.co.uk/sport/hi/english/football/default.stm’ by looking for the full URL in DMOZ. If we cannot find that, we would look for www.bbc.co.uk/sport/hi/english/football/, if this fails we would search for www.bbc.co.uk/sport/hi/english/ etc.

There are a number of possible pitfalls with this process. For example, many profiles will contain a set of URLs, which are created by the browser program that they use. Few users are likely to delete all of these links, reasoning that they may be useful at some stage. This may create a situation of artificial similarity between users, which would prevent the Artificial Immune System from functioning effectively.

Secondly, the process of placing URLs into categories is likely to involve some truncation if at first there is no clear category involved. This could lead to several subtly different addresses being classified into the same category due to the truncation look-up. Depending on whether the truncated sites are from genuinely different URLs or not this could be good or bad. In the first case, the category may appear to be more popular than it should be whereas in the second case the number in the category is a clear indication of interest in that category. Until the data is fully assembled, individual examples are checked, and the Artificial Immune System is constructed, it will not be possible to judge how critical some of these problems will be.

In the film recommendation research, described in Cayzer and Aickelin, each user was coded as a user identification number followed by pairs of film identification numbers with the corresponding rating of the film. The target user became the antigen, whilst the current database members were potential antibodies. In each iteration, antibodies were added to the Artificial Immune System and those judged to be more similar in their film ratings, using a variation of Pearson's similarity measure, had their concentration increased.

A unique feature of that particular approach was the application of the idiotypic network theory by Jerne. This was implemented such that antibodies which were very similar to each other had their concentration reduced. This has the effect of creating a set of users who are similar to the new user but quite different to each other and thus enhancing the recommendation accuracy of the system. The intention is to use the same mechanism for the web site recommender.

In order to do this, we also have to decide on the encoding of a user's web profile, for which there are two possibilities. In both cases, a user is encoded as a list of category IDs and the number of bookmarks within each category. The difference is in the category IDs; they can be either an integer or a reference to the tree structure. To illustrate the difference, Figure 9 shows the same user's bookmarks for both encodings. The figures in bold indicate how many bookmarks fall into a particular category:

Encoding with the Tree structure:
1.13.12.1.5:**5**;
1.13.12.1.6:**3**;
1.16.3.2.11.5:**1**;
1.18.1.2:**1**;

Encoding with integer category IDs:
22343:5;
495771:3;
334921:1;
3409:1;

Figure 9 Integer versus Tree Encoding.

If the second encoding is used together with the number of sites within each category as a rating of the popularity of that category then the problem becomes similar to the film recommendation problem.

However, it will have a considerably sparser search space. In the film database, there were approximately 20,000 entries whereas in the DMOZ directory there are over 400,000 categories. This sparseness may prevent the system from working since many users might have nothing in common, or, at best some categories that are common to the vast majority of the data. Further, it is possible that many users will have only one entry in a large number of categories, leading to increased similarity since the 'rating' of that category will be the same. These problems may prevent an Artificial Immune System based on this encoding being successful in identifying a group of similar users.

Furthermore, there is another problem with using integer category IDs. Because DMOZ is an evolving classification system, new categories are added and removed regularly. This can have the effect that two very similar categories end up with very different integer IDs as these are handed out consecutively. For instance, Star Wars part four might have ID 20,004 when it was classified years ago, but Star Wars part two might end up with ID 420,012 because it has only

recently entered the DMOZ system. A similar effect can be seen in Figure 7 for the first two bookmarks. At the same time Figure 7 shows that using the tree structure IDs might prevent some of these problems as similar categories that have been added to DMOZ at different times still end up near each other in the tree.

The alternative to the integer encoding is to use the encoding which includes the tree structure in the form of a category address. This might allow the construction of a different similarity measure that would recognise categories that are 'close' within the structure to be judged more similar. For example, it would need to judge the parent / child or the sibling relationship as being more similar than a first cousin or grandparent type relationship. However, construction of such a measure is far from simple and is considered in chapter 4.

3.3 Building the Artificial Immune System

The artificial immune system will be based on code written by Steve Cayzer for the film recommendation problem. This is a set of classes written in Java which interact to run the AIS. The code includes a Graphical User Interface with which the user can make choices about the parameters involved in the system, start and stop the process, and view what is happening in the system as it changes. The code is a sophisticated and generalizable structure that should allow the application of artificial immune systems in different circumstances and with different data.

However, in order to amend the code so that it will process the web site information a number of steps are needed. Firstly I needed to learn how to read and use Java code. Secondly the code could be simplified considerably; there is no need for a graphical user interface, the system could be applied to text files rather than a database and the data itself would be simpler as there would be no genre information (eg horror, comedy etc), merely addresses/identification numbers, and 'votes'. The third stage, having rebuilt the AIS code would be to test the system by adjusting the parameters, and also, if possible, using the alternative similarity measure discussed in chapter 4.

CHAPTER 4 THE SIMILARITY MEASURE

4.1 Preliminary discussion

Consider the two simple trees below in Figure 10. User 1 has entries at categories G, E, J and L, whilst user 2 has entries at D, I, J and F. We have to consider how the entries at the different categories (branches or twigs) within the structure should contribute to a measure of how connected the two users are. Clearly, matches should be scored more highly the lower down the tree they are because this indicates a more precise match (the categories of information are more specific). Additionally, 'close' relationships within the tree structure (eg G to H or I to E) should count more towards the match than ones separated by several 'generations' (to use a family tree metaphor).

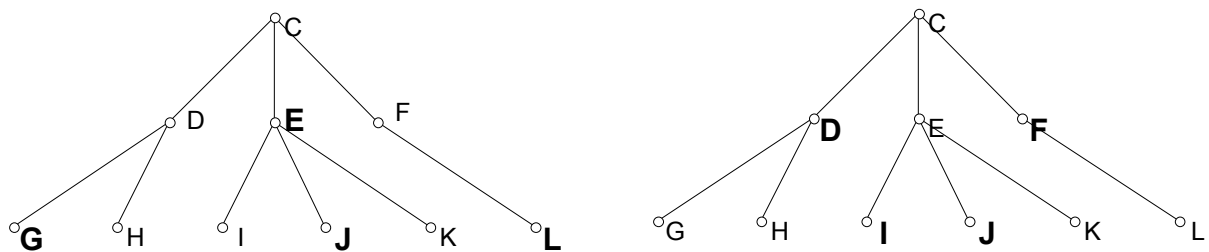


Figure 10: Simple tree structure showing two web profiles.

Whilst it seems to make sense that, since both of these users have an entry in category J, they should have their similarity measure increased, a question remains what to do with J afterwards. Should this match be discarded once it has been counted by the measure or should the entries at I and J for user 2 be

counted as 2 entries at the parent branch (E) for comparison with user 1? The danger with discarding matches once counted is that two users might have 'perfect' matches for all of the 10 categories that the first user has in their profile, whilst the second user has another 100 entries.

However, if one does not discard categories that have already been matched with another category then it is possible that one quite high level category might be 'matched' with all the different entries at sub categories for another user. This might not matter since the 'strength' of the match would have been reduced by the generational distance and the weakness of the high level category's contribution.

4.2 Principles for tree similarity measure construction

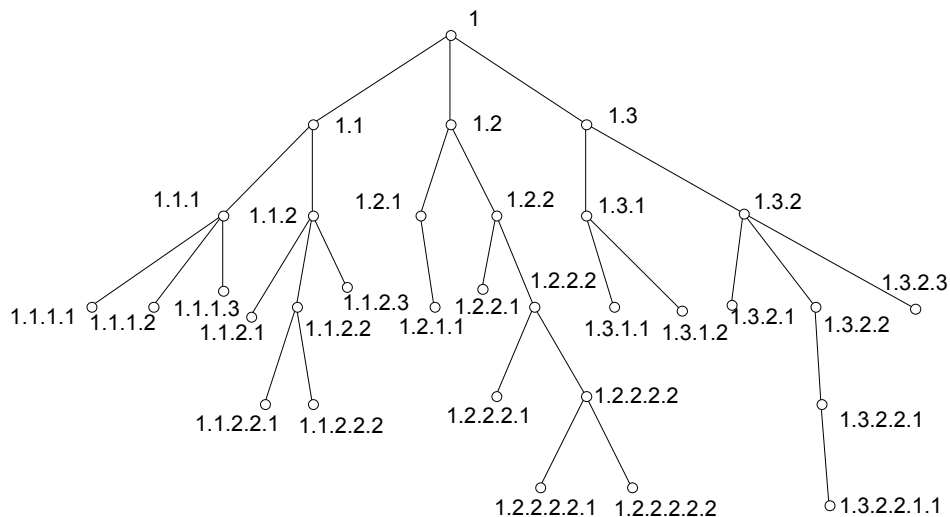
After consideration of the above discussion the following principles seem important in constructing a tree based similarity measure:

1. Matching at categories lower down the tree structure should contribute more to the measure than matching higher up.
2. Matches at the top level of the tree (i.e. the 'Top' category in the DMOZ database should have a contribution of zero.
3. Matching contribution should be reduced for 'imperfect matches' i.e. those not in exactly the same category. The reduction in contribution should be proportional to the generational distance (i.e. a grandparent child relationship has a generational distance of 2.)

4. The matching metric should be scaled (averaged) so that it ranges from 0 to 1.
5. The matching metric should take into account all possible matches between the entries in each web profile, i.e. if there are 10 entries in 1 and 20 in the other then all $10 \times 20 = 200$ potential matches should contribute to the measure.

4.3 Examples

Taking an example, suppose that we wish to calculate the matching coefficient for the category addresses 1.3.1.1 and 1.3 in the sample tree diagram below. We need to define an 'edge distance' as the number of 'steps' apart any two addresses are. For example, 1.1 and 1.1.2.2.1 have an edge distance of 3, as do 1.2.2.2 and 1.2.1. This equates the relationship between grandparent and grandchild as the same strength as that between siblings.



NB All the categories roughly on a line are at the same level but are shown this way in order to fit in their labels
i.e. 1.1.2.3 is on the same level as 1.2.1.1

Figure 11 A more complicated tree structure

By staged truncation of the longer category address (CA) until they are the same we would have a match at CA 1.3 with two numbers discarded (but counted as an edge distance of 2). This match would have a strength determined by the category level (level 2) of the matching CA, and by the edge distance (2).

Now consider the CAs 1.1.2.2 and 1.1.2.3. In this case they are both the same length but do not match. Either one can now be truncated first before using the same routine as above. These would match at 1.1.2 (level 3) and two edges would have been discarded before the match was found. Such a match should be stronger than the one discussed above since it occurs at a lower level even though it has the same ED.

4.4 The effect of the edge distance

How should the edge distance affect the value of the overall match? One possibility would be to use $\frac{1}{ED}$ as this would be a smaller value as the ED increases. However, this would not work when the CA match perfectly as we would be dividing by zero, and the value for this match should be the highest ie 1. Using $\frac{1}{ED+1}$ would solve these problems, although the contribution this makes to the overall measure is not linear and the effect of this will have to be considered.

4.5 The effect of the level

How should the level affect the value of the overall match? It seems useful to make the level number the same as the number of integers in the CA. In the example above there are 6 levels. However, it is not of uniform depth. Whilst it should be the case that matches at lower levels are scored more highly since they show a more precise agreement in the topic matter does this mean that a perfect match at the bottom of one set of branches (e.g. 1.1.2.2.2) should score less highly than a perfect match at the bottom of another, say 1.3.2.2.1.1? The DMOZ database is a human classification of human knowledge. To some extent the classifications are partly arbitrary as they are a human philosophical construct. Whilst they accord with what many people would expect, they do not have to be exactly like that. They are the result of pragmatic as well as epistemological considerations. Therefore it seems incorrect to only allow a perfect match score when it occurs at the lowest level. In the example above it might be advisable to allow perfect matches to contribute fully at levels 4,5 and 6. Remembering that a match at the top level should count as zero then a formula to give the level effect factor would be $\frac{L-1}{4-1}$ ie level 4 would have a value of 1, level 3 $\frac{2}{3}$, level 2 $\frac{1}{3}$, whilst the top level would have a value of zero. However, this would not work for values of L greater than 4. One way to solve this would be to use the minimum of 1 or the value of this expression.

In the general case the formula becomes $\min\left\{1, \frac{L-1}{ML-1}\right\}$ where ML stands for the level at which the maximum contribution starts. In the case of DMOZ one could

use level 9 for example. An examination of its structure would be necessary before a final decision.

An alternative to this would be to use a measure that is monotonically increasing from level 1 to 16 (in DMOZ's case) but that get close to 1 fairly quickly, say at level 8, and then approaches more slowly such as in the graph below. A function that would achieve this could be $-\frac{l_{ij}^2 - 33l_{ij} + 32}{240}$ where l_{ij} is the matching level between CA_i and CA_j . This seems a better compromise since it still agrees with the principle that matches at lower levels should score higher but does not unduly penalise the branches that do not go down to the full 16 levels.

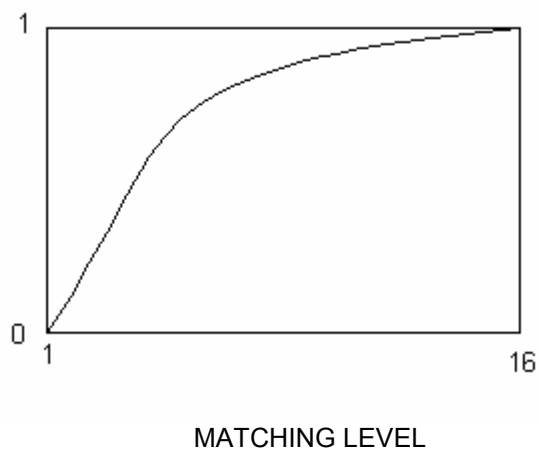


Figure 12 A graph illustrating the matching level function

4.6 The initial version of the similarity measure

Assuming we sum the contributions of all the potential matches the total would have to be divided by the total number of matches to transform the metric to a 0 - 1 scale. Therefore we arrive at the similarity measure shown below.

$$s = \frac{\sum_{i=1}^n \sum_{j=1}^m \left(\frac{1}{ed_{i,j} + 1} \times \left(\frac{l_{i,j}^2 - 33l_{i,j} + 32}{240} \right) \right)}{n \times m}$$

where

web profile1 contain CA_i ($i = 1 \dots n$) category addresses

web profile2 contain CA_j ($j = 1 \dots m$) category addresses

$ed_{i,j}$ be the edge distance from CA_i to CA_j

$l_{i,j}$ be the matching level for CA_i and CA_j

This measure will produce a value on a 0 – 1 scale with answers closer to 1

indicating a closer match.

CHAPTER 5 THE PROCESS OF THE INVESTIGATION

5.1 Processing the data

Once the data had been collected the processing could begin and it was also possible to get a feel for the typical nature of the web profiles. It immediately became apparent that there was a very large disparity between the number of bookmarks that individuals had in their profile. This had implications for the similarity measure in that I had to consider whether a match produced by two people with only 5 items in their web profile should be lower than one between two people who had 50 items in their web profiles, and also how to deal with two people who had very different numbers in their profiles. This will be considered in more detail in section 5.2.

The processing of web profiles involved mapping each of the URLs to a category in DMOZ where it was possible to do so. Some were exact matches to URLs contained in DMOZ, but if they were not then two criteria were used to judge whether they should be included. The first of these, 'initial intersection', examined the first six characters of the two web site addresses. If they matched then the next test was used whereas if they did not then that URL was rejected (ie it could not be mapped to DMOZ). The second criteria, 'percentage overlap', checked each character in one URL against the corresponding character in the second. The percentage match was then calculated and if it was 80% or over then the

URL was mapped to the corresponding category in DMOZ and the percentage was recorded. Again, if the match was less than 80% then the URL was rejected.

However, often several URLs in a web profile were matched to the same category in DMOZ, and occasionally a very large number were. This meant that a 'vote' for that category was then produced by summing all the percentage overlaps for URLs which mapped to a particular category. This would typically produce a file such as the one shown below. Note that this file is for category identification numbers not category addresses.

Category ID	Vote	Category ID	Vote
5012	200	5426	100
5064	100	5428	100
5152	100	5464	81
5294	81	5468	200
5307	100	5472	100
5310	600	5473	100
5365	100	5479	100
5374	100	5480	100
5407	200	5493	300
5417	179	5495	300
5421	200	5497	300
5422	200	5499	300

Figure 13 Part of a file of processed URLs

As can be seen from figure 13 many of the 'votes' are either 100, 200 or 300. However, there are some which are not divisible by 100 showing that the percentage overlap criteria is useful in allowing some non exact matches. The highest 'vote' in this particular file is 600 showing six exact matches at that category but there are cases where 'votes' can be as high as 19900 or more, showing 199 perfect website matches in a particular category (or even more slightly imperfect matches!). The bar chart in figure 14 shows a picture of one file

(although Excel hasn't drawn it very correctly). As can be seen the vast majority of votes are around 100 with fewer at 200 and only a very few above 500.

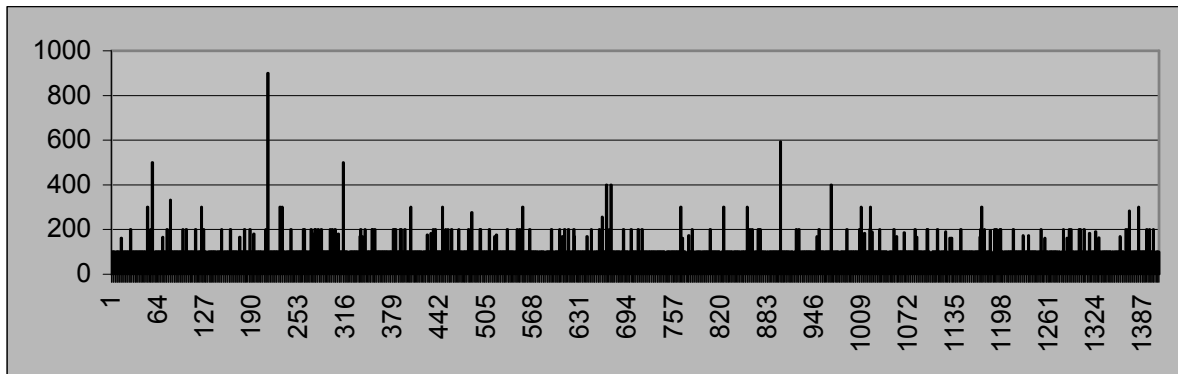


Figure 14 Bar chart showing a typical web profile after transformation

One concern about the amount of processing of the data is that what you have at the end of the time bears so little relation to what you started with that any results will be meaningless. In this case two effects were noted. In the first stage of the process where the data was cleaned as described in section 3.2 the files reduced to about 10% of their original size. Most of this was due to filtering out unnecessary characters and user categorization but nevertheless the concern was that the resulting files would be too small to be able to be useful within an AIS. However, the mapping process had the opposite effect and increased the size of the files by nearly a factor of 10 again. This effect was because within DMOZ a particular URL may well be filed in several categories. This is because a particular site may be relevant to a number of different disciplines. Therefore a user file with a small number of URLs might well become a file with quite a few more categories in it.

5.2 Refining the similarity measure

The discussion above leads to two potential changes to the similarity measure. The first is whether there is an effect on the validity of the match if the web profiles have very different numbers of URLs within them (which I will call the disparity correction factor), and the second is to do with the accuracy of the translation of the original URLs into category ids within DMOZ and therefore our confidence in the category addresses obtained (confidence factor).

If one web profile has only 10 items whilst the other has 100 then a match from these two people would seem to be less valid than one based on web profiles containing 50 and 60 items since in the first case the 10 entries from the first profile have been used proportionately more in calculating the match. Assuming that web profile 1 (n entries) is smaller than web profile 2 (m entries) then finding the fraction n/m would give a higher result to those pairs of profiles which have similar numbers of entries. However, it would give a perfect score to two profiles with only one identical URL in. Clearly the measure has to 'reward' web profiles that have a larger number of entries. One way to do this would be to include the sum of the number of entries. However, some profiles contain a very large number of entries, say five or six hundred, but these are likely to be clear outliers. In order to produce a measure that yields a range from 0 to 1 I am going to let profiles with more than 100 entries to be counted as though they have 100 entries. The spreadsheet below shows the calculation of such a measure under the assumptions above.

n	m	n/m	(n+m)/200	$n/m*(n+m)/200$	$a+(1-a)*n/m*(n+m)/200$
100	100	1.00	1.00	1.00	1.00
80	100	0.80	0.90	0.72	0.89
60	100	0.60	0.80	0.48	0.79
40	100	0.40	0.70	0.28	0.71
20	100	0.20	0.60	0.12	0.65
80	80	1.00	0.80	0.80	0.92
60	80	0.75	0.70	0.53	0.81
40	80	0.50	0.60	0.30	0.72
20	80	0.25	0.50	0.13	0.65
60	60	1.00	0.60	0.60	0.84
40	60	0.67	0.50	0.33	0.73
20	60	0.33	0.40	0.13	0.65
40	40	1.00	0.40	0.40	0.76
20	40	0.50	0.30	0.15	0.66
20	20	1.00	0.20	0.20	0.68
10	20	0.50	0.15	0.08	0.63
10	10	1.00	0.10	0.10	0.64
5	100	0.05	0.53	0.03	0.61
1	100	0.01	0.51	0.01	0.60
Disparity scaling factor				a	0.6

Figure 15 Table illustrating the disparity correction factor

The fifth column contains the proposed factor. However, I then became concerned that this might have too strong an effect on the matching function. The effect of multiplying two numbers on a 0 – 1 scale is to reduce the size of the answer and whilst the disparity factor should have some effect it should not be as strong as the original measure. Therefore I added a scaling parameter, a, to reduce the range of the disparity factor. The parameter determines the lowest value in the range (a,1) which the disparity factor can take.

The web profiles will take the form (id no.; CA 1, vote 1; CA 2, vote 2;).

Where CA stands for category address. As outlined in section 5.1 the vote is a

measure of to what extent the original URL is matched by the URL from the DMOZ database. This is complicated by the fact that a person may have a large number of URLs that would map onto the same category, leading to votes of more than 100, in fact up to 2600 in some cases. For this reason whilst the strength of a match between two CAs will need to reflect the votes, the overall measure will then have to be scaled down by the total votes of the two people rather than the number of entries in each web profile. Therefore the overall similarity measure would be as shown below.

5.3 The amended measure

Using the same variables as in the previous version, with 'a' being the scaling parameter for the disparity correction factor the similarity measure will become:

$$s = \frac{\sum_{i=1}^n \sum_{j=1}^m \left(\frac{1}{ed_{i,j} + 1} \times \left(\frac{l_{i,j}^2 - 33l_{i,j} + 32}{240} \right) \times (\text{vote } i + \text{vote } j) \right)}{\left(\sum_{i=1}^n \text{vote } i + \sum_{j=1}^m \text{vote } j \right)} \times \left(a + (1-a) \frac{n(n+m)}{200m} \right)$$

5.4 Change of direction

Despite working my way through a book on Java and quite a bit of help from Steve Cayzer in trying to understand his code, my progress was too slow to enable me to amend what he had written and produce a working version of an AIS in time to complete the dissertation. It was probably too ambitious to try to learn a

completely new computer language and amend such a large amount of code in the time that I had. This is disappointing since it is not now possible to test if the work done so far on transforming the data would enable a successful artificial immune system for recommendation.

Having decided to stop attempting to amend the code I decided at least I should try to test the similarity measure in some other way. I decided to use spreadsheet simulation to try to test the measure.

5.5 Using a spreadsheet simulation to test the similarity measure

I decided to start with a very simple tree with three branches at each of three levels as shown in figure 16 below.

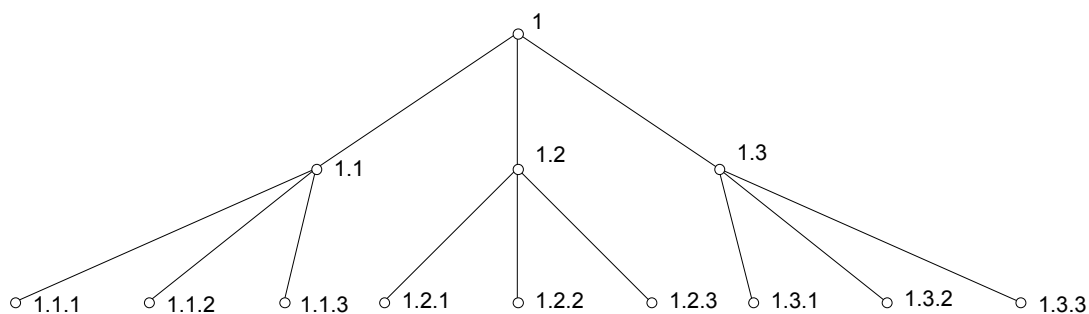


Figure 16 The simple tree structure used in the spreadsheet analysis

In the spreadsheet below I have coded the address 1.1 as 1.1.0 in order that I can use these cells in part of the function. I have not included the disparity correction factor as I am dealing with such a small number of categories. Notice that only the 4 by 4 boxes on the leading diagonal can have numbers other than 0 in them. This is because the match for the addresses in these cells would occur at the top

level and therefore should contribute zero to the overall match. For example 1.2.1 and 1.1.1.

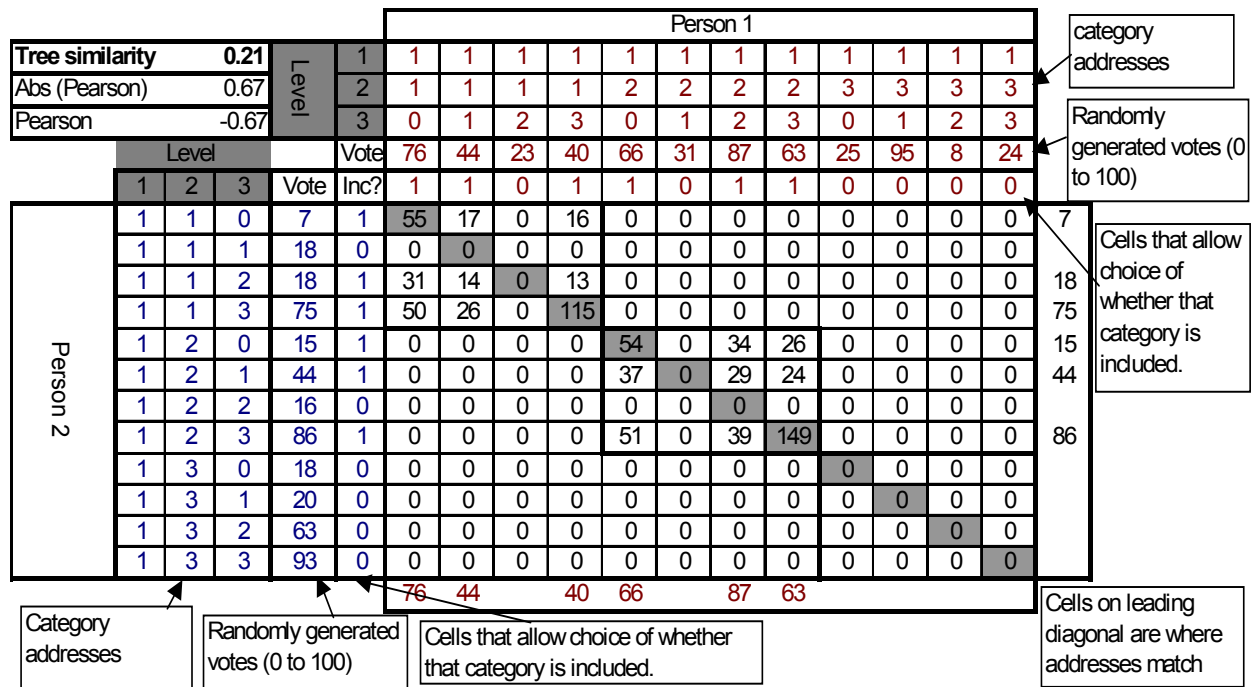


Figure 17 The spreadsheet for a 3 level 3 branch tree with random votes.

Having set up the spreadsheet I used the F9 key repeatedly (for recalculating) and one error became apparent immediately. Occasionally the tree similarity measure was greater than 1. This prompted me to look at the mathematics of the measure again and in particular to exam the effect of the votes section on the final result.

Isolating this section of the equation leads to:

$$s = \frac{\sum_{i=1}^n \sum_{j=1}^m (\text{vote } i + \text{vote } j)}{\left(\sum_{i=1}^n \text{vote } i + \sum_{j=1}^m \text{vote } j \right)}$$

I realised that the top of the equation would calculate m times the sum of vote i plus n times the sum of vote j thus allowing the quotient to be greater than 1.

Thus the amended equation should read:

$$s = \frac{\sum_{i=1}^n \sum_{j=1}^m \left(\frac{1}{ed_{i,j} + 1} \times \left(\frac{l_{i,j}^2 - 33l_{i,j} + 32}{240} \right) \times (\text{vote } i + \text{vote } j) \right)}{\left(m \times \sum_{i=1}^n \text{vote } i + n \times \sum_{j=1}^m \text{vote } j \right)}$$

The spreadsheet shown above actually uses the amended measure.

Pressing F9 repeatedly allows an initial impression of the results that are likely to occur with randomly generated votes. This can be tried with various different category addresses included and leads to some initial conclusions but also generates a number of further questions.

The first thing I noticed that the Pearson measure was much more volatile than the tree measure for any particular set of category addresses. This is shown in the set of results below which were produced by writing a macro in Visual Basic that would put the results for 100 trials in successive rows of the worksheet (only a few results shown).

Tree Sim	0.15	0.14	0.14	0.14	0.15	0.15	0.15	0.13	0.16	0.14	0.14	0.14	0.15
Pearson	0.68	0.49	0.78	0.34	0.00	0.51	0.56	-0.17	-0.49	0.02	0.32	0.43	-0.72

Figure 18 An initial comparison of the tree similarity measure and the Pearson coefficient

To some extent this is unsurprising since the Pearson measure is completely dependent on the votes provided at least three category addresses are identical. Obviously for only two identical addresses the Pearson result would be one and

for less than two the result can't be calculated. This last fact illustrates one of the key advantages of the Tree measure and that is that it can be calculated even if there are no exact matches of category address. The results in figure 18 are calculated on the basis of only 3 exact category matches but with some other category addresses that the Tree measure would use.

The non volatility of the Tree measure reflects that fact that it includes a measure of how similar the two people's 'pictorial structure' is as well as reflecting the votes. By pictorial structure I mean the picture you would get if you coloured in all the category addresses that they had in their profile on a large diagram of the tree structure. However, although it is important that the pictorial structure should influence the measure there is a question of balance ie whether the votes or the structure should have the larger effect on the overall measure.

This question of balance led to a further question and that is to do with the relative effects of the edge distance and the level of match on the overall result. There are now three elements to consider in trying to decide what balance there should be; edge distance component, level of match component and vote component.

5.6 Adjusting the balance of the measure components

If we think back to the original problem, it is a matter of deciding the 'distance' between two category addresses. Which of the above components should be most significant? A significant fact here is that the level component and the edge

component compound each other. For example, if you consider the tree diagram in figure 19, and the category addresses 1.1.1.1 and 1.2.1.1 then they will have an edge distance of 6 (the maximum for this tree) and a matching level of 1. In this case, they will have a tree measure of 0 as they meet at the top level, but even if you did not include the level component, the edge distance component would ensure that these two addresses would contribute only one seventh of the vote sum to the overall measure.

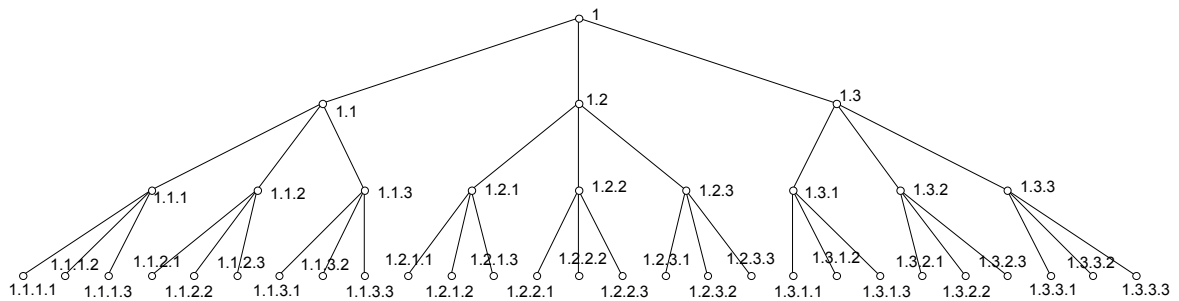


Figure 19 A 4 level, 3 branch tree structure.

A more significant example would be the matching between 1.1.1.1 and 1.1.3.3.

These have an edge distance of 4 (ie reducing the vote sum to $\frac{1}{5}$ of its value) and a matching level of 2 (ie reducing the vote sum to $\frac{1}{2}$ of its value). The two components together multiply the effect.

The compounding effect described above is illustrated in the 4 level version of the spreadsheet in figure 17. If we use uniform votes then we can see the effect of the other two components on their own. The table in figure 20 shows the effect on the tree measure of adding more and more category addresses on the overall result. All the votes were set at 1 and so the effects are all due to the other components.

User A category addresses	User B category addresses	Tree Similarity measure	Comment
1.1.1.1	1.1.1.1	1.00	Lowest level = perfect match
1.1.1	1.1.1	0.83	Level 3
1.1	1.1	0.50	Level 2
1.1.1.1 1.2.1.1	1.1.1.1 1.2.1.1	0.50	2 identical, level 4, spread out
1.1.1.1 1.1.1.2	1.1.1.1 1.1.1.2	0.64	2 identical, level 4, same branch
1.1.1.1 1.1.1.2 1.1.1.3	1.1.1.1 1.1.1.2 1.1.1.3	0.52	3 identical, level 4, same branch
1.1.1.1 1.1.2.1 1.1.3.1	1.1.1.1 1.1.2.1 1.1.3.1	0.40	3 Identical, level 4, spread out
1.1.1.1 1.1.2.1 1.1.3.1	1.1.1.2 1.1.2.2 1.1.3.2	0.16	3 non identical, level 4, spread out
1.1 1.1.1.1 1.1.1.3 1.1.2.1 1.1.2.3 1.1.3.1 1.1.3.3	1.1 1.1.1.1 1.1.1.3 1.1.2.1 1.1.2.3 1.1.3.1 1.1.3.3	0.26	7 Identical addresses
1.1 1.1.1.1 1.1.1.3 1.1.2.1 1.1.2.3 1.1.3.1 1.1.3.3	1.1.1 1.1.1.2 1.1.2. 1.1.2.2 1.1.3 1.1.3.2	0.19	7 non identical addresses
All of 1.1.x.x	All of 1.1.x.x	0.24	13 identical, all levels
All of 1.1.x.x	1.1.1.1	0.23	1 against 13
20 addresses	20 addresses	0.12	20 Identical
20 addresses	19 addresses	0.11	None identical
39 addresses	39 addresses	0.12	Entire tree
39 addresses	1.1.1.1	0.08	1 against 39

Figure 20 Selected results for a 4 level, 3 branch tree

My first thought when I started to see such low results was that I had made a mistake but the top line of the table shows that it is possible to get a value of 1. However, if you add more identical pairs of addresses then I think that the result should stay at 1, not reduce as it does. Apart from that the effects seem to be in the right order, ie non identical addresses score less than identical, spread out addresses score less than grouped ones etc.

The reason that the addition of more identical addresses leads to a reduced value of the tree measure is that the denominator of the vote section increases faster

than the overall effect on the numerator, as the numerator, despite being a sum of more values, is always reduced by the fractional effects of the edge and level components.

If we consider the effect of moving from figure 21 to figure 22, then we can see that we have added the bottom row and right hand column contributing $5\frac{1}{9}$ to the total. However, the denominator of the measure will have been increased from $3^2 + 3^2 = 18$ to $4^2 + 4^2 = 32$, an increase of 14. It is clear that the overall measure has to be redesigned.

					1	1	1	1	1
					2	1	1	1	1
					3	0	1	2	3
					Vote	1	1	1	1
Tree 0.49					Inc?	1	1	1	0
1	2	3	Vote	Inc?	1	1	1	0	
1	1	0	1	1	1.33	0.67	0.67	0	
1	1	1	1	1	0.67	2	0.44	0	
1	1	2	1	1	0.67	0.44	2	0	
1	1	3	1	0	0	0	0	0	

Figure 22 3 identical addresses

					1	1	1	1	1
					2	1	1	1	1
					3	0	1	2	3
					Vote	1	1	1	1
Tree 0.44					Inc?	1	1	1	1
1	2	3	Vote	Inc?	1	1	1	1	
1	1	0	1	1	1.33	0.67	0.67	0.67	
1	1	1	1	1	0.67	2	0.44	0.44	
1	1	2	1	1	0.67	0.44	2	0.44	
1	1	3	1	1	0.67	0.44	0.44	2	

Figure 23 4 identical addresses

It seems to me that the edge distance is more significant in judging the closeness of two categories in the DMOZ database than the level of the match. The level becomes significant when considering two identical edge distances but at different places in the tree. Perhaps we could consider removing the level component, or at least reducing its effect.

If we consider a 3 level tree, then the greatest possible edge distance that **doesn't** pass through the root of the tree would be 2. We could have a function

that gives a value of 0 for edge distances of greater than 4. For a 4 level tree the greatest possible edge distance is 4 whilst for a 5 level tree it is 6, leading to an maximum edge distance of $2(n-2)$ for an n level tree.

For 3 and 4 level trees we would require the following tables (figure 24)

3 Level Tree	
Edge Distance	Value
0	1
1	
2	
3	0
4	0

4 Level Tree	
Edge Distance	Value
0	1
1	
2	
3	
4	
5	0
6	0

Figure 24 Tables showing required edge distance function values

A function that would achieve this neatly would be $(3 - ED)/3$ and $(5 - ED)/5$ for the 3 and 4 level trees respectively. However, there would be a problem with the edge distances of 4 and 6 for the 3 and 4 level trees as these give values of minus one third and minus one fifth respectively. However, this could be avoided by either specifying a maximum possible edge distance when this was being calculated, or setting any negative values given by the function to zero. For an n level tree the new edge distance function becomes:

$$\frac{((2n - 3) - ED)}{(2n - 3)}$$

The table in figure 24 would have the values shown in figure 25.

3 Level Tree	
Edge Distance	Value
0	1
1	2/3
2	1/3
3	0
4	0

4 Level Tree	
Edge Distance	Value
0	1
1	4/5
2	3/5
3	2/5
4	1/5
5	0
6	0

Figure 25 The completed tables

We then have to consider whether to have any level component at all. The results in the tables in figure 25 are quite different to the previous edge distance component in that they reduce in a linear fashion whereas the previous function gave results of 1, $\frac{1}{2}$, $\frac{1}{3}$, $\frac{1}{4}$ etc, ie it reduced more at the top of the table and less as the edge distance increased. The DMOZ structure has 16 levels and an edge distance of 2 could imply a matching level of 15 or 2. Clearly we need some sort of 'tie breaker' under these circumstances. However, amending the edge distance component alone whilst slightly increasing the overall value of the tree measure will not prevent the reduction effect on adding more identical category addresses. And omitting the level component would allow matches at the top level to contribute to the overall measure which does not meet the original principles for the construction of the measure.

The reduction effect is due to the measure using the contributions of every category address against every other one. Even though we might have two pairs of identical addresses from the two users, the tree measure will also include the

effect of all the combinations of addresses, not just identical address against identical address. This inevitably leads to a reduction in the value of the measure.

In fact this leads me to question whether the measure is fatally flawed in its current form. Whilst it has the property that it will give a value in circumstances where the Pearson measure would not, this very property may be the one that produces a reduced value as more perfect matches are added. Philosophically this goes against what we require of the measure. Surely, two people with 20 category addresses in common should have a higher value from the measure than two people with only 5 category addresses in common. A more radical rethink is needed.

5.7 A radical redesign

The problem outlined above seems to arise from one of the original principles, namely that the measure should be the sum of the contributions of the matches of all the elements on one web profile with all the elements of the other. An alternative is to take the first element of one web profile and find its **best** match with the elements of the other profile. Then take the next element and do the same thing. Sum all of these and divide by the total of the votes of the two users. This will not produce a symmetrical measure (unless the two users only have exactly identical sets of category addresses) and the implications of this would have to be considered carefully. In the context of an AIS then this might be beneficial. For example an antibody would have its concentration increased proportionally to how well it matched another antibody but have its concentration

reduced proportionally to how well the other antibody matched it. Presumably the antibody's match to the antigen would determine any increase in concentration rather than the other way around.

This would mean that for an p level tree structure there would be two measures:

For user 1 with n category addresses:

$$s_1 = \frac{\sum_{i=1}^n \text{MAX}_{j=1}^m \left[\left(\frac{(2p-3) - ED_{i,j}}{(2p-3)} \right) \times \left(\frac{l_{i,j}^2 - (2p+1)l_{i,j} + 2p}{p(p-1)} \right) (\text{vote}_i + \text{vote}_j) \right]}{\left(\sum_{i=1}^n \text{vote}_i + \sum_{j=1}^m \text{vote}_j \right)}$$

And for user 2 with m category addresses

$$s_2 = \frac{\sum_{j=1}^m \text{MAX}_{i=1}^n \left[\left(\frac{(2p-3) - ED_{i,j}}{(2p-3)} \right) \times \left(\frac{l_{i,j}^2 - (2p+1)l_{i,j} + 2p}{p(p-1)} \right) (\text{vote}_i + \text{vote}_j) \right]}{\left(\sum_{i=1}^n \text{vote}_i + \sum_{j=1}^m \text{vote}_j \right)}$$

Testing this with a 3 level, 3 branch, chosen vote spreadsheet as in figure 26 below, I found that adding identical category addresses at the lowest level kept the similarity measure at 1, as one would expect. However, I soon found that it was possible to go over 1 on the similarity measure, under certain conditions. If one user had only one category address, 1.1, with a very large vote (1,000,000) whilst the other had category addresses at 1.1, 1.1.1, 1.1.2, and 1.1.3 with votes

of say 1 then it is possible to get the similarity measure to reach 2. In order to avoid this and find the correct term for the denominator I tried to find a mathematical way of working out the expansion of the above equations but could not deal with the expansion of the MAX part of the equation. It is possible to replace the denominator with $m \times \text{sum of vote } i + m \times \text{sum of vote } j$ as before but we then immediately move back into the situation where the addition of more exact category matches reduces the value of the tree measure.

														User 2														
														1	1	1	1	1	1	1	1	1	1	1	1	1	1	
														2	1	1	1	1	2	2	2	2	3	3	3	3	3	
														3	0	1	2	3	0	1	2	3	0	1	2	3	3	
User 1				Vote	100	100	100	100	100	100	100	100	100	100		Row max												
1	2	3	Vote	Inc?	0	1	0	0	0	0	0	0	0	0														
1	1	0	100	0	0	0	0	0							0													
1	1	1	100	1	0	200	0	0						100	200													
1	1	2	100	0	0	0	0	0							0													
1	1	3	100	0	0	0	0	0							0													
1	2	0	100	0					0	0	0	0					0											
1	2	1	100	0					0	0	0	0					0											
1	2	2	100	0					0	0	0	0					0											
1	2	3	100	0					0	0	0	0					0											
1	3	0	100	0						0	0	0	0					0										
1	3	1	100	0						0	0	0	0					0										
1	3	2	100	0						0	0	0	0					0										
1	3	3	100	0						0	0	0	0					0										
					100										200	200	1											
Column max					0	200	0	0	0	0	0	0	0	0	200													
														Similarity measure for user 2				1	Total vote									

Figure 26 Revised model spreadsheet

The difficulty comes when trying to incorporate the vote information along with the edge distance and level components. If we just use these last two then the addition of more categories tends to increase the matching measure as one would expect. Possibly the way round this is to limit the effect that the vote component

can have on the overall measure. This could be achieved by calculating the vote section separately and then only allowing it to reduce the contributions of each match to the overall measure by, say, 10%.

We can then move back to using a symmetric similarity measure as shown on the following page.

$$s = \frac{\sum_{j=1}^m \text{MAX}_{i=1}^n \left[\left(\frac{(2p-3) - ED_{i,j}}{(2p-3)} \right) \times \left(\frac{l_{i,j}^2 - (2p+1)l_{i,j} + 2p}{p(p-1)} \right) \times Z \right] + \sum_{i=1}^n \text{MAX}_{j=1}^m \left[\left(\frac{(2p-3) - ED_{i,j}}{(2p-3)} \right) \times \left(\frac{l_{i,j}^2 - (2p+1)l_{i,j} + 2p}{p(p-1)} \right) \times Z \right]}{n+m}$$

Where Z is as follows:

$$Z = \left[\left(\frac{\text{vote } i + \text{vote } j}{m \times \sum_{i=1}^n \text{vote } i + n \times \sum_{j=1}^m \text{vote } j} \right) \times (1 - k) + k \right]$$

And k is the lower bound of the percentage that the matching value can be reduced by. So, for example, K might be set to 0.9 thus limiting the reduction effect of the votes to 10%.

Investigating the new equation with the aid of a 3 level 3 branch spreadsheet with k set to 0.9 leads to the following conclusions.

1. The addition of extra perfect category matches does lead to a reduction in the overall measure but one that is very small after the first reduction (from 1 perfect match $s = 1$, to 2, $s = 0.925$ to 3 perfect matches, $s = 0.911$)
2. An increase in the vote for categories that match perfectly increases the value of s
3. An increase in the vote for categories that do not match, but are in the same section of the tree, leads to an increase in s , whilst an increase in the vote for categories that are in different sections of the tree leads to a decrease in s .
4. An increase in the number of categories included can lead to a reduction in s if the added categories are in separate sections of the tree. However, addition of further categories in sections of the tree that are already populated lead to increases in s .

5.8 Summary

The last version of the measure, although a compromise, does seem to have the properties that would be required of a tree similarity measure. It is a drawback that there is a reduction, albeit small, in the overall value of the measure as you add more and more perfect matches. However, in a larger scale tree this is extremely unlikely to happen and so the fact that the measure will produce a result that increases when more elements exist in a particular section of the tree, and the

fact that it will cope with no perfect matches in category addresses would seem to outweigh the drawback noted above.

CHAPTER 6 CONCLUSION

6.1 The results in context

Billsus and Pazzani (1998) commented that there were few collaborative filtering techniques available and that there are severe limitations with the use of the Pearson correlation coefficient. Chapters 4 and 5 detail my attempt to address the second of these comments in the context of the web site recommendation problem. Without applying the proposed measure within an AIS it is not possible to test whether it is useful nor to compare its performance with the Pearson measure. However, the spreadsheet simulations have allowed the modelling of the measure and has exposed several flaws along the way. The final measure produced does fulfil the principles outlined in section 4.2. It has been tested using specific scenarios and by using randomly generated data. It seems to robustly reflect the closeness of the 'pictorial structure' of two users. The effect of the votes on the measure is purposefully limited but could act as a tiebreaker for two near identical structures.

6.2 Further work

The final measure has evolved from its initial conception and therefore there has not been enough time to fully explore its properties. There have been decisions made about the construction of the measure, for example the quadratic function for the matching level factor, and about the constant parameters within it, for example the limiting of the vote effect to 10% of its original value, which will affect

the performance of the measure. These will need to be tested carefully. In particular it would be useful to know how it reacts in comparison with the Pearson coefficient when used in an AIS. The necessary transformations to the data to allow it to be used within an AIS may be such that the performance of the AIS is compromised and a thorough assessment of this factor needs to be undertaken in line with the comparison of the two measures proposed above. Finally, the last version of the measure is the sum of two, non symmetric, part measures. There was not time to fully evaluate the way that these part measures interact in different circumstances to produce the overall result. It is possible that a non symmetric measure would be more appropriate in an AIS. I do not know if there is any more inspiration that can be taken from Biology to guide us in this question.

6.3 Conclusion

It has not been possible to achieve one of the initial aims of this research, to build an artificial immune system in order to recommend new web sites to a user. This is disappointing and it is to be hoped that this work can be carried on in some way in the future. However, it has been possible to do quite extensive work on the construction of a new similarity measure that would exploit tree structures. The final test of such a measure would have to be in an artificial immune system but it has been possible to build a number of spreadsheet models to test the properties of the proposed measures. This has enabled the evolution of the measure to better meet the original principles, and the examination of the original principles themselves.

The eventual measure is not perfect. It may be that it is not possible to achieve all the aims that we desire in one measure, or it may be that a different formulation could do this. It has also become quite complicated. There is virtue and elegance in simplicity in that the performance of the measure is more easily predicted and it is easier to understand. The final measure, whilst fairly complicated compared with the original version, is built to cope with more factors, and may well be the simplest formulation that can achieve this.

Bibliography

Amazon.com	http://www.amazon.com
Billsus, D. and Pazzani, M. (1998).	"Learning Collaborative Information Filters" In Shavlik, J., ed., <i>Machine Learning: Proceedings of the Fifteenth International Conference</i> , Morgan Kaufmann Publishers, San Francisco, CA.
Burnet, F. M. (1959)	<i>The Clonal Selection Theory of Acquired Immunity</i> . Cambridge University Press, Cambridge.
de Castro, L. N. & Von Zuben, F.J. (1999).	Artificial Immune Systems. <i>Technical Report</i> Santa Fee University
Cayzer, S. and Aickelin, U. (2001).	A recommender system based on the immune network. Proceedings of CEC 2002
Compaq Systems Research Centre. EachMovie collaborative filtering data set	http://www.research.compaq.com/SRC/eachmovie/ .
DMOZ ontology	http://dmoz.org/ .
EXCAVATOR	http://compbio.ornl.gov/structure/excavator/command.html
Farmer J.D., Packard N.H. and Perelson A.S., (1986)	The immune system, adaptation, and machine learning <i>Physica D</i> , vol. 22, pp. 187-204, 1986.
Gokhale A., (1999)	Improvements to Collaborative Filtering Algorithms 1999. Worcester Polytechnic Institute. http://www.cs.wpi.edu/~claypool/ms/cf-improve/ .
Hajela, P. and Yoo, J. (1999)	Immune Network Modelling in Design Optimization, <i>New Methods in Optimisation</i> , D. Corne, M. Dorigo and F. Glover (Eds), McGraw-Hill, pp. 203-216,.
Hofmeyr, S.A. and Forrest, S. (2000).	Architecture for an Artificial Immune System. <i>Evolutionary Computation</i> 7, pp 45-68
Jerne N.K. (1973)	Towards a network theory of the immune system <i>Annals of Immunology</i> , vol. 125, no. C, pp. 373-389.

Jerne, N.K. (1973).	The immune system. <i>Scientific American</i> . 229 pp 52-60.
Kim, J and Bentley, P.J. (2001a).	An evaluation of Negative Selection in an Artificial Immune System for Network Intrusion Detection. <i>Genetic and Evolutionary Computation Conference</i> . pp1330-1337
Kim, J and Bentley, P.J. (2001b).	Towards an Artificial Immune System for Network Intrusion Detection : An investigation of clonal selection with a negative selection operator. <i>Special issue on AIS from I.E.E.E evolutionary computing</i> .
Noel, S. and Chu, C.H. (2002)	Visualisation of Document Co-Citation Counts 6 th <i>International Conference on Information Visualisation</i> London (email snoel@gmu.edu)
Perelson A.S., and Weisbuch G. (1997)	Immunology for physicists <i>Reviews of Modern Physics</i> , vol. 69, pp. 1219-1267
Timmis, J., Neal, M. and Hunt, J. (2000)	An Artificial Immune System for Data Analysis. <i>Biosystems</i> 55 pp143-150
Yibg, X., Olman, V, and Xu, D.	Clustering Gene Expression Data using a Graph-Theoretic Approach: An application of minimum spanning trees (In press) (see EXCAVATOR URL above)

Appendix A

DONATE YOUR BOOKMARKS TO SCIENCE AND WIN CASH

For full details go to <http://www.bookmark.ac>

Artificial Immune Systems (AIS) are a new paradigm modelled after the 'real' thing. We believe that they could be developed into an extremely powerful tool to extract information from a database. In order to confirm this conjecture, we decided to experiment with the task of extracting useful information from a database of Internet addresses. Will an AIS be able to recommend sites of interest to the user? Can it match 'antigen' users with an 'antibody' database?

To do this research we ask for your help. We would like you to email us the set of website addresses (URLs) contained in your favourites or bookmark file. Please be assured that we will keep NO PERSONAL DETAILS once we have received the email. We will simply copy the bookmarks into a database and delete the email so we will not be able to contact you in relation to your bookmarks, nor will we have any way of matching up the list of bookmarks with you.

Please send your files to the following address, which you can also use to ask us any questions you might have or if you are simply curious about our research:

Research@bookmark.ac

We estimate that we need about 1000 people's lists to make the research viable and so we would really appreciate it if you can help us. So far we have collected 520. If you would like details of the results of the research once it is finished then please indicate. If you agree, we will keep all these email addresses separately to enter them into a draw for two 15 Pound tokens and so that we can send the research details.

Please pass on our request for help to other people and mailing lists. More information and detailed instructions how to extract and submit the bookmark / favourite file from most browsers can be found at <http://www.bookmark.ac>