FUZZY ARTIFICIAL IMMUNE SYSTEM APPROACH TO RECOMMENDER SYSTEMS

Dissertation submitted to
Jawaharlal Nehru University
In partial fulfillment of the requirement
For the award of the Degree of

MASTER OF TECHNOLOGY

In

COMPUTER SCIENCE AND TECHNOLOGY

By

GAGANJOT KAUR AWAL

Under the Supervision of
Prof. K.K.Bharadwaj

SCHOOL OF COMPUTER AND SYSTEMS SCIENCES
JAWAHARLAL NEHRU UNIVERSITY
NEW DELHI -110067

JULY 2006

जवाहरलाल नॅहरू विश्वविद्यालय

JAWAHARLAL NEHRU UNIVERSITY

School of Computer & Systems Sciences
NEW DELHI- 110067, INDIA

CERTIFICATE

This is to certify that the dissertation entitled "**FUZZY ARTIFICIAL IMMUNE SYSTEM APPROACH TO RECOMMENDER SYSTEMS**", being submitted by Ms. Gaganjot Kaur Awal to the **School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi** in partial fulfillment of the requirement for the award of the degree of **Master of Technology** in **Computer Science and Technology**, is a record of original work done by her under the supervision of Prof. K.K.Bharadwaj. This work has not been submitted in part or full to any other University or Institution for the award of any degree or diploma.

Gaganj

Gaganjot Kaur Awal
(Student)

28-07-06

Prof K.K.Bharadwaj
(Supervisor)

Prof. S. Balasundaram
(Dean, SC & SS, J.N.U., New Delhi-67)

Dedicated to

My Parents

Acknowledgements

First and foremost, I would like to thank Lord Almighty at the completion of my dissertation. It is His grace and benevolence that I have been blessed with a devoted and sincere supervisor and extremely supportive parents.

I wish to express my sincere gratitude to my supervisor Prof. K.K.Bharadwaj. I thank him for his constant support, guidance and patience during all stages of my research work including the writing of this dissertation. He gave me the freedom to choose the topic of my interest and provided me with relevant material for the same, as and when needed. His constant guidance helped me to explore alternatives without losing perspective. Working under his supervision has been a great learning experience. The best method to teach something is by giving an example; more than what he has taught, I have learnt from what he is – a sincere, disciplined and helpful guide.

I would like to thank all my teachers at SC&SS for their motivation and blessings. I would also like to thank all those at J.N.U. with whom I have been associated, for making this period a memorable experience.

I am grateful to my parents for their unlimited support and encouragement without which this work would not have been possible. I would also like to thank my brother and sister for their feedback and help in completion of this dissertation.

I acknowledge and thank each one of those who, directly or indirectly, helped me in this work.

Gaganjot Kaur Awal

Contents

List of Figures

List of Acronyms

AI	Artificial Intelligence
AINE	Artificial Immune NEtwork
AIS	Artificial Immune Systems
AMD	Athlon Micro Devices
ANN	Artificial Neural Networks
AOL	America Online
APC	Antigen Presenting Cells
API	Application Programmers' Interface
ARB	Artificial Recognition Ball
B2C	Business to Consumer
CBF	Content-Based Filtering
CF	Collaborative Filtering
DNA	Deoxy-ribo Nucleic Acid
EA	Evolutionary Algorithms
FAIR	Fuzzy AIS based Recommender System (Algorithm)
FAIS	Fuzzy AIS
GA	Genetic Algorithm
HP	Hewlett Packard
MAE	Mean Absolute Error
ML	Machine Learning
NAT	Network Affinity Threshold
NLP	Natural Language Processing

List of Acronyms (contd.)

RLAIS	Resource Limited Artificial Immune System
RNA	Ribo-Nucleic Acid
SC	Soft Computing
top-N	most highly stimulated
URL	Uniform Resource Locator
WCM	Web Content Mining
WSM	Web Structure Mining
WUM	Web Usage Mining

Abstract

The advent of the Internet and the information revolution has put serious challenges before computer scientists as computational problems have now become more complex. Hence, novel approaches to solve these problems are being explored. Nature has been a source of inspiration for many successful approaches and paradigms, such as, Artificial Neural Networks, Genetic Algorithms, etc. Immune System (IS) is yet another such biological system that exhibits powerful information processing capabilities and therefore, has a great potential as a computational paradigm. The immune system is highly distributed, highly adaptive, self-organizing in nature, maintains a memory of past encounters and has the ability to continually learn about new encounters. Recent efforts to capitalize on these features of the IS metaphor and incorporate them into computational models have led to the emergence of a separate area of research, viz. Artificial Immune Systems (AIS). AIS have been successfully applied in many domains. Various models inspired by theories from immunology have been proposed in literature. This work is based on a fuzzy AIS model. The Web exhibits similar characteristics to the environment in which biological immune systems operate and therefore is a suitable candidate for the application of AIS as it can easily adapt to dynamic environments like the WWW. The development of e-commerce has placed an increasing importance on web personalization and recommender systems. Collaborative Filtering (CF) is one of the most popular and successful techniques used in recommender systems. In this work, a general framework and a CF algorithm (FAIR) for design of a recommender system based on Fuzzy AIS has been proposed. A prototype movie recommender system based on this design was developed for experimentation. The encouraging results that we have obtained suggest the suitability of Fuzzy AIS for collaborative filtering as incorporation of fuzzy set theory in AIS helps in keeping the size of the AIS in limits without compromising quality of recommendations.

Chapter 1

Introduction

1.1 Artificial Immune Systems

Biological systems have always been a source of inspiration to computer scientists for solving complex computational problems. This is because living organisms have sophisticated learning and processing capabilities that allow them to survive and proliferate generation after generation in their dynamic and competitive environments. Artificial Neural Networks, genetic algorithms, ant colony optimization are examples of biological metaphors successfully applied to solve real world computational problems. Artificial Immune systems (AIS) is yet another such system that takes its inspiration from nature.

AIS can be defined as computational system inspired by theoretical immunology, observed immune functions, principles and mechanisms in order to solve problems. (The system of animal body, which protects it from various infectious agents and cancer, is known as *immune system* and the study of the immune system is known as *immunology.*)

Artificial immune systems have been successfully applied in various domains like intrusion detection, fault diagnosis and tolerance, pattern recognition, data mining tasks – supervised and unsupervised learning, job-shop scheduling, web mining, user profiling and recommendation systems, robot navigation, multimodal optimization, etc.

This field is still in its infancy and many issues need to be resolved. Nonetheless, many experiments performed in the above-mentioned application areas using AIS have shown results that are comparable or superior to other machine-learning techniques indicating the promising nature of AIS as a computational paradigm.

Before discussing what Artificial Immune Systems are, here is an overview of the natural immune system that acts as an inspiration for AIS.

1.1.1 *Natural Immune System*

All living beings have an immune system whose complexity varies according to their characteristics. The immune system of human beings is the major source of inspiration for AIS, primarily because of its interesting features and the great knowledge available about it. It is composed of a vast array of cells, molecules, and organs that work together to maintain life. This makes it a robust, complex, adaptive system that defends the body from foreign pathogens. This section give a brief overview of the immune components and processes that have been focus of AIS practitioners.

Immunity is of two types: *innate* (non-specific) and *acquired* (adaptive/ specific). Acquired immunity, in particular, is of interest to AIS researchers because of its special cognitive properties that characterize intelligence, such as, memory and recognition, learning, adaptation, etc. The acquired or adaptive immune system is comprised mainly of lymphocytes (*B-cells* and *T-cells*) which are special types of white blood cells that detect and destroy *pathogens*, such as viruses and bacteria. The features that allow the identification of a particular pathogen are the *antigens* (which are mostly proteins) on the cell-surface of the pathogen. Special proteins receptors on the B-cell surface, called *antibodies* react to a particular antigen by *binding* to this antigen. And this binding relation is *specialized* so that only certain antibodies can bind and hence recognize a particular antigen. Even though there are around 10^{16} antigen varieties, the immune system is armed with only 10^{8} antibody types in its repertoire at any given time. Hence lymphocytes bind only *approximately* to pathogens, to allow the recognition of a larger number of antigens. Lymphocytes are only activated when the bond is strong enough, and this minimum strength may be different for different lymphocytes. A stronger binding with an antigen induces a lymphocyte to *clone* more copies of itself, hence providing reinforcement. Furthermore, to diversify their repertoire and be able to recognize more antigens, lymphocytes undergo *somatic hypermutation* (*hyper* because the mutation rate is very high as compared to the (evolutionary) mutation which is a rare event).

B-cells give rise to *plasma cells (effector B-cells)* and *memory B-cells.* Some of the activated B-cells enlarge, divide and differentiate into a clone of plasma cells. Although

plasma cells live for only a few days, they secrete enormous amounts of antibody during this period. A few days after exposure to an antigen, a plasma cell secretes hundreds of millions of antibodies daily and secretion occurs for about 4 or 5 days until the plasma cell dies. Some activated B-cells do not differentiate into plasma cells but rather remain as memory cells. They have a longer life span and remain dormant until activated once again by a new quantity of the same (similar) antigen.

Primary and Secondary Immune Responses

A primary response is invoked when the immune system encounters an antigen for the first time. A number of antibodies are produced by the immune system in response to the infection. If this antigen or a similar antigen is encountered again in the future, secondary immune response is generated which is specific to the antigen and causes a very rapid growth in the quantity of B-cells and antibodies. This second, faster response is attributed to the memory cells specific to this antigen. This is what is called 'developing immunity' in layman terms and provides the basis for vaccination and immunization.

1.1.2 *Capabilities of the Immune System*

Natural Immune System can be seen as a parallel and distributed adaptive system which possesses the following capabilities [6]:

- ***Recognition***: The immune system has the ability to recognize, identify and respond to a vast number of different (antigenic) patterns. Additionally, the immune system can differentiate between malfunctioning *self-cells* and harmful *non-self* cells, therefore maintaining some sense of self.
- ***Feature Extraction***: Through the use of Antigen Presenting Cells (APC) the immune system has the ability to extract features of the antigen by filtering molecular noise from disease causing agents called an *antigen,* before being presented to other immune cells, including the lymphocytes.
- ***Diversity***: There are two major processes involved in the generation and maintenance of diversity in the immune system. First, is the generation of receptor molecules through the recombination of gene segments from gene libraries. By

recombining genes from a finite set, the immune system is capable of generating an almost infinite number of varying types of receptors, thus endowing the immune system with a large coverage of the universe of antigens. The second process, which assists with diversity in the immune system, is known as somatic hypermutation. Immune cells reproduce themselves in response to invading antigens. During reproduction, they are subjected to a somatic mutation process with high rates that allow the creation of novel patterns of receptors molecules, thus increasing the diversity of the immune receptors.

- ***Learning***: The mechanism of somatic hypermutation followed by a strong selective pressure also allows the immune system to fine-tune its response to an invading pathogen; a process termed *affinity maturation.*. Affinity maturation guarantees that the immune system becomes increasingly better at the task of recognizing patterns. The immune network theory is another powerful example of learning in the immune system. It suggests that the immune system has a dynamic set of mutually recognizing cells and molecules, and the presence of an invading antigen causes a perturbation in this network. As a result, the dynamic immune network, which presents an intrinsic steady state in the absence of antigens, has to self-organize its pattern of behavior again, so as to accommodate the disturbance. Therefore, invading antigens require the immune network to adapt itself to this new element.
- ***Memory***: After an immune response to a given antigen, some sets of cells and molecules are endowed with increased life spans in order to provide faster and more powerful immune responses to future infections by the same or similar antigens. This process, known as the maturation of the immune response, allows the maintenance of those cells and molecules successful at recognizing antigens. This is the major principle behind vaccination procedures in medicine and immunotherapy. A weakened or dead sample of an antigen (e.g., a virus) is inoculated into an individual so as to promote an immune response (with no disease symptoms) in order to generate memory cells and molecules to that antigen. Another theory for memory is the immune network theory.

- ***Distributed detection***: There is inherent distribution within the immune system. There is no one point of overall control; each immune cell is specifically stimulated and responds to new antigens that can invade the organism in any location.
- ***Self-regulation***: Immune systems dynamics are such that the immune system population is controlled by local interactions and not by a central point of control. After a disease has been successfully combated by the immune system, it returns to its normal steady state, until it is needed in response to another antigen. The immune network theory explicitly accounts for this type of self-regulatory mechanism.
- ***Metadynamics***: The immune system is constantly creating new cells and molecules, and eliminating those that are too old or are not being of great use. Metadynamics is the name given to this continuous production, recruitment and death of immune cells and molecules
- ***Immune Network***: In 1974, N. Jerne proposed the immune network theory as an alternative to explain how the immune system works. He suggested that the immune system is a dynamic system whose cells and molecules are capable of recognizing each other, thus forming an internal network of communication within the organism. This network provides the basis for immunological memory to be achieved, via a self-supporting and self-organizing network.

1.1.3 *AIS Models and Approaches*

AIS practitioners are trying to model the components and processes of the mammalian immune systems using one or more of these immunological theories. Various models and algorithms have been and are being developed to arrive at a computational model that has one or more of the above-mentioned capabilities of the natural immune system.

A number of implementations of artificial immune systems rely on the immune network metaphor. One of the earliest applications of the network idea to a machine learning problem was given by Hunt and Cooke [11], who developed an AIS to classify sequences of DNA as promoter-containing or promoter-negative. This work attempted to closely adhere to the biological model, for example it modeled B-Cells containing gene libraries

and messenger RNA from which antibodies could be produced via a transcription mechanism, and it utilized matching rules weighted in favor of contiguous matching regions. B-Cells were stimulated according to the algorithm given by Farmer, and clones of B-Cells produced via somatic hypermutation. New clones were then integrated into the network. Whilst the work yielded some promising results, it was unable to perform as well as a previously published neural network approach to classifying the data.

The model was improved in [12] in an attempt to build an immune system capable of case based reasoning. The idea was that each B-Cell in the network would represent a case, and similar cases would be linked together via the network which was self-organizing in nature. The system contained both specific and generalized cases, attempting to mimic the way that the natural system can generalize over infections. This model still exhibited some major limitations as far as application to real-world complex data-sets. In particular, many problems were associated with building the immune network-if the network was randomly initialized, it took a long time to build useful patterns within the network, and there was an extremely high overhead associated with insertion and deletion of nodes into and from the network, especially as the size of the network grew. Furthermore, attempting to mimic the method by which matching occurs in the real immune system proved too simplistic and only applicable to binary data strings.

Building on the foundations laid by Hunt *et al*, a sequence of improvements presented in [27, 28] has led to the emergence of a system originally named *RLAIS*, Resource Limited Artificial Immune System, which was later renamed to *AINE*.

AINE introduces the concept of the *Artificial Recognition Ball*, or *ARB*. A network consists of a number of linked *ARBs*, with links representing similarity between them. Each ARB represents a data item (representing a no. of identical B cells) that could be matched by Euclidean distance to an antigen or to another ARB in the network. A link is created if the affinity between the two ARBs is below a network affinity threshold NAT. The network initially consists of a cross-section of the data to be learnt, with the remainder of the training data comprising the antigen set. The system contains a fixed

number of B-Cells; the *ARBs* compete for the ability to represent these B-Cells, according to their current stimulation level. Stimulation *sl* of an ARB is determined by three factors;

- the primary stimulation of the ARB by antigen (i.e. the data), *ps*
- the stimulation by the neighbors , *nn*
- the suppression by the neighbors, *ns*

The stimulation level is computed as:

$$sl = ps + nn - ns = \sum_{x=0}^{a}(1 - pd_x) + \sum_{x=0}^{n}(1 - dis_x) - \sum_{x=0}^{n}(dis_x) \qquad (1.1)$$

where a sis the number of antigens an ARB has been exposed to, pd_x is the distance between the ARB and the xth antigen in the normalized data space, and dis_x is the distance of the xth neighbor from the ARB.

B-Cells re allocated to ARBs, depending on their stimulation level, regardless of how many B-Cells are actually available. Then, the weakest ARBs (with minimum no. of B-cells) are systematically removed until the number of B-Cells allocated is exactly equal to the maximum available (resource limited model—here B-cells are the resource). This introduces competition between ARBs and provides a mechanism for achieving population control. Remaining ARBs are cloned and mutated according to their stimulation level, and the clones are integrated into the network if their affinity to other *ARBs* in the network is below some fixed threshold. This gives rise to a meta-dynamical system which eventually stabilizes into a network that represents the patterns within the data. The network is visualized in order to observe clusters. The system requires tuning of three parameters: the threshold governing insertion of cells into the network, the number of resources allowed, and mutation rate which controls diversity. The ARB concept has been successfully applied to the classification task in AIRS by Watkins *et al* [30]. But, this model has some limitations like premature convergence, sacrificing diversity for the sake of scalability, etc.. These limitations have been addressed by [20]. A Fuzzy

Artificial Immune Model has been proposed in [18, 19] to model the approximate nature of antigen-antibody binding. This model is discussed at length in Chapter 2.

The next section gives a brief overview of the field of Web Mining which has recently emerged as a research area with its own set of issues and challenges. Web Mining is a broad area with very real and widespread implications as it concerns all the users of the Web. The mining tasks on Web data are more complex than traditional data mining tasks and demands for more sophisticated computational techniques. Many AI, ML and Soft Computing (SC) techniques have proved to be strong foundations for building web mining applications. AIS is yet another approach whose potential as a computational paradigm is being explored in various domains, web mining being just one of them. Section 1.2 describes the field of Web mining in broader terms leading in to section 1.3 where the focus is on more specific tasks in the field of Web mining, *viz.* Web Personalization and Web recommendation.

1.2 Web Mining

World Wide Web has become an integral part of our daily lives - from connecting to friends to conducting businesses, from expressing opinions to conducting research, no sphere of life has been left untouched by the Web.

Though the Web has become an important source of information, the explosive growth of the Web has posed many problems for its users. Users of the Web are currently facing *information overload* due to the overwhelming amount of information that is available over the web. The main reason is the unorganized, uncontrolled, and non-standard nature of the Web data.

Today, the Web is the most popular publishing medium; a distinctive feature is that almost anybody can use this medium for publishing due to the absence of a central control / editorship or authority. Web data is primarily unstructured, unlabeled, heterogeneous and multilingual. Moreover, the Web is a huge and dynamic repository of such data. To be able to get relevant information from the Web, the users need assistance of intelligent software agents for finding, sorting and filtering the available information.

Web Mining refers to the research in that direction. It can be defined as: "The use of data mining techniques to automatically discover and extract information from World Wide Web documents and Services." Earlier, the research in this area was referred to as Data Mining on the Web Data. But now, Web Mining has evolved as a separate research area, primarily because of the unique nature of the Web and the special treatment required for extracting knowledge from it. 'Web Mining', now refers to the discovery and analysis of useful information on the World Wide Web. Web Mining Research draws techniques from various disciplines like Databases, Information Retrieval, Data mining, Text Mining, Machine Learning, NLP, etc.

The book on Web Mining by Chakrabarti [5] is first and original effort to present the concept of Web Mining in its entirety from elementary principles to applications in the real world. It is a useful work for beginners and comprehensive enough for those who want to start pursuing research in this field.

Web Mining Components

A data-centric decomposition of the Web Mining process was proposed by [15] and is the most-favored Web Mining Taxonomy today.

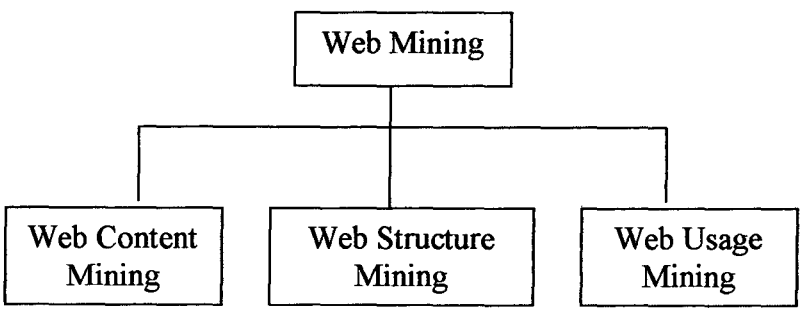


Figure 1: Web Mining Taxonomy

According to this taxonomy Web Mining Research can be categorized into three types. [3] (See Figure 1):

1) Web Content Mining
2) Web Structure Mining
3) Web Usage Mining

Web Content Mining (WCM)

WCM refers to the discovery of useful information from the Web contents/data/ documents .Web consists of several types of data such as textual, image, audio, video, metadata as well as hyperlinks. Research on mining multi-types of data is called multimedia data mining and may be considered an instance of Web Mining. WCM focuses primarily on mining knowledge from text and hypertext in Web documents.

Mining knowledge from text is called text mining. Text data in the Web may exist in unstructured form (like free text), semi-structured form (like HTML documents) or structured form (as in database generated HTML pages / table data). Text mining can also be considered as an instance of Web Mining.

Web Structure Mining (WSM)

WSM tries to discover the model underlying the link structure of the Web. The model is based on the topology of the hyperlinks. This model is used to categorize the Web pages and is useful to generate the information such as the similarity and the relationship between different Web sites/ pages. WSM is also being used to discover the authority sites and the overview/hub sites for a particular subject.

Web Usage Mining (WUM)

WUM mines knowledge from the secondary data on the web, *viz.* the usage data unlike WCM and WSM that operate on the real/primary data on the web. The web usage data includes the data from the Web Server logs; proxy, browser logs, registration data, user queries, bookmarks data, mouse clicks and scrolls and any other data as the results of user interactions on the Web.

WUM may be used to learn user profiles in adaptive interfaces (personalized) or in generalized access pattern tracking which can be applied in system improvement, site modification, business intelligence and usage characterization.

Web Mining Applications

One of the most important and well researched applications of Web Mining is Web Search. Google is one of the most popular and widely used search engines. It provides users an easy access to over 25 billion web pages that it has indexed on its server. Google also provides various other services besides Web Search, e.g. Google Toolbar, advertising by Google (only relevant text ads), Google Mail, Google News, etc.

Other applications include personalized B2C E-Commerce (e.g. Amazon.com), understanding Web Communities (e.g. AOL), understanding Auction behavior (e.g. e-Bay), Personalized Portal for the Web (e.g. MyYahoo), Digital Libraries and automatic citation indexing (e.g. CiteSeer).

Web Mining-Tools and Techniques

As mentioned before, Web Mining Research is an intersection of research in various disciplines. Many Machine Learning techniques have proved to be useful in Web Mining applications like neural networks, genetic algorithms, etc. Data Mining Techniques like classification and clustering the Web documents and users of the Web have been successfully applied in many systems. Other fields that contribute to Web Mining Research are Databases (developing query language interface for the Web), NLP, Bibliometrics, and Sociology etc.

Research Issues & Challenges

Various research issues posed by Web Mining have been discussed in [5]. Developing good Web Metrics and Measurements, countering spam attacks in Web Searches, personalized Web Searches, temporal web mining, click stream analysis, etc. are some of the research areas. Privacy and Spamming are some of the Challenges that the Web

Mining researchers have to face besides the major challenge posed by the very nature of the Web, viz. Huge, diverse, dynamic and unstructured repository of heterogeneous data.

1.3 Web Personalization

"If I have 3 million customers on the Web , I should have 3 million stores on the Web."

-Jeff Bezos, CEO of Amazon.com™

This statement is the essence of what is termed 'personalization'. Web Personalization is defined as any action that adapts the information or services provided by a Web site to the needs of a particular user or a set of users. In terms of the fast emerging area of Customer Relationship Management, personalization enables e-business providers to implement strategies to lock in existing customers and win new customers. Web Personalization is a broad area which broadly covers the following three types of personalized web services:

- Customization
- Recommender Systems (e.g. Amazon[1])
- Adaptive web sites

Three aspects of a Web site that affect its utility as a service provider are: the content provided on the website , the layout of the individual pages and the overall structure of the Website; any/ all of these aspects can be personalized/ tailored to meet the needs of individual users/ user groups. Web Personalization technology involves software that learns patterns, habits and preferences.

Recommender Systems

Over the years, since its inception, the role of the Web has changed and evolved quite rapidly and 'vividly'. The role that warrants sufficient attention from the researchers and technology developers is its role as a medium for e-commerce. The business opportunities provided by the Web are enormous—both for the customers and the businesses. The customers now have more bargaining power as they have more options to choose from. But more the options, more is the confusion and that is what the web

customers face today – *Information Overload.* Search Engines, which were initially developed to tackle this daemon (of information overload), have themselves fell prey to it. Any Web search returns millions of matching *documents*, and again, the user faces the giant task of analyzing, choosing and deciding what suits his needs the most. And sometimes, in fact, many-a–time the user himself/herself doesn't know, what *exactly* he/she needs or what he/she might want and how to search for it.

What web-users need today, are *intelligent agents* that help them explore the Web for alternatives and filter out their preferences. AI community has come forward to *infuse this intelligence* into the Web. Many machine learning and AI techniques have been applied to help users to utilize the services and information on the Web effectively and efficiently.

Recommender systems are used by e-commerce sites to suggest products to their customers and to provide consumers with information to help them decide which products to purchase. Today, many websites have employed recommender systems in some form or the other to enhance their service value to the customers. Collaborative Filtering (CF) is one of the techniques that has been used in many recommender systems. It clusters users according to the similarity in their preferences and recommends items to a user according to the preferences of the other members of the cluster to which the target user belongs. CF is just one of the techniques that are used for generating recommendations. Recommender Systems are discussed in detail in Chapter 2.

1.4 Scope and Objectives of this Work

Artificial Immune Systems have been shown to be useful in unsupervised learning tasks like clustering. Since Collaborative Filtering is based on the 'Cluster Hypothesis' ,i.e. , similar users like similar items, AIS can be used for collaborative filtering in a recommender system. This dissertation proposes a design and algorithm for a recommender system that user a 'fuzzy' artificial immune system model for information filtering. The fuzzy set properties are used in simulating immune system dynamics to meet the challenge of limited computer resources. A prototype movie recommender system has been developed using the proposed design.

1.5 Organization of Dissertation

The remainder of this dissertation is organized as follows: Chapter 2 gives a brief overview of the Recommender Systems Technology in section 2.1. Section 2.2 describes the fuzzy artificial immune system model. Chapter 3 describes the proposed architectural design and the algorithm (FAIR) to develop a recommender system based on the fuzzy AIS model. Chapter 4 comprises of the implementation details and results of the movie recommender system that has been developed using the framework described in Chapter 3. Chapter 5 presents the conclusions and suggestions for future work.

Chapter 2

Background

This chapter is divided into two sections. Section 2.1 is a discussion on recommender systems technology; various issues and approaches to the same have been presented along with the task components that constitute a recommender system. In section 2.2, the fuzzy AIS algorithm is discussed, on which this work is based.

2.1 Recommender Systems

Recommender Systems have caught the fancy of many researchers as well as the e-businesses due to many reasons, the chief reason being the competitive advantage such a service provides to the businesses like an efficient and motivated sales force, at much less effort and at such a large scale , especially when the competitors are just a click or two away.

The information revolution and rapid development in the field of e-commerce has brought about a paradigm shift from *database marketing* to *one-to-one marketing.* Database marketing refers to the process of maintaining a customer database and segmenting the customers based on their demographic attributes or their purchase history. The marketing activity is then customized according to needs of each customer segment as a whole. One-to-one marketing, on the other hand, refers to personalization of services and products for each individual customer; it attempts to improve the nature of marketing by using technology to assist businesses in treating each customer individually.

Earlier, the marketers used data-mining and data-analysis tools as decision support systems to learn user preferences and various behavioral patterns of their (prospective) customers to frame their marketing strategies but with the proliferation of e-commerce, and because of the paradigm shift mentioned above, these decision-support systems have moved closer to the customers by interacting directly with them through Web Interface

and playing the role of virtual salesperson without any need for marketer's intervention. Recommender systems are a powerful means of extracting additional value for a business from its customer databases. Recommender Systems enhance e-commerce sales in 3 ways :

- Converting browsers into buyers
- Increasing Cross-sell
- Building Loyalty

Today recommender systems (in various forms) are being used in many domains like product recommendation (e-commerce sites) like books (e.g. Amazon[1]), movies(e.g. Movielens) , music (e.g. Ringo[26]), news recommendation(e.g. Grouplens), web site / webpage recommendation (e.g. Syskill/Webert, WebWatcher [13], etc.), document recommendation, email filtering (e.g. Tapestry), ad targeting etc. For various prevalent recommender systems technologies and issues [24], [25], [17] and [8] can be referred. These survey papers have also discussed various commercial recommender systems.

The main tasks in a recommender system can be classified broadly into four tasks-*profile representation, profile adaptation, profile exploitation* and *making recommendations. User Profile* is the information about a user that may contain demographic information, user preferences or interest. A user profile or user model is defined as a set of information structures designed to represent one or more of the following elements (as quoted in [Frias-Martinez, et al]): representation of assumptions about the knowledge, goals, plans preferences, tasks and/or abilities about one or more types of users:

- representation of relevant common characteristics of users pertaining to specific user subgroups (stereotypes)
- the classification of a user in one or more of these subgroups
- the recording of user behavior
- the formation of assumptions about the user based on the interaction history
- the generalization of the interaction histories of many users into stereotypes.

User profiles are maintained by e-commerce sites to provide personalized services and recommendations to its users based on their individual interests.

I. Profile Representation

All other steps in the recommendation process rely on the method used to represent user profiles. Many methods for modeling user preferences have been experimented with, some of them being:

- User-item ratings matrix –(e.g. GroupLens [14,22], Ringo , Firefly, CDNow, Amazon)
- Demographic Features
- Purchase history –(e.g. CDNow, Amazon)
- Feature Vector (Boolean/ Weighted/ Probabilistic) (Features can be demographic, related to user interactions on the Web, or any other suitable preference indicator) –(e.g. Fab, Letizia, MovieLens, Syskill,Webert, WebWatcher)
- Classifiers (decision trees, inducted rules, ANNs etc.)- (e.g. Syskill, Webert)
- Weighted n-grams
- Any combination of the above methods

The choice of method selected to create user profiles depends on the kind of *inputs* to the recommender systems which can be *implicit* or *explicit* or a combination of both.

Explicit input from the user can be of the following types:

- Demographic and preference information at the time of registration
- Ratings for items purchased.
- Text Reviews / Comments
- Feed-back on recommender system's utility and performance, etc.

Implicit input from the user can be in the following forms

- User -Purchase history

- Navigation History
- Various actions/ interactions- pages visited, mouse-scroll, time-spent viewing a page, saving a document, bookmarking, deleting, resizing window, etc.

Each of these input techniques has its own pros and cons, e.g. users may be reluctant to divulge personal information or may not have time to rate items/give feedback. Implicit information like navigation patterns may require complex web usage mining techniques to process. A particular input technique or a combination of techniques is used depending on the domain and the goals of the Web site and resources available.

Profile Learning Techniques

The way a profile is created depends on the form of input data and the form of profile representation. Sometimes unstructured input data has to be preprocessed before using it for profiling .The initial profiles are created either manually (by the users themselves) while registering or while using the system to search for an item. In some systems the initial profiles are created by using user's initial interactions as training sets e.g. in systems that use classifiers as profile representations Various classifier models have been used for user profiling such as neural networks, Bayesian networks, decision trees , association rules, etc.

Profile learning techniques construct the user profiles (in the selected representation format) by taking users' relevance feedback into account. *Relevance Feedback* refers to the feedback from the user on how useful the recommendations have been. This feedback, which can be explicit or implicit , is useful for the recommender system in profile learning as it helps the system to fine- tune the user profiles according to user's interests.

Some systems do not require a profile learning technique, e.g. The systems which keep purchase history or ratings history information and use this information to represent profiles. Also, the systems which segment users based on demographic information don't need a separate profile learning technique.

II. Profile Adaptation/ Modification Techniques

Since user interests and preferences change over time, recommender systems must adapt to the ever-changing needs and moods of the users by updating the user profiles as and when required. Some systems let the user update her profile herself. Some systems use evolutionary models or other machine learning techniques for the same.

Relevance feedback from the user doesn't only help to create/ learn initial profiles, but also helps the recommendation engine to adapt these profiles according to changing interests of the users.

III. Profile Exploitation Techniques

Profile Exploitation techniques refer to the information filtering techniques that make personalized recommendations to users based on their profiles. The information filtering can be done in many ways, such as:

- Demographic Filtering
- Association Rules Based Filtering (market-basket analysis)
- Content-Based Filtering
- Collaborative Filtering
- Hybrid Filtering

Demographic filtering

Demographic filtering groups the customers into segments based on demographic features like age, income-group, education, occupation, etc and makes recommendations based on this segmentation. The problem with demographic filtering is that its too general and non-adaptive.

Association Rules Based Filtering

Association Rules Based Filtering is one of the most commonly used *data mining* techniques in e-commerce applications which basically deals with finding association rules between a set of co-purchased products, i.e. discovering association between two sets of products such that the presence of some products in a particular transaction implies that products from the other set are also present in the same transaction.

An association rule is an expression of the form X=>Y where X and Y are subsets of $A = \{l_1, l_2, \ldots l_m\}$, the set of items. We say that X=> Y holds with confidence s if s % of transactions in Database *D* that support X also support *Y*. The rule X=>Y has *support s* in the transaction set T if *s%* of transactions in *T* support X U Y. The rule is said to have a *confidence c* if *c%* of the transactions supporting *X* also support X U Y. Given a transactional database *D*, the problem of mining association rules is to discover all rules that have support and confidence greater than or equal to the user-specified minimum support *min_sup* and minimum confidence *min_conf*, respectively.

Association rules can be used to develop top-N recommender systems in the following way: For each one of the N customers we create a *transaction* containing all the products that they have purchased in the past (this purchase-history can be used to represent user profile, in this case). We then, use an association rule discovery algorithm to find all the rules that satisfy given minimum support and minimum confidence constraints. Now, for each customer *u* that we will like to find his/ her top-N recommended products we proceed as follows:

First, we find all the rules that are supported by the customer (i.e., the customer has purchased all the products that are in the left-hand-side of the rule). Let P_u be the set of unique products that are being predicted by all these rules and have not yet been purchased by customer *u*. Next, we sort these products based on the confidence of the rules that was used to predict them, so that products predicted by rules that have a higher confidence are ranked first. Note that, if a particular product is predicted by multiple rules, we use the rule that has the highest confidence. Finally, we select the first N highest ranked products as the recommended set. This kind of association rule mining is also referred to as market-basket analysis. Traditionally, market-basket analysis has been used for following marketing tasks, besides generating recommendations:

- Planning marketing /advertising strategies
- Catalog Design
- Designing different store layouts
- Designing discounts/sales offer combinations.

Content-based Filtering

In Content-based Filtering (CBF), the system recommends items that the user has liked in the past or that match the user's current search specification. Various similarity measures have been used in this user-item matching step like, keyword based similarity, cosine measure, nearest neighbor algorithms, etc. Various Information Retrieval Techniques come in handy for content based filtering. But, CBF suffers from some limitations, such as:

i. Overspecialization
ii. No serendipitous/ novel finds
iii. Only the targeted user's ratings are taken into account which may not be a good indicator of her preferences especially when she might not know of items that may potentially interest her. Also, a single user's ratings are very sparse (with respect to the whole database of items available)
iv. CBF is difficult to perform on items that are hard to be analyzed automatically, such as movies, ideas, music, etc.

Collaborative filtering

Collaborative filtering (CF) makes recommendation to a user based on the preferences of the users which are 'similar' to the target user. 'Similar', here, refers to the users who have interest and preferences similar to the target (active) user and is referred to as the *neighborhood* of the active user. Neighborhood formation is the essence of any CF-based recommender system. The main goal of neighborhood formation process is to find , for each customer u, an ordered list of l customers $N=\{N_1, N_2, ..., N_l\}$ such that $u \notin N$ and similarity(u,N_1) is maximum, similarity(u, N_2) is the next maximum and so on . The similarity between two users can be measured using similarity measures like correlation, cosine measure. Distance measures can also be used to measure similarity, indirectly; here, the distance between N_1 and u will be the minimum, between N_2 and u is the next minimum and so on. After computing similarity / proximity between the users, neighborhoods are formed. There are many schemes for neighborhood formation depending on the algorithm/technique used for the same, e.g. center-based neighborhood, aggregate neighborhood, etc. Centre based neighborhood scheme forms a neighborhood

for a particular customer , by simply selecting l 'nearest' other customers .Here nearest means most-similar/ least-distant according to the similarity measure/ distance measure used. Aggregate based neighborhood formation scheme works as follows: First, the closest neighbor of u is picked and added to its neighborhood. Now, the centroid of this neighborhood is computed. The next user to be added to the neighborhood is the user who is nearest to this centroid; after adding this user to the neighborhood, the centroid is recomputed and the next neighbor selected based on its proximity to this newly computed centroid. This process is repeated until l users get added to u's neighborhood. The affinity to the neighbors can be computed using many methods , such as, similarity weighting, significance weighting, variance weighting, etc. These methods have been discussed in detail by Herlocker *et al* [10].

Collaborative Filtering is, essentially, the automation of 'word of mouth'. While making recommendations, it takes into account the opinion of other users as well and is capable of producing high-quality and useful recommendations. CF methods can take into account the factors like quality, taste and preference because it is based on community (human) input; this is not possible in pure CBF as computers can analyze only content and not these factors. Also, CF can recommend items that are not related to the items the user is searching for but, that might potentially interest him which is not possible in pure content-based recommendations. But, CF has its own set of limitations, some of which are:

i. Cold-Start Problem: It has problems recommending when the system is new and no items have been rated. Same problem arises when a new item is added to the database since it's not been rated by any user. A new user also poses a similar problem because the CF system has problem in finding a neighborhood of similar users for the new user since his preferences are not known.
ii. Recurring Startup Problem: This is related to problem i. In domains like online newspapers, news articles are generated every day which have no ratings. Recommending such articles is a challenge for CF based recommenders.

iii. Lack of Transparency: This is because the reason why a particular product is recommended may not be clear. It affects the trustworthiness of a recommender system because the users are not sure whether the recommendations are purely CF generated or have been manipulated for some profit goals.

iv. Sparsity: In a typical CF-based recommender system, the input data is usually a user-item ratings matrix (ratings history) or a user-items purchase matrix(purchase history). In big commercial systems, even an active customer may have rated/ purchased well under 1% of the items in the whole database. This is known as the problem of sparsity/ reduced-coverage. Therefore a CF algorithm may not be able to make any recommendations for a particular user. Also , the accuracy of recommendations suffers due to reduced coverage.

v. Scalability: CF algorithms are based on neighborhood formation and this computation grows both with number of customers and number of items leading to scalability problems.

vi. Synonymy: Pure CF systems ignore the latent similarity (that may exist) between certain items while finding neighbors. Incorporating this information could considerably improve the quality of recommendations.

Hybrid Filtering

Efforts have been made to combine CBF and CF in a fashion so that each technique complements the other.

IV. Making Recommendations

The outputs of recommender systems can be in any of the following forms;

- Suggestions (one or many items)
- Text reviews / Comments
- Predictions
- Statistically summarized ratings / purchased history(e.g. bestsellers lists)
- Any combination of the above techniques.

Soft Computing Approach to User Modeling in Recommender Systems

Machine Learning techniques have been applied to recognize regularities in user traits and interactions while interacting with recommender systems and to integrate them as part of the user model/ profile. The limitations of traditional machine learning techniques for modeling human behavior led to the introduction of Soft Computing (SC) for User Modeling (UM). Various SC technologies that have been used for user modeling/ profiling have been discussed in [9], viz., fuzzy logic, neural networks, genetic algorithms, fuzzy clustering, neuro-fuzzy systems, etc. Web Mining techniques in soft computing framework is also discussed in [21]. SC technologies provide an approximate solution to an ill-defined problem and can create user models in an environment, such as a hypermedia application, in which users are not willing to give feedback on their actions and/or designers are not able to fully define all possible interactions. Human interaction is a key component of any hypermedia application, which implies that the data available will be usually imprecise, incomplete and heterogeneous. In this context SC seems to be the appropriate paradigm to handle the uncertainty and fuzziness of the information available to create user models.

The elements that a user model captures (goals, plans, preferences, common characteristics of users) can exploit the ability of SC of mixing different behaviors and capturing human decision processes in order to implement a system that is more flexible and sensible in relation to user interests. Different techniques provide different capabilities. For example, Fuzzy Logic (FL) provides a mechanism to mimic human decision-making that can be used to infer goals and plans; it defines a framework in which the inherent ambiguity of real information can be captured, modeled and used to reason with uncertainty. Due to its ability to handle uncertainty it is used in combination with other machine learning techniques in order to produce behavior models that are able to capture and to manage the uncertainty of human behavior.

An AIS based CF model that uses fuzzy logic for user profiling is discussed in the next section.

2.2 Fuzzy AIS Model

Most organisms' cognitive systems and natural immune systems seem to have almost unlimited ability to learn new concepts. But despite all the advances in hardware and software industry, learning ability and performance of AI systems is still largely limited by the availability of resources like memory, processor speed, etc.. According to the stringent practical demands of today's data mining applications, AINE's [28] information processing and results are still considered extremely expensive both from a computational and storage point of view.

Nasaroui *et al* have proposed a new ARB model in [18,19]. The *fuzzy* ARB represents not just a single data item, but instead defines *a fuzzy set* over the domain of discourse consisting of the training data set. The fuzzy set's shape can be any continuous function that decreases with distance from the center of the ARB (prototype / best exemplar). It differs from the AINE algorithm in the following three aspects:

i. Each fuzzy ARB is allowed to have its *own* scale / radius of influence (σ_i) instead of a single *NAT* threshold or all ARBs
ii. crisp thresholding is no longer necessary because the fuzzy membership function will gradually exclude antigens that are far away from the prototype
iii. the fuzzy membership function serves also as a *robust* weight function that will decrease the influence of noise and *outliers*. For ARB_i, they have define the following weight/membership function:

$$w_{ij} = \exp(-\frac{d_{ij}^2}{2\sigma_i^2}) \qquad (2.1)$$

The *stimulation level* is defined as the density of the antigen population around a certain ARB:

$$s_i = \frac{\sum_j w_{ij}}{\sigma_i^2} = s_{i\_antigens} \tag{2.2}$$

According to the authors, the above definition has certain desired features from an immune system point of view: the numerator acts like a *stimulation* factor trying to cover as many antigens as possible, while the denominator will limit the covered area of the ARB, thus acting like a *suppression* mechanism, i.e., numerator by itself promotes *generalists* (cover as many antigens as possible), while the denominator promotes *specialists* (cover a smaller area). The fraction is expected to achieve a certain *balance* between the two desired extremes.

Unlike the original AINE ARB's, each fuzzy ARB scale value (σ_i) is dynamically updated in each iteration to maximize its *stimulation* level (and hence *survival chances*). By setting $\frac{\partial s_i}{\partial \sigma_i^2} = 0$, we obtain

$$\sigma_i^2 = \frac{\sum_j w_{ij} d_{ij}^2}{\sum_j w_{ij}} \tag{2.3}$$

The ARBs located in each other's influence regions should either be merged to limit the population growth or pulled away from each other to explore new areas (for instance by penalizing their stimulation level via a second suppression term).

Stimulation can be been redefined as:

$$si = si\_\text{antigens} - \beta(t) \,.\, si\_\text{neighboring\_antibodies} \tag{2.4}$$

The modified ARB scale update equations would become:

$$\sigma_i^2 = \frac{\sum_j w_{ij} d_{ij}^2 - \beta(t) \sum_{ARB_k} w_{ij} d_{ik}^2}{\sum_j w_{ij} - \beta(t) \sum_{ARB_k} w_{ik}} \tag{2.5}$$

The fuzzy AINE algorithm is as follows:

```
Initialize fuzzy AINE (ARB pop and σi²) using a cross section
of the input data;
Load antigen population = remaining training data;
Repeat Until termination condition
{
    Repeat for each antigen
    Present antigen to each fuzzy ARB in network and update wij
    using equation (2.1);
    Repeat for each fuzzy ARBi
    {
      Compute ARBi's stimulation level using equation (2.4);
      Update σi² using equation (2.5);
    }
    Allocate B cells to fuzzy ARB's based on stimulation
    level;
    Remove weakest ARBs (0 B cells) from population;
    Clone and mutate remaining fuzzy ARBs;
    Integrate new fuzzy ARBs into fuzzy AINE;
}
Consolidate final ARB population;
```

Note: in the cloning process, the clone inherits the scale value of the parent ARB.

In their experiments with the algorithm, the authors find that the suppression term that penalizes close ARBs which prevents the very good ARBs from dominating the less

good ones, since their stimulation cannot grow beyond a limit. The system exhibits a niching mechanism and encourages diversity. [19] also describes how this algorithm was successfully applied to a web usage mining task.

In our work, we have tried to exploit these capabilities of a fuzzy-matching mechanism in a recommender system application based on AIS dynamics. In chapter 3, we have presented an algorithm that is based on the concept of fuzzy ARBs(antibodies) and approximate nature of the interactions between antibodies and antigens as well as the inter-antibody interactions (idiotypic network assumption).

Chapter 3

Fuzzy AIS for Recommender Systems

3.1 Framework for AIS Applications

Computing paradigms like neural networks, genetic algorithms, fuzzy systems have matured as procedures since the methods to implement these techniques have more or less, become standardized.

To establish AIS in the main realm of soft-computing, a framework has been proposed by deCastro and Timmis [7] for developing AIS applications. In this paper AIS is discussed in context with other soft computing approaches like ANNs, EA and Fuzzy Systems while elaborating upon the various components of the framework.According to them, a framework to design a computationally inspired algorithm requires, at least, the following basic elements:

- *A representation for the components of the system;*
- *A set of mechanisms to evaluate the interaction of individuals with the environment and each other.* The environment is usually simulated by a set of input stimuli, one or more fitness function(s), or other mean(s);
- *Procedures of adaptation that govern the dynamics of the system*, i.e. how its behavior varies over time.

This is the basis of the proposed framework to a layered design of artificial immune systems which is composed of following procedure (refer Figure 2.): choose a representation to create abstract models of immune organs, cells and molecules; select a set of functions, termed affinity functions, to quantify the interactions of these artificial elements, and define a set of general purpose algorithms to govern the dynamics of the AIS.

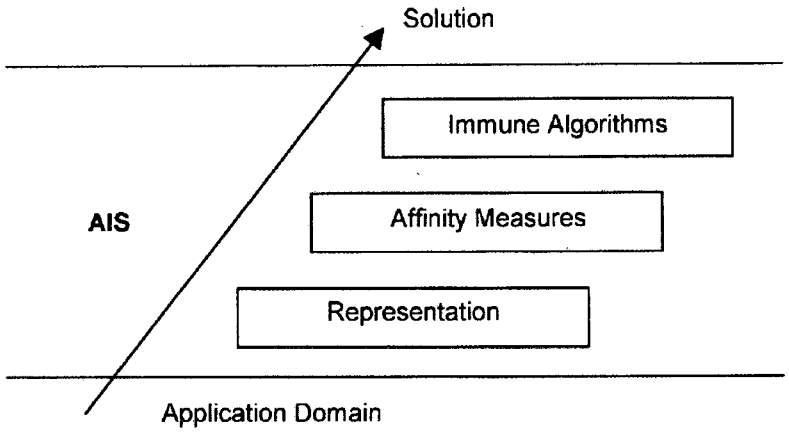


Figure 2. Framework for AIS Applications

Representation

Out of many components of the natural immune system, the immune cells- B-cells were considered as most relevant to the computational intelligence research because these cells are the basis of the properties of immune system which characterize intelligence. Antibodies are the molecules on the surface of B-cells that act as receptors and help in antigen recognition. Therefore the B-cells and the antibodies/ antigens are the elements that have to be modeled and used to create AIS. Many of the AIS models have assumed that a B-cell contains only one type of antibody on its surface which implies that B-cell and antibody can be treated as synonyms and there is no need to model them separately in the system.

In biological immune system, the degree of binding between an antibody and an antigen is determined by the complementarity in their shapes and how much they fit into each other (like lock and key). The set of features that describe the relevant properties of a molecule from a recognition perspective is termed its *generalized shape.* The generalized shape of an antibody is described by a set of L parameters. This means that we can represent an antibody/ antigen as a point in L-dimensional *shape-space* (a vector of L features) Also, it is assumed that each antibody interacts with all antigens whose complements are within a small surrounding region characterized by an *affinity threshold* which indicates whether the antibody in question would bind with an antigen or not.

To use AIS as computational model for a problem, the first step is 'representation', i.e. mapping of entities in the problem domain to the antibodies and antigens, which in turn are points in L-dimensional space. This shape space can be integer, real-valued, boolean or symbolic in nature depending upon the requirements of the problem at hand.

Affinity Measures

To model the basic function of antibody-antigen matching, an affinity measure needs to be selected. This is usually determined by the shape space representation chosen in the first step. If the co-ordinates of an antibody are given by $Ab = \langle Ab_1, Ab_2, \ldots, Ab_L \rangle$ and those of an antigen are given by $Ag = \langle Ag_1, Ag_2, \ldots, Ag_L \rangle$. The affinity can be estimated using any of the distance measures between two feature-vectors representing antibody/ antigen such as, Euclidean (equation 3.1), Manhattan distance (equation 3.2), Hamming distance (equation 3.3), etc.

$$D = \sqrt{\sum_{i=1}^{L} (Ab_i - Ag_i)^2} \tag{3.1}$$

$$D = \sum_{i=1}^{L} |Ab_i - Ag_i| \tag{3.2}$$

$$D = \sum_{i=1}^{L} \delta i \quad \text{where } \delta i = \begin{cases} 1 \text{ if } Abi \neq Abj \\ 0 \text{ otherwise} \end{cases} \tag{3.3}$$

Immune Algorithms

This step comprises of some iterative procedure(s) that determine the adaptive and evolving behavior of AIS over time. Algorithms based on immunological theories like clonal selection theory, negative selection mechanism, immune network theory have been proposed in the literature. The choice at this level can be made based on what type of behavior is expected from the AIS, i.e. it depends on the application domain and the nature of the problem being modeled.

3.2 AIS Framework for a Recommender System Application

An immune system based recommender system can be designed in a layered manner as proposed in the above mentioned framework. Following is a description of how the system is modeled.

Representation

This step requires mapping the main components of our application domain to the components of an immune system. User profiles are the most important part of a recommender system. A profile may consist of many features that represent user interest and preferences, e.g. a set of votes, etc. These features can be conveniently modeled as an L-dimensional feature vector that represents the antibody/ antigen in our artificial immune system. Modeling user profiles as antibodies and antigens is quite intuitive if we look from a collaborative filtering (CF) perspective since antibody-antigen matching and inter-antibody interactions in an idiotypic network help in finding similar users, which is the essence of CF.

The active (target) user (to whom the recommendation has to be made) is presented to the system as an antigen. (From representation point of view, an antigen is modeled just like an antibody- as an L-dimensional feature vector).Immune algorithms and processes can be used to filter recommendations based on the preferences of users with similar interests (matching antibodies).

Affinity Measures

Since the antibodies/antigens are represented as feature vectors, distance measures like Euclidean distance, Manhattan distance etc. can be used to measure the distance (as discussed in the previous section). Affinity can be defined in terms of distance between two points in the shape space since smaller the distance, greater the affinity. Alternatively, methods like Pearson's correlation, Spearman's correlation or Cosine similarity can be used to determine the affinity between two points.

The Pearson's correlation is computed as :

$$corr_{a,b} = \frac{\sum_i (r_{ai} - \bar{r}_a)(r_{bi} - \bar{r}_b)}{\sqrt{\sum_i (r_{ai} - \bar{r}_a)^2 \sum_i (r_{bi} - \bar{r}_b)^2}} \tag{3.4}$$

where r_{ai} / r_{bi} is the value of the i^{th} feature of the user a/b. $\bar{r}_a / \bar{r}_b$ is the average of the values of all features for the user a/b.

Cosine Similarity is computed as:

$$\cos(\vec{a}, \vec{b}) = \frac{\vec{a}.\vec{b}}{\|\vec{a}\|^2 * \|\vec{b}\|^2} \tag{3.5}$$

Immune Algorithms

On presentation of the antigen, the composition of the AIS changes – in terms of antibody concentration, links between antibodies, etc. The selection of algorithms depends on the goals of the system, the type of domain , the type of input and profile representation.

3.3 Design of a fuzzy-AIS based Recommender System

The design of the proposed fuzzy AIS (FAIS) based recommender system is based on the framework mentioned in the previous section. The architecture of the system is illustrated in figure 3.

The databases and data sources maintained by the system include:

- *Users' Login Information*
- *Users' Profiles Database*
- *Item Database*
- *System Parameters*

The ***Users' Login Information*** contains the login information of each user. Each user is assigned a unique identifier. This is a necessary component for the system to provide personalized content-recommendations.

The ***Users' Profiles Database*** includes information related to user preferences. This may be in the form of ratings of items, purchase history etc. This database can also contain the demographic information about the users such as age, gender, occupation, income level, educational level, country, language, etc. Any/ all of the information can be used by the CF agent to generate recommendations.

This database consists of fuzzy profiles, where a *fuzzy profile* is a representative profile for more than one user. The degree to which, this fuzzy profile represents an actual user is indicated by the associated *fuzzy membership value*.

Item Database consists of information about the items that the system provides/ recommends. Besides item identifier, this database can also include content information about the items so that content-based filtering can also be combined with collaborative filtering to give better recommendations and dealing with problems like cold-start, unique users, sparsity, etc.

System Parameters database consist of the values of various parameters for the fuzzy artificial immune system like number of antibodies, initial concentration of antibodies and antigens, initial number of antibodies, minimum and maximum allowed concentration of antibodies, similarity measures to be used, various thresholds, etc. These parameters can be modified as and when required to improve the performance of the system.

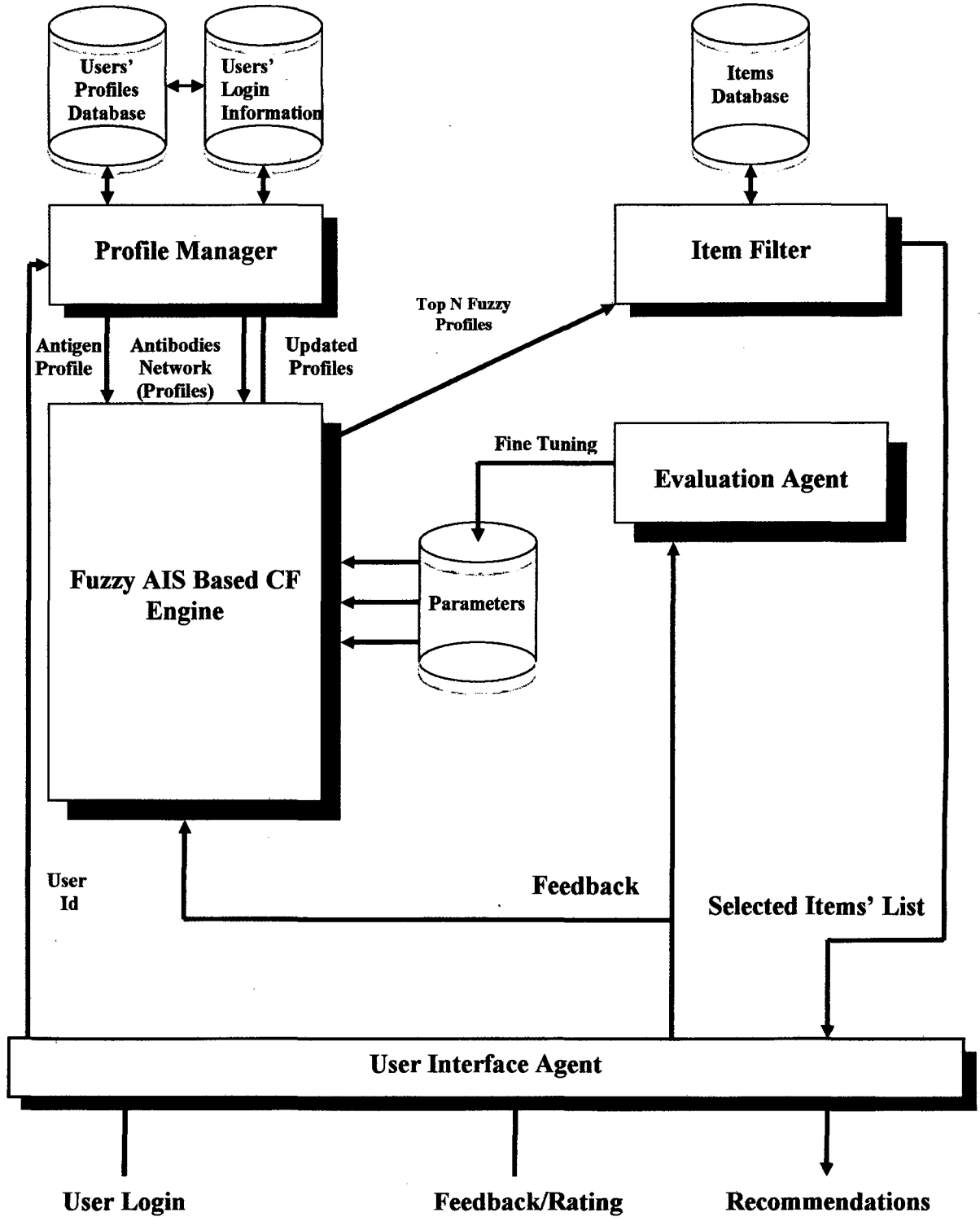


Figure 3: Architecture of the Fuzzy AIS Based Recommender System

The major task components of the system are:

- *Profile Manager*
- *User Interface Agent*
- *Fuzzy AIS based Collaborative Filtering Engine*
- *Item Filter*
- *Evaluation Agent*

The ***Profile Manager*** is a system module that maintains profiles of the users and interacts directly with the users' profiles database. When a user logs in, it is the task of the profile manager to retrieve its profile from the database and provide it as an antigen to the FAIS.

The ***User Interface Agent*** is the interface between the system and the users. A user logs in the system and receives the recommendations generated by the system through the interface provided by this agent. Feedback from the user (in the form of ratings, etc.) is also given to the system through the user interface agent.

The ***Fuzzy AIS based CF Engine*** is the core component of this recommender system. The *profile manager* provides the engine with the antibodies network and the antigen (the profile of the current/ target user). The system responds to the change caused by the antigen and the best matching antibodies / fuzzy profiles are provided to the *Item Filter* for generating recommendations. The system also adapts based on the feedback/ ratings provide by the users for the recommended items and incorporates this information in the profile of the current user and hence the FAIS. The details of the algorithm employed in the CF engine are given in the next section.

The ***Item Filter*** module interacts directly with the *Item Database* and filters most highly rated items based on the selected fuzzy profiles provided by the *CF engine* .These items are then sorted based on their estimated relevance and top-N items are presented to the user as recommendations through the *User Interface Agent.*

The ***Evaluation Agent*** module evaluates the performance of the recommender system based on the feedback from the user. It can be in the form of ratings or purchase of items. A number of evaluation methods have been suggested in the literature [23], [2,3]- some of them are discussed below:

i. *Prediction Accuracy*: Mean Absolute Error (MAE) is a very widely used measure of prediction accuracy. It is computed as:

$$MAE = \frac{\sum |actual - predicted|}{n_p} \tag{3.6}$$

where n_p is the number of predictions.

ii. *Mean Number of Recommendation*: total number of unique items rated by the neighbors.

iii. *Mean Overlap Size*: number of recommended items that user has also rated.

iv. *Mean Accuracy of Recommendations*: Each overlapped item has an actual rating and a rating predicted from the neighbors. The overlapping items are ranked on both actual and predicted rating. The two ranked lists can be then, compared using Kendall's Tau. This measure reflects the level of concordance in the lists.

This is computed by using the following formulae:

$$\tau = 1 - \frac{4N_D}{n(n-1)} \tag{3.7}$$

$$N_D = \sum_{i=1}^{n} \sum_{j=i+1}^{n} D(r_i, r_j) \tag{3.8}$$

$$D(r_i, r_j) = \begin{cases} 1 \ if \ ri > rj \\ 0 \ otherwise \end{cases} \tag{3.9}$$

where n is the overlap size and r_i is the rank of the item i as recommended by the neighborhood. Here i refers to the antigen rank of the item. N_D is the number of discordant pairs .D is set to one if the rankings are discordant.

v. Some standard metrics used by Information Retrieval Community can also be used to evaluate the system, such as recall and precision [23] .The data set is divided into two parts: the training set and the test set.

The Filtering algorithm works on the training set and generates a set of recommendations which is called the top-N set. The actual ratings for the test set data are hidden and not presented to the filtering algorithm. The ratings from the test set data are matched with the top-N set. The products that appear in both the sets (with MAE less than a predefined threshold) constitute the hit set.

Recall in this context, is defined as the ratio of the hit-set size to the test-set size.

$$recall = \frac{|test \cap top - N|}{|test|} \tag{3.10}$$

Similarly, *precision* is defined as ratio of hit-set size to the top-N size.

$$precision = \frac{|test \cap top - N|}{N} \tag{3.11}$$

These two measures are, however, often conflicting in nature. For instance , increasing the number N tends to increase recall but decreases precision. Since both these measures are important for estimation of recommendation quality, a combination of the two viz. the standard *F1 metric* is used. F1 metric is computed as:

$$F1 = \frac{2 * recall * precision}{recall + precision} \tag{3.12}$$

The metrics *recall, precision* and/or *F1 metric* are computed for each individual user and then an average is taken to estimate the metric value for the recommender system.

The *Evaluation Agent* evaluates the system based on one or more of these metrics and fine-tunes the system parameters to improve its performance. Automation of evaluation agent is not in the scope of this work. The system will be tested for its sensitivity to different parameters manually and the optimal values will be hard-coded in the implementation. However, this agent can be automated by using some technique, e.g. ANNs, GAs can be used to learn the optimal values for these parameters.

3.4 The Fuzzy AIS based CF Algorithm

3.4.1 Representation of System Components

The main components of the proposed fuzzy AIS (FAIS) system are the fuzzy antibodies organized in an idiotypic network. Each *fuzzy antibody* is representative of a number of user profiles. The degree to which a user profile is represented by a fuzzy antibody is given by the associated membership value.

Following are the data items that characterize antibodies:

- *Scale Value*: defines the radius of influence for a fuzzy antibody.
- A list of {*user-id, MValue*} pairs : membership values (*MValue*) are computed using formula 2.1
- Genotype: consists of one or more genes.
- Stimulation level (computed using formula 2.2)
- Concentration

Scale value, stimulation level and membership value (weight function) (has been discussed at length in section 2.1. (Note that, here we take an Antibody as a synonym for ARB and the number of resources of the ARB will be referred to as the concentration of the antibody.)

A *gene* consists of a vector of *{item, rating}* pairs where each item in the gene belongs to a single item category. Grouping the items into conceptual categories incorporates *content information* in the model and helps to overcome the shortcomings of collaborative filtering.

The item-ratings database is used to construct genes for each antibody. In case an (fuzzy) antibody represents two users with different ratings for the same item, the genotype would contain a single gene for that item, say $item_i$, with the rating $rating_i$ as the average of ratings given by the two users. This definition can be extended to cases in which a fuzzy antibody represents more than two users.

The *genotype* of an antibody contains several genes and is the data item that represents the user preferences. In our system, we have chosen to encode the users' ratings for items as the genotype and ignore the demographic information for user profiling.

The network structure is not stored explicitly in this model. However, the system exhibits idiotypic properties because the effect of neighboring antibodies in the system is taken into consideration while computing stimulation level of each antibody (this effect can be either stimulating or suppressing).The stimulation level, in turn, determines the concentration of a particular antibody in the system in the next iteration of the algorithm. The algorithm is iterative in nature and the iterations continue till the network stabilizes. The next subsection describes this algorithm in detail. This work is inspired by the work in Cayzer and Aickelin [2,3] for the application of AIS to recommender systems and Nasraoui et al [18,19] for the fuzzy AIS model.

3.4.2 The FAIR Algorithm

Following is the broad outline of the Fuzzy Artificial Immune System Based Recommender System (FAIR) Algorithm.

Algorithm Outline

1. Initialize Fuzzy Artificial Immune System(using a cross-section of input data.
2. Encode target user as antigen Ag.
3. **WHILE**(FAIS not stabilized) and (reviewers available)
4. Add next user as an antibody Ab.
5. Calculate affinity between Ab and Ag using formula (2.1)
6. Calculate affinity between Ab and other antibodies in the system.
7. **IF** affinity(Ab, $Antibody_i$)> theta, for some *i* **THEN** FuzzyMerge(Ab, $Antibody_i$)
8. Compute Stimulation Level for all Antibodies using formula (2.2)
9. Update Antibody Concentration based on the stimulation level.
10. Remove weak/weakest antibodies from the system.
11. **END WHILE**
12. Select N antibodies with highest stimulation level.
13. **FOR** each antibody selected in Step 12
14. Extract item information from genes.
15. **IF** affinity(Ag, $Antibody_i$)> theta, for some i **THEN** FuzzyMerge(Ag, $Antibody_i$).
16. **ELSE** Add Ag as an antibody to the FAIS.
17. **END FOR** (selected antibody)
18. Select M items with highest expected rating.
19. Present these M items as recommendations to the target user.
20. Update scale values for each antibody using the formula (2.3)

Algorithm Details

Step 1: Initialize Fuzzy Artificial Immune System(using cross-section of input data)

This initialization process would include following steps:

i. Create fuzzy antibodies from a randomly selected part of the data set .

ii. Initialize scale values for these antibodies.

iii. Initialize concentration of each antibody.

Step 2: Encode target user as antigen Ag.

The target user is the current user for whom the system has to generate recommendations. Here, we assume that antigens are presented to the system, one at a time. The *Profile Manager* searches the profile for this user and encodes it as an antigen. The antigen is encoded in the same way as a fuzzy antibody is, except for the fact that the antigen represents a single user , i.e., the fuzzy set defined by the antigen Ag is a singleton set. The scale value is initialized for the antigen too.

Step 3:

WHILE(FAIS not stabilized) and (reviewers available)

The system iterates through steps 4 through 10 till the system stabilizes or no antibody is left to be added to the system.

Steps 4, 5 and 6:

A fuzzy antibody is added to the system from the *Users' Profiles Database* in each iteration. Its affinity to the (current) antigen and to the other antibodies in the system is computed using the fuzzy membership formula mentioned in section 2.1. i.e. $w_{ij} = \exp(-\frac{d_{ij}^2}{2\sigma_i^2})$ where σ_i is the scale value of antibody i.

Step 7: IF affinity(Ab, $Antibody_i$)> theta, for some *i*

THEN FuzzyMerge(Ab, $Antibody_i$)

If the affinity between the newly added antibody to another antibody is greater than a threshold value- theta, it means that the two fuzzy antibodies are very similar and only one can be used to represent both of them. The FuzzyMerge Algorithm creates a new fuzzy antibody that is representative of both these matching antibodies.

The FuzzyMerge algorithm is as follows.

FuzzyMerge(I,J)

[1] Create a new antibody K

[2] Put all the genes, i.e. (item, rating) pairs from I and J into K such that:

[3] IF item in the gene is in $I \cap J$, Put it into K and let its rating be the weighted average of the two ratings; here the weights are the scale values of the two antibodies.

[4] Compute the $A_{I,K}$ <-affinity(I,K) and $A_{J,K}$<-affinity (J,K)

[5] Put the user members of I into K's User_List where the Membership value of user U MValue(K,U) is computed as MValue (I,U)* $A_{I,K}$.

[6] Repeat the above step for user members of J.IF user V of J is already a member of K, re-compute its membership value as Max(MValue(J,V),Mvalue(K,V)) (because we are doing the union of two fuzzy sets)

[7] Discard I and J.

[8] Add newly created fuzzy antibody K into the FAIS.

Note: Here, MValue(X,U) gives the membership value of user U in the fuzzy set defined by the fuzzy profile X.

In step 6, we take the union (logical *or*) of the two fuzzy sets. In [31] Yager introduced a new aggregation technique based on the ordered weighted averaging (OWA) operators. These OWA operators can provide for aggregations lying between the logical *or* and *and*. An OWA operator of dimension n is a mapping $F:R^n \rightarrow R$ that has an associated n vector

$W=[w_1,w_2,K,w_n]^T$ such that $w_i \in [0,1]$ *and* $\sum_{i=1}^{n} w_i = 1$.

The aggregation operator F is defined as:

$F(a_1,a_2,K,a_n) = \sum_{j=1}^{n} w_j.b_j$ where b_j is the j^{th} largest of a_i.

A fundamental aspect of this operator is the reordering step, in particular an aggregate a_i is not associated with a particular weight w_i, but rather a weight is associated with a particular ordered position of an aggregate. When we view the OWA weights as a column vector we shall find it convenient to refer to the weights with the low indices as weights at the top and those with the higher indices with weights at the bottom.

It is noted that different OWA operators are distinguished by their weighting function. As described in [31], there are three important special cases of OWA aggregations:

1. F^* . In this case $W = W^* = [1, 0, ..., 0]^T$
2. F_* . In this case $W = W^* = [0, 0, ..., 1]^T$
3. F_{Ave}. In this case $W= \left[\frac{1}{n}, \frac{1}{n}, K, \frac{1}{n}\right]^T$

It can easily be seen that

1. $F^*(a1, a2, ..., an) = \text{Max}(a_i)$
2. $F_*(a1, a2, ..., an) = \text{Min}(a_i)$

In Step [6] of FuzzyMerge , we have used F^* . Any variant of F can be used for the same.

Steps 8, 9 and 10: The stimulation level of all the antibodies is computed using the formula $s_i = \frac{\sum_j w_{ij}}{\sigma_i^2} = s_{i\_antigens}$. Based on this stimulation level, the concentration of each antibody is updated as:

$$Updated\ Concentration\ Value = k \log (s_i) \tag{3.13}$$

where k (concentration update constant) is a system parameter.

Steps 12 through 19:

Antibodies are ranked according to their stimulation level and the top-N (most highly stimulated) antibodies are selected. The genes / item information is extracted from these antibodies. M items having the highest rating are picked and presented to the target user as recommendations. In case the item is present in the genotypes of several antibodies, its rating is computed as a weighted average of all the ratings with weights equal to the corresponding *scale values* of the antibodies. This weighted average is called the expected rating of the item. Also, the antigen is added to the system as an antibody so that, next time it is encountered, the response is faster.

Step 20: The *scale values* of the antibodies are updated using the formula (2.4)

Chapter 4

Implementation and Results

In this work, we have tried to implement a prototype movie-recommender system based on the algorithm and design proposed in Ch.3. A part of EachMovie dataset was used for experimentation. *EachMovie* Dataset provided by Compaq research is a database that has been used as a standard dataset to test various recommendation techniques. The next section describes this dataset. The implementation details of our system are given in section 4.2. The experimental results are presented in section 4.3. In section 4.4, some inferences and discussion on the system are presented.

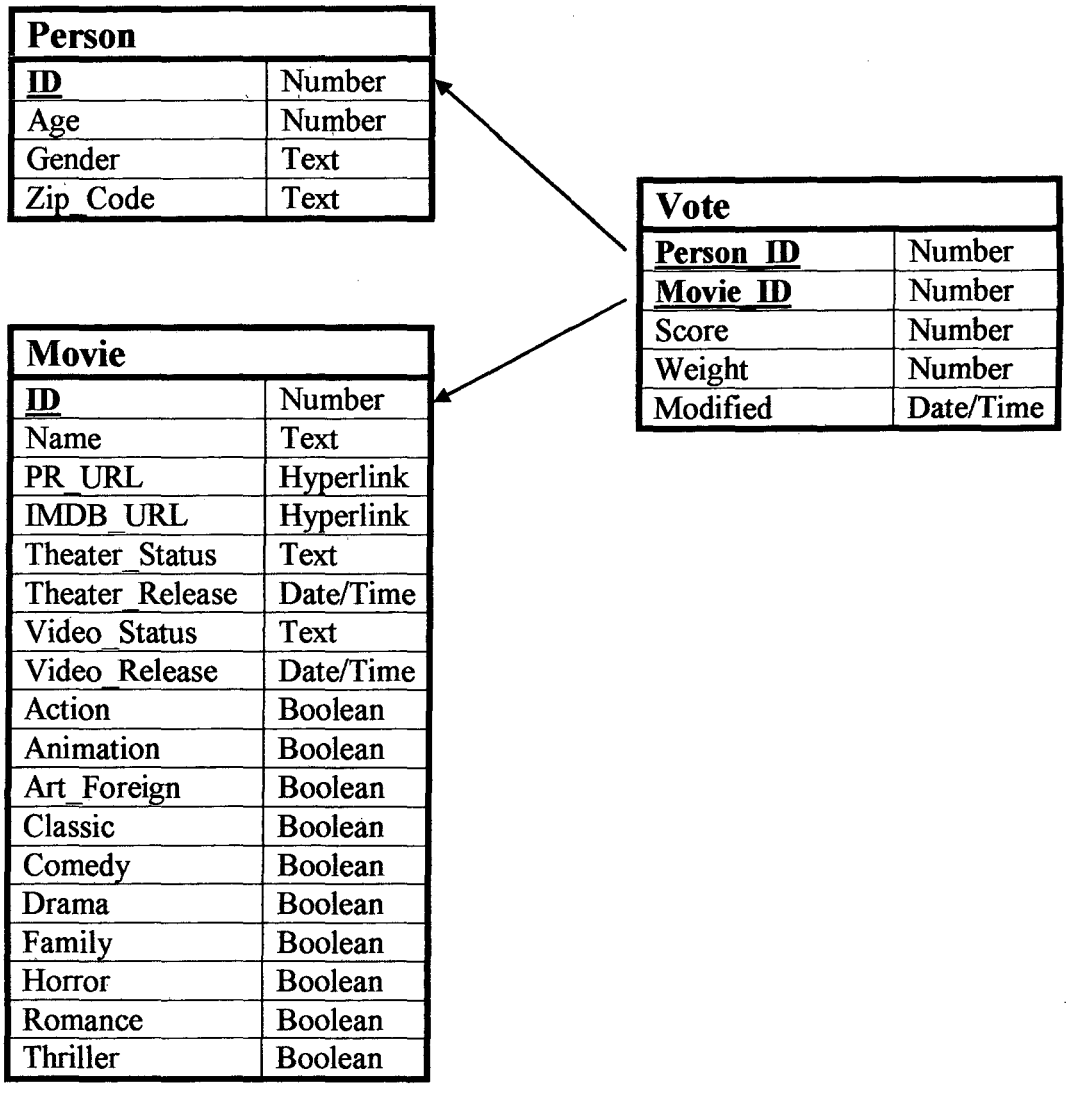
4.1 The *EachMovie* DataSet

The E*achMovie* dataset was collected by DEC (now Compaq) research over an eighteen month period. Although the project was closed in 1997, the data has been freely distributed for research. It consists of 72916 users, 1628 movies, and 2811983 movie votes. Votes indicate how a user rated a movie, ranging from extreme dislike (0.0) to sublime delight (1.0). EachMovie dataset has been used as a standard dataset for experimenting with various approaches for recommender system; for instance, Ujjin *et al* [29] have used a GA approach, Li *et al.* [16] have applied inductive learning to generate movie recommendations. The relational version of the dataset that we have used is an access database with three tables. Figure 4 describes the tables and how they are related.

The table *Person* includes the demographic information about the 1628 users of the system which includes age, gender (M/F) and zip code apart from the primary key attribute ID that uniquely identifies each user in the database.

The table *Movie* contains the information about movies which includes the title of the movie, URL information, theater/video status which can be "old" or "new". It also includes ten Boolean fields each for a different genre such as, action, comedy, thriller,

romance, etc. The values of these fields indicate whether or not a movie belongs to a particular genre. It is assumed that a movie can belong to more than one genre. Each movie has a unique id (field ID) which identifies the movie uniquely in the database.



Person	
ID	Number
Age	Number
Gender	Text
Zip_Code	Text

Vote	
Person_ID	Number
Movie_ID	Number
Score	Number
Weight	Number
Modified	Date/Time

Movie	
ID	Number
Name	Text
PR_URL	Hyperlink
IMDB_URL	Hyperlink
Theater_Status	Text
Theater_Release	Date/Time
Video_Status	Text
Video_Release	Date/Time
Action	Boolean
Animation	Boolean
Art_Foreign	Boolean
Classic	Boolean
Comedy	Boolean
Drama	Boolean
Family	Boolean
Horror	Boolean
Romance	Boolean
Thriller	Boolean

Figure 4: Relational Structure of the *EachMovie* Dataset

The table *Movie* contains the information about movies which includes the title of the movie, URL information, theater/video status which can be "old" or "new". It also includes ten Boolean fields each for a different genre such as, action, comedy, thriller,

romance, etc. The values of these fields indicate whether or not a movie belongs to a particular genre. It is assumed that a movie can belong to more than one genre. Each movie has a unique id (field ID) which identifies the movie uniquely in the database.

The table *Vote* represents a collection of votes/ratings given by a user to a particular movie. Each vote is uniquely identified by a (Person_ID, Movie_ID) combination which is the primary key for this table. The 'Score' field is a numeric value which can be a rating from the set {0.2, 0.4, 0.6, 0.8, 1.0}, i.e. a five-point scale. The 'Weight' field indicates whether or not the person has actually seen the movie or not. The 'Modified' field stores the date/time (timestamp) at which the rating was given.

In our application, we have ignored the weights and timestamp information. We have also not considered the demographic information of the user or the URL /status information about the movies. Category/ genre information about the movies has been incorporated in the system to give quality recommendations.

4.2 Implementation Details: A movie recommender system

The implementation of the above mentioned movie recommender system based on the proposed (FAIR) algorithm was done on an AMD Athlon, 650 MHz processor running Windows XP. The program was coded in JAVA™. The system has been developed using JDK 1.3 and is built on the top of the AIS library [4] developed by Steve Cayzer of HP labs. The cut-down version of EachMovie dataset provided with this library was used for experimentation. The library provides some standard interfaces and some implementing classes that together specify an API for developing AIS applications. The core component of the library is the package *model* that is comprised of some simple Java interfaces that represent the basic components of an AIS. The hierarchy of these interfaces is shown in figure 5.

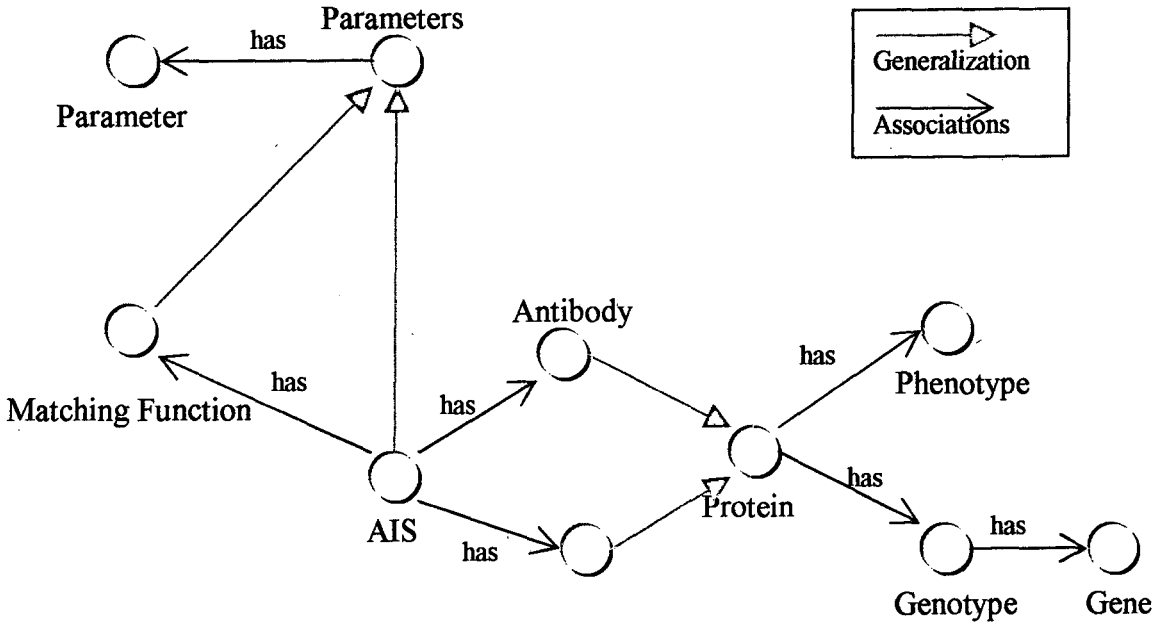


Figure 5: Class Diagram showing AIS *model* interfaces

There is an interface that represents the *AIS* itself which is composed of antibodies and antigens. The AIS also uses a *matching function* to calculate the binding affinity of antibodies and consists of the main algorithm that governs (artificial) immune system dynamics.. This decoupling allows one particular AIS model to be used with a range of matching functions. *Antibody* and *Antigen* are modeled as specializations of the *Protein* interface. Each protein has a *Genotype* and *Phenotype*. AIS systems are free to implement the genotype and phenotype in any way they choose, as long as there is a decode/encode pathway from one to the other. Some implementations don't require a separate phenotype representation.

Each genotype consists of a number of Genes. Again, the exact form of each gene is left unconstrained by the model. The *Parameters* interface is a useful device to allow an implementer access to an undefined set of attributes / parameters. The *Parameter* (singular) interface allows access to each undefined attribute as a tuple {name, type, value}.

A number of these interfaces are likely to have a common implementation in any application. A set of classes that conform to these interfaces have been packaged in another module called common. Figure 6 shows the common package, which provides canonical implementations of many of the *model* interfaces.

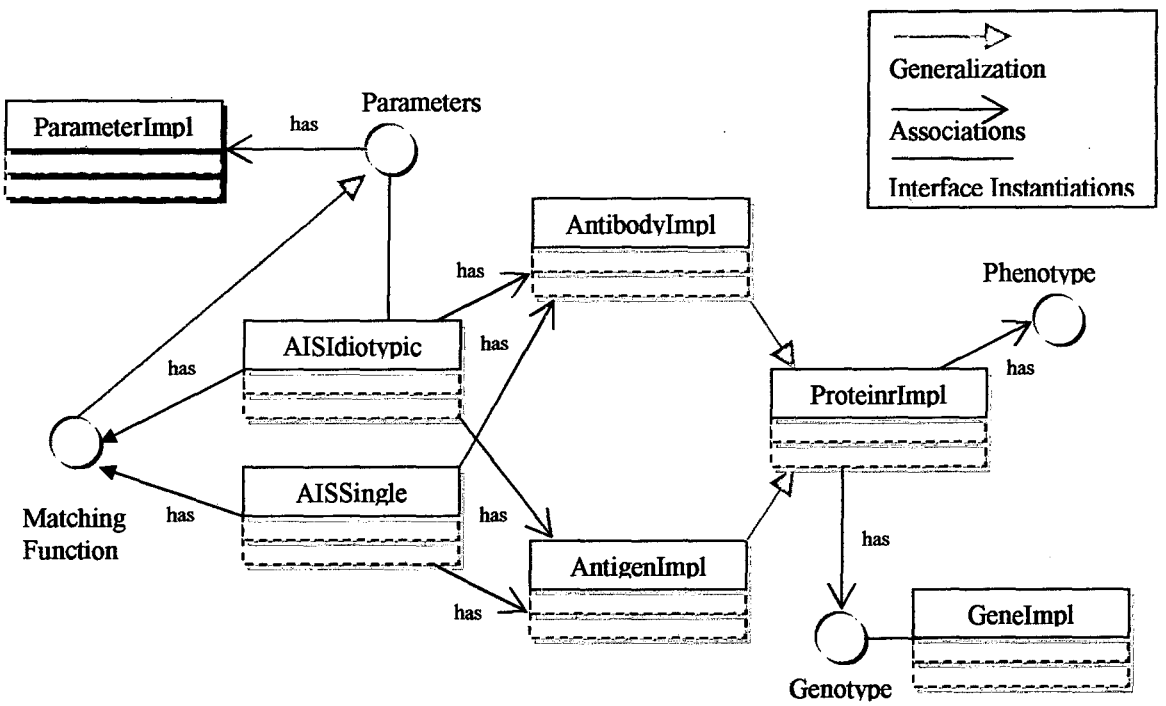


Figure 6: Class Diagram showing AIS *common* classes.

There are two implementations of the AIS in the common package- *AISIdiotypic* and *AISSingle*. The implementations of *Antibody, Antigen, Protein* and *Gene* are relatively straightforward interpretations of the model. However the *Genotype* and *Phenotype* classes cannot really be interpreted generically: they are left for the application to define in a suitable way. The matching function is dependent on genotype/phenotype encoding and so it also is problem dependent. The *Parameters* interface does not actually need a special implementing class; rather it specifies a contract

that must be fulfilled by *AIS* and *MatchingFunction* classes. The *Parameter* interface on the other hand is relatively straightforward to implement.

The author of the library has also implemented a recommender system application using the architecture described above [4, 5]. The system is a collaborative filtering application where the genotype encodes a user profile. We have developed our system on similar lines. Figure 7 shows the architecture (class diagram) of our implementation of a movie recommender system based on FAIS.

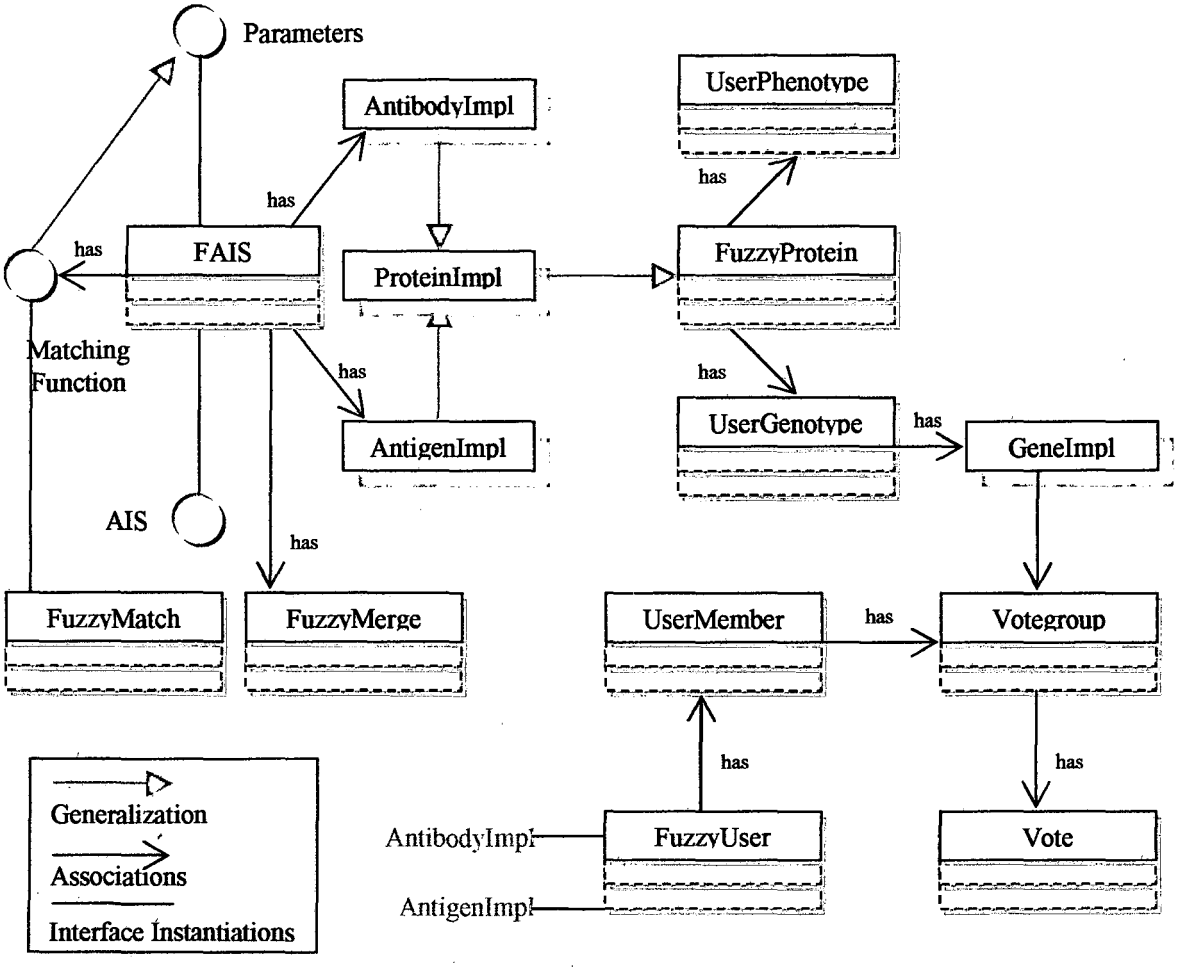


Figure 7: Class Diagram showing FAIS classes

In the class diagram, the already mentioned classes have been shadowed. *FAIS* is the main class that implements the *AIS* interface and contains the *FAIR* algorithm. The class *FuzzyMatch* implements the interface *MatchingFunction* and contains the method for calculating fuzzy affinity-score between two antibodies or between an antibody and an antigen. *FuzzyMerge* incorporates the logic of the sub-procedure FuzzyMerge discussed in Chapter 3.

FAIS is composed of fuzzy antibodies and antigens. However we use the *common* implementations of the antibodies and antigens since the 'fuzziness' is incorporated into the class implementing the *protein* and each antibody/ antigen has an associated protein. The class *FuzzyProtein* embodies the user profile in the form of user genotype/ phenotype. A fuzzy protein has a scale value that determines the radius of influence of a fuzzy antibody and a stimulation level that governs the immune system dynamics (for instance, change in antibody concentration). User profiles consist of voting records in the form {movieID, category, and vote}. Each gene encodes a *voting group* (which is a record of all *votes* in a particular category/genre). The *user genotype* is thus a series of such votes. The library implementations of the classes VoteGroup, Vote and UserGenotype/Phenotype have been used in the system.

The class UserMember embodies the demographic information of each individual user. A *FuzzyUser* is a class that depicts the fuzzy- profile i.e., a fuzzy set over the domain of users. It consists of a list of *user-members* of this fuzzy set along with their membership values in the set. FuzzyUser also comprises the aggregated profile / preference information for the members of this set. In this implementation, while merging two fuzzy users, we take the union of the two sets. It can be any other aggregation operator as discussed in Chapter 3.

Below we state the parameters employed (sub-section 4.2.1) and the experimental results obtained (sub-section 4.2.2).

4.2.1 Parameters Employed and Performance Measures Used

The table in figure 8 describes the parameters employed in the system and their values.

Parameter	**Value**
Initial Concentration of fuzzy antibodies	10
Initial Number of Antibodies in the System	10
Initial Scale Values for fuzzy antibodies.	0.45
Concentration-update constant (Formula 3.13)	1.0
Minimum allowed concentration per antibody	0
Maximum allowed concentration per antibody	500
Theta	Experimental Variable

Figure 8: Parameters Employed in the System

The performance of the system was judged based on the following measures

1. Mean Absolute Error (Formula)
2. Neighborhood Size –No. of (fuzzy) users used to generate recommendations

4.2.2 Results Obtained

Theta was chosen as the experimental variable for the system. It is the (fuzzy) affinity threshold value; fuzzy antibodies having affinity more than Theta are merged, thereby reducing the size of the fuzzy-antibody repertoire in the system. The effect of varying this value on the two performance measures mentioned above was studied in the experiments.

Figure 9 depicts the effect of varying Theta value on the prediction accuracy of the system, measured in terms of mean absolute error. We find that that decreasing the Theta value leads to a slight increase in the mean absolute error. Six Theta values in the range [0.95, 1.00] were used for experimentation.

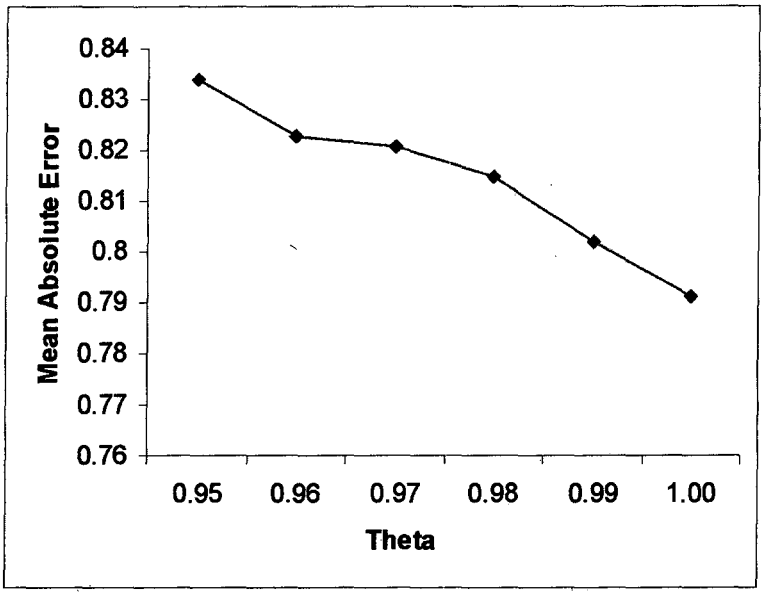


Figure 9. Effect of Theta value on the Mean Absolute Error

Figure 10 shows the effect of the variable Theta on the neighborhood size, i.e. no. of users used for generating good recommendations. We find that increasing the Theta value leads to increase in the neighborhood size for the same value of mean absolute error. Here also, the values of Theta were varied between 0.95 and 1.00 in steps of 0.01.

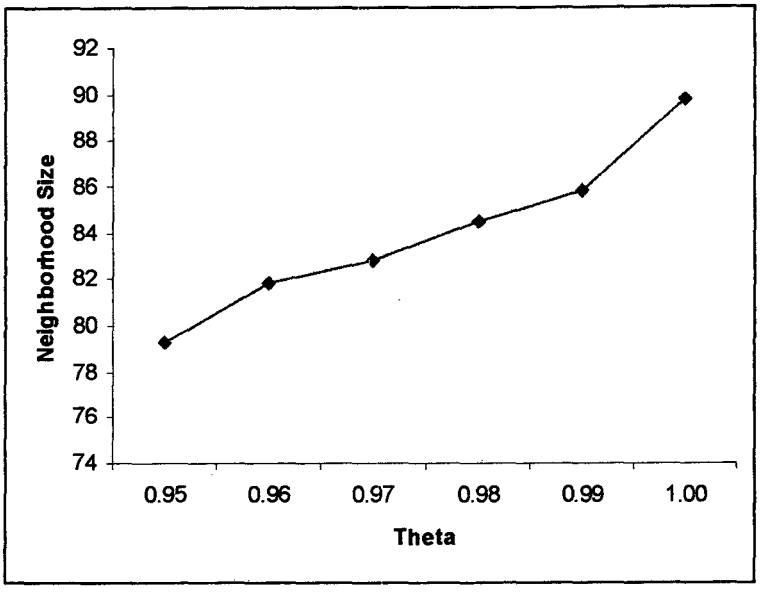


Figure 10. Effect of Theta value on the Neighborhood Size

Discussion on Results

The increase in MAE due to decrease in Theta value can be because of the following reason: A decrease in Theta value lowers the threshold and more antibodies will be merged during the FAIR iterations and this compromises some accuracy due to the loss of information due to aggregation. The no. of antibodies in the neighborhood decrease at a faster rate with decrease in Theta value. We find that there is a considerable decrease in the no. of (fuzzy) antibodies (90 to 79) in the system with the decrease in the theta value where as the MAE increases only slightly due to this (varies between .79 to .83). . However, MAE still remains within acceptable limits.

Chapter 5

Conclusions

The World Wide Web has become integral and indispensable part of our lives. But, the unorganized, unstructured, non-standard and dynamic nature of the Web along with its exponential growth in the recent past has posed many challenges for the computer scientists, information overload on users being one of them. Web Personalization and in particular, Recommender Systems have emerged as a popular research area because of these challenges and prolific growth in e-commerce websites. Recommender Systems are crucial to the success of e-businesses especially when the competitor is a click or two away. Various techniques drawn from Web Mining, Machine Learning, etc. have been used in developing recommender systems.

Artificial Immune Systems (AIS) is a novel computational paradigm which exhibits excellent learning capabilities and ease of adaptation to dynamic environments that characterize the World Wide Web. We find that AIS can be intuitively applied to Collaborative Filtering for generating recommendations and have proposed a general architectural design for Fuzzy-AIS based recommender system which can be adapted to any domain. An important concept in AIS for recommender system applications is the concept of diverse repertoire, i.e. a multitude of good solutions are searched for in parallel , not just the best one (as in GAs); this is because what we need is top-N recommendations and not just one.

The algorithm (FAIR) proposed in this work for collaborative filtering uses fuzzy-matching for antibody-antigen and antibody-antibody interactions where each antibody defines a fuzzy set in an L-dimensional space (L features used to describe user profile). We find that incorporation of fuzzy logic in the AIS dynamics can help us meet the constraints posed by computer resources (like memory, CPU time, etc.).

A movie recommender system was developed based on the above-mentioned design .The experimental results show that FuzzyMerge sub-procedure used in the FAIR helps in limiting the growth of the antibody repertoire while still maintaining diversity and quality in the FAIS, thereby generating good recommendations. The average accuracy is comparable to other approaches including the (simple) AIS approach. We also find that incorporating movie genre information in the collaborative filtering process helps in exploiting the logical grouping of movies and providing recommendations even when the neighborhood is sparse. The experiments conducted for this work show how FAIS is a suitable paradigm for collaborative filtering.

Future work can explore the incorporation of users' demographic information in collaborative filtering process which can provide benefits similar to those provided by including content information, i.e. some demographic patterns can be used to filter information, in case the neighborhood is sparse. Also, fuzzification of these demographic attributes will make the system more robust. Applicability of FAIR can be explored in other domains and it may need some adaptation since FAIR assumes a user-item ratings matrix as the input. If suitably adapted, FAIR can be used successfully for technical paper recommendation, job/job-candidate recommendation, etc. Various aggregation operators can be experimented with, for merging two fuzzy profiles. FAIS can also be used in combination with other techniques; for instance, GAs or neural networks can be used to automatically and continually evolve the optimal set of system parameters.

References

[1] Amazon, 2006, Recommendations, http://www.amazon.com/

[2] Cayzer, S. and Aickelin, U., 2002. "A Recommender System based on the Immune Network", *In Proceedings of CEC2002*, pp. 807-813, Honolulu, USA.

[3] Cayzer, S. and Aickelin, U., 2002. "On the Effects of Idiotypic Interactions for Recommendation Communities in Artificial Immune Systems", *In Proceedings of the 1st International Conference on Artificial Immune systems(ICARIS-2002)*, pp. 154-160, Canterbury, UK.

[4] Cayzer, S. Artificial Immune Systems library, http://www.hpl.hp.com/personal/Steve_Cayzer/ais_download.htm

[5] Chakrabarti, S., 2002."Mining the Web: Analysis of Hypertext and Semi Structured Data", Morgan Kaufmann.

[6] Dasgupta, D., 1998. "An Overview of Artificial Immune Systems and Their Applications", *Artificial Immune Systems and their Applications. Ed. Dipankar Dasgupta Pub. Springer-Verlag.* ISBN 3-540- 64390-7., pp. 3-21.

[7] De Castro, L.N. and Timmis, J., 2003. "Artificial Immune Systems as a Novel Soft Computing Paradigm", *Soft Computing Pub. Springer-Verlag* , pp. 526-544

[8] Eirinaki, M. and Vazirgiannis, M. , February,2003. "Web Mining for Web Personalization", *ACM Transactions on Internet Technology*, Vol 3, No. 1, pp. 1-27

[9] Frias-Martinez E., Magoulas G.D., Chen S., and Macredie R., Aug. 2004. "Recent Soft Computing Approaches to User Modeling in Adaptive Hypermedia". *In Paul De Bra, Wolfgang Nejdl (eds), Adaptive Hypermedia and adaptive web-based systems, Proceedings of 3rd Int Conf Adaptive Hypermedia-AH 2004,* Eindhoven, The Netherlands, *Lecture Notes in Computer Science,* vol. 3137, Springer, pp. 104-113.

[10] Herlocker, J., Konstan, J.A., Borchers, A., and Riedl, J. , 1999. "An algorithmic framework for performing collaborative filtering", *In Proceedings of SIGIR'99*, pp. 230-237.

[11] Hunt, J. and Cooke, D., 1995. "An adaptative, distributed learning system, based on immune system," *In Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, Los Alamitos, CA, pp. 2494–2499.

[12] Hunt, J. E. & Cooke, D. E., 1996. "Learning Using an Artificial Immune System", *Journal of Network and Computer Applications*, 19, pp. 189-212.

[13] Joachims, T., Freitag, D. and Mitchell, T. WebWatcher, 23-29 August 1997. "A Tour Guide for the World Wide Web", *In Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI '97)* Nagoya, Aichi, Japan, pp. 770-775.

[14] Konstan, J.A., Miller, B., Maltz, D., Herlocker, J., Gordon, L., and Riedl J, 1997. "GroupLens: Applying collaborative filtering to Usenet news", *Communications of the ACM*, 40(3), pp. 77-87.

[15] Kosala, R. and Blockeel, H., July 2000. "Web Mining Research: A survey", *SIGKDD Explorations*, vol2 (1), pp. 1-15.

[16] Li and Yamada, S., December 2004. "A Movie Recommender System Based on Inductive Learning", *The 2004 IEEE Conference on Cybernetics and Intelligent Systems (CIS-2004)*, pp.318-323, Singapore.

[17] Montaner, M., López, B., Lluís de la Rosa, J., 2003. "A Taxonomy of Recommender Agents on the Internet", *Artificial Intelligence Review* 19(4), pp 285-330.

[18] Nasraoui, O., Dasgupta, D. and Gonzalez, F., 2002. "An artificial immune system approach to robust data mining", *In Proceedings of Genetic and Evolutionary Computation Conference (GECCO)*, New York, NY, pp. 356–363.

[19] Nasraoui,O. F., Gonzalez, F. and Dasgupta, D., 2002. "The fuzzy artificial immune system: Motivations, basic concepts and application to clustering and web profiling", *In Proceedings of International Joint Conference on Fuzzy Systems. Part of the World Congress on Computational Intelligence*. Honolulu, HI.: IEEE, pp. 711–717.

[20] Nasraoui, O., Dasgupta, D., and Gonzalez, F., April 2002. "The Promise and Challenges of Artificial Immune System Based Web Usage Mining: Preliminary Results," *in SIAM Workshop on Web Analytics*, Arlington, VA, pp. 29-39.

[21] Pal, S. K., Talwar, V., Mitra, P., September 2003. "Web Mining in Soft Computing Framework: Relevance, State of the Art and Future Directions", *IEEE Transactions on Neural Networks*, Vol. 13, No.5, pp. 1163-1177.

[22] Resnick, P. et al., 1994. "GroupLens: An Open Architecture for Collaborative Filtering of Netnews", *In Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, Chapel Hill, NC, pp.175- 186.

[23] Sarwar, B. M., Karypis, G., Konstan, J. A., and Riedl, J., 2000. "Analysis of Recommendation Algorithms for E-Commerce", *In Proceedings of the ACM EC'00 Conference. Minneapolis*, MN, pp. 158-167.

[24] Schafer, J.B., Konstan, J.A. and Riedl, J., 2001. "Ecommerce Recommendation Applications", *Data Mining and Knowledge Discovery* 5(1/2), pp.115-153.

[25] Schafer, J.B., Konstan, J.A., and Riedl, J., 1999."Recommender Systems in E-Commerce", *In ACM Conference on Electronic Commerce (EC-99)*, pp. 158-166.

[26] Shardanand, U. and Maes, P. , 1995. "Social information filtering: Algorithms for automating "word of mouth"", *In Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, pp. 210-217.

[27] Timmis, J, Neal, M and Hunt, J., 2000. "An Artificial Immune System for Data Analysis", *Biosystems*, 55(1/3), pp. 143-150

[28] Timmis, J and Neal, M., June 2001." A Resource Limited Artificial Immune System for Data Analysis", *Knowledge Based Systems*, 14(3-4), pp.121-130.

[29] Ujjin, S. and Bentley, P. J., 2003. "Using Evolution to Learn User Preferences". *An invited chapter in Tan, K.C., Lim, M.H, Yao, X. and Wang, L. (Eds) Recent Advances in Simulated Evolution and Learning. World Scientific series on Advances in Natural Computation*, pp. 20-35.

[30] Watkins, A and Timmis, J., September 2002. "Artificial Immune Recognition Systems (AIRS): Revisions and Refinements", *In Proceedings of the 1st International Conference on Artificial Immune Systems*, pp. 173-181, University of Kent at Canterbury.

[31] Yager, R.1988. On ordered weighted averaging aggregation operators in multi-criteria decision making. *IEEE Transactions On Systems, Man and Cybernetics.* Vol. 18. p.p. 183-190.