دانشکده مهندسی کامپیوتر

استخراج قوانین انجمنی با استفاده از سیستم ایمنی مصنوعی

دانشجو:

بنت‌الهدی حلمی

پایان‌نامه برای دریافت درجه کارشناسی ارشد
در رشته مهندسی کامپیوتر - گرایش هوش مصنوعی

استاد راهنما:

دکتر عادل ترکمان رحمانی

بهمن ماه ۱۳۸٦

تقدیم به:

پیشگاه **خداوندگار** منان

تشکر و قدردانی:

اول شکر خدا می‌کنم که نعمت‌هایی به من عطا کرد که باید شکرشان را به جا آورم.

دوم مادر و پدر را که چند سال اول زندگی در چشم خرد من خدا بودند و بعد از آن چونان به پای من ریختند که همیشه خدا در نظرم مجسم می‌شد.

سوم استاد را که حق پدری بر گردنم دارد و راهنماییهایش هدایت‌گر راه است.

چهارم دوست را که استاد من بود و همراه و راهنمایم.

نمی‌توان به کمال قدردانی چنان کسانی را کرد، نه با کلمات و نه حتی با بهترین کارها.

هنگام قدردانی‌ای چنان سخیف، خجلت و شرمندگی است که تمام وجودم را فرا می‌گیرد ولی بپذیرید از من که تمام سرمایه‌ی من همین است و هر چه از پی این می‌آید.

.....

چکیده

با گسترش وب و ازدیاد اطلاعات موجود در آن، وجود ابزارهای مناسب برای دسته‌بندی این داده‌ها، عرضه‌ی داده‌ها به کاربران بر اساس علایق و نیاز آن‌ها و تغییر نوع عرضه‌ی اطلاعات با توجه به ذائقه‌ی کاربران ضروری به نظر می‌رسد. هدف استخراج اطلاعات از داده‌های دسترسی به وب نیز آسان‌سازی استفاده کاربران از وب، دسترسی سریع و آسان آن‌ها به اطلاعات و کمک به طراحان و عرضه‌کنندگان اطلاعات است تا با کمترین هزینه، بهترین خدمات را به کاربران ارائه و بیشترین سود را نصیب خود کنند. وب دارای ویژگی‌هاییست که وجود ویژگی‌های خاصی را برای الگوریتم‌های مرتبط با آن می‌طلبد. واضح است که مدل‌هایی که برای کاوش اطلاعات از وب طراحی می‌شوند باید دارای قابلیت تطابق‌پذیری با زمان ، مقاومت و کارایی در مقیاس‌های بزرگ باشند. اما سیستم ایمنی طبیعی که مدلی جدید است که بر اساس الهام از طبیعت به‌وجود آمده دارای ویژگی‌هاییست که برای کاوش وب بسیار مناسب به نظر می‌رسد. مهمترین ویژگی این سیستم طبیعی، ذات پویای آن است که بسیار مشابه ذات پویای کاوش اطلاعات وب است. علاوه بر این ویژگی، انگیزه‌های زیاد دیگری نیز برای استفاده از سیستم ایمنی در این کاربرد وجود دارد، این انگیزه‌ها عبارتند از تشخیص، تنوع، حافظه، خود تنظیمی و یادگیری که در سیستم ایمنی وجود دارد.

در این پروژه به طراحی، پیاده‌سازی و ارزیابی الگوریتمی بر اساس سیستم ایمنی مصنوعی خواهیم پرداخت که به منظور استخراج اطلاعات از داده‌های دسترسی به وب طراحی شده است. در این الگوریتم از فرآیندهایی از جمله شبکه‌ی ایمنی و تئوری خطر برای استخراج مجموعه آیتم‌هایی (URL) که مکرراً با هم در مجموعه‌ی داده‌های دسترسی به وب ظاهر می‌شوند، استفاده می‌شود. نتایج حاصل از الگوریتم ارائه شده، نشان دهنده‌ی درستی پیش‌بینی‌های انجام شده در مورد تناسب انتخاب AIS به عنوان الگویی برای حل مسئله‌ی پیدا کردن مجموعه آیتم‌های مکرر در داده‌های دسترسی به وب، است.

فهرست مطالب

عنوان **صفحه**

چکیده **ت**

فصل ۱: مقدمه **۲**

۱-۱ انگیزه و پیش‌زمینه‌ها ۲
۱-۲ چالش‌ها ۵
۱-۳ اهداف ۷
۱-۴ ساختار پروژه ۷

فصل ۲: استخراج اطلاعات از داده‌های دسترسی به وب **۱۰**

۲-۱ مقدمه ۱۰
۲-۲ داده کاوی و استخراج دانش ۱۰
۲-۳ وب کاوی ۱۲
۲-۳-۱ کاوش محتویات وب ۱۳
۲-۳-۲ کاوش ساختار وب ۱۳
۲-۳-۳ کاوش اطلاعات از داده‌های دسترسی به وب ۱۴
۲-۴ کاوش اطلاعات از داده‌های دسترسی به وب، مسائل مرتبط و مروری بر تحقیقات انجام شده ۱۴
۲-۴-۱ منابع داده ۱۵
۲-۴-۲ پیش پردازش ۱۸
۲-۴-۳ تکنیک‌ها ۲۵
۲-۴-۴ از تکنیک تا کاربرد ۳۴
۲-۵ جمع‌بندی ۳۷

فصل ۳: سیستم ایمنی مصنوعی **۳۹**

۳-۱ مقدمه ۳۹
۳-۲ سیستم ایمنی مصنوعی: الگویی الهام گرفته شده از بیولوژی ۳۹
۳-۳ ایمنی ۴۰

۳-۳-۱ ایمنی ذاتی ۴۱
۳-۳-۲ ایمنی اکتسابی ۴۱
۳-۳-۳ ایمنی اکتسابی و مهندسی ۴۹
۳-۴ سیستم ایمنی مصنوعی ۵۰
۳-۵ چهارچوب سیستم ایمنی مصنوعی ۵۱
۳-۵-۱ طرز نمایش اجزاء ۵۲
۳-۵-۲ میل پیوندی ۵۴
۳-۵-۳ پردازش‌ها ۵۶
۳-۶ سیستم ایمنی مصنوعی برای داده‌کاوی و کاربردهای دیگر..... ۶۴
۳-۶-۱ داده‌کاوی با سیستم ایمنی مصنوعی ۶۵
۳-۶-۱-۳ راه‌حل‌های تئوری خطر ۶۹
۳-۶-۲ سیستم‌های یادگیری پیوسته مبتنی بر سیستم‌های ایمنی مصنوعی دیگر..... ۷۱
۳-۶-۳ وب و داده‌کاوی با سیستم ایمنی مصنوعی..... ۷۱
۳-۷ جمع بندی ۷۳

فصل ۴: الگوریتم پیشنهادی برای استخراج مجموعه آیتم‌های مکرر از داده‌های دسترسی به وب ۷۶

۴-۱ مقدمه ۷۶
۴-۲ چرا سیستم ایمنی؟ ۷۷
۴-۳ سیستم ایمنی مصنوعی برای استخراج اطلاعات از داده‌های استفاده از وب ۷۹
۴-۴ الگوریتم پیشنهادی ۸۲
۴-۴-۱ نکات کلی ۸۲
۴-۴-۲ فلوچارت الگوریتم پیشنهادی ۸۳
۴-۴-۳ اجزاء سیستم و طرز نمایش آن‌ها ۸۴
۴-۴-۴ معیارهای هم‌نواختی و جذابیت نشست..... ۸۵
۴-۴-۵ تابع پیوند ۹۰
۴-۴-۶ مراحل الگوریتم ۹۲
۴-۴-۷ شبه کد..... ۱۰۸
۴-۵ جمع‌بندی ۱۰۸

فصل ۵: تحلیل نتایج ۱۱۲

۵-۱ مقدمه ۱۱۲
۵-۲ معیارهای مناسب برای ارزیابی الگوریتم استخراج اطلاعات از داده‌های دسترسی به وب توسط تکنیک سیستم ایمنی مصنوعی ۱۱۳
۵-۲-۱ تعریف معیارهای مورد استفاده برای ارزیابی الگوریتم ۱۱۵
۵-۳ نتایج شبیه‌سازی برای استخراج مجموعه آیتم‌های مکرر از داده‌های دسترسی به وب

با یک عبور از داده‌ها ۱۱۹
۵-۴ ارزیابی سود استفاده از وزن آیتم، وزن نشست و تئوری خطر..... ۱۳۰
۵-۵ مقایسه‌ی ویژگی‌های الگوریتم با الگوریتم‌های دیگر..... ۱۳۴
۵-۶ جمع‌بندی ۱۳۵

فصل ۶: نتیجه‌گیری و پیشنهادات ۱۳۷

۶-۱ مقدمه ۱۳۷
۶-۲ ارزیابی الگوریتم پیشنهادی ۱۳۷
۶-۳ پیشنهادات برای کارهای آتی ۱۳۹
۶-۴ جمع‌بندی ۱۴۰

مراجع ۱۴۲

پیوست الف- کدهای الگوریتم پیشنهادی به زبان C++ ۱۵۷

فهرست اشکال

عنوان **صفحه**

شکل ۲-۱: نمایش اساس اپریوری. اگر $\{C,D,E\}$ مکرر باشد، همه‌ی زیرمجموعه‌های این مجموعه‌ی مکرر نیز مکرر خواهد بود ۲۶
شکل ۲-۲: شبه‌کد الگوریتم اپریوری ۲۷
شکل ۳-۱: پاسخهای اولیه و ثانویه‌ی ایمنی. آنتی‌ژن ناآشنا در زمان t1 پاسخ X2 را تولید می‌کند ولی به تاخیر بین t1 و X1 توجه کنید. همان آنتی‌ژن که در زمان T2 وارد شده است تقریبا فورا در Y1 پاسخ ایجاد شده است و پاسخ Y2 از پاسخ X2 قویتر است. ۳۹
شکل ۳-۲: دیاگرام سلول B. آنتی‌بادی‌های زیادی در سطح سلول B دیده می‌شود..... ۴۰
شکل ۳-۳: مولکول آنتی‌بادی و ژنومش. (a) منطقه متغیر (منطقه V) مسئول تشخیص آنتی‌ژن و منطقه ثابت (منطقهC) مسئول اعمال متعددی مانند تثبیت مکمل. (b) پروسه بازآرایی که منجر به شکل‌گیری منطقه متغیر زنجیره سنگین مولکول آنتی‌بادی می‌شود: تکه‌های ژن (دقیقاً یکی از هر کتابخانه‌ی ژنی) به ترتیب به هم الحاق می‌شوند. سپس محصول نهایی به مولکول آنتی‌بادی کارا ترجمه می‌شود. V، D، J ، کتابخانه‌های منحصربفردی هستند که در تولید پاسخ ایمنی شرکت می‌کنند..... ۴۱
شکل ۳-۴: آناتومی یک آنتی‌بادی بر اساس تئوری شبکه‌ی ایمنی جرن. در این شکل آنتی‌بادیB با اتصال از طریق پاراتوپش به ایدیوتوپ آنتی‌بادیA تحریک می‌شود. آنتی‌بادی A با این عمل سرکوب (تحریک منفی) نیز می‌شود. ۴۳
شکل ۳-۵: مدل تئوری خطر ۴۵
شکل ۳-۶: قدرت اتصال بین لنفوسیت (L) و آنتی‌ژن ((Ag..... ۴۷
شکل ۳-۷: شباهت بین پذیرنده‌ی سلول B با بردار ویژگی سلول ایمنی مصنوعی. بردار ویژگی سلول‌های مصنوعی، برای مثال می‌توانند بولین باشند و نشان دهنده‌ی وجود یا عدم وجود یک کلمه در یک سند ۴۹
شکل ۳-۸: درون فضای تجسمی S، فضایV وجود دارد که در آن آنتی‌بادی (•) و آنتی‌ژن (×) قرار گرفته‌اند. فرض بر اینست که یک آنتی‌بادی همه‌ی آنتی‌ژن‌هایی که مکمل آن‌هاست و درون فضای قرار دارند را می تواند شناسایی کند ۵۰
شکل ۳-۹: منطقه‌ی تشخیص و آستانه‌ی فعالیت ضربدری. (A) نشان‌دهنده‌ی لنفوسیت (L) در پیوند مستحکمی با آنتی‌ژن (Ag1) است. آنتی‌ژن (Ag2) شباهت کمتری با لنفوسیت (L) دارد. ۵۱

شکل ۳-۱۰: فرآیند ایجاد یک مولکول آنتی‌بادی از ترکیب قطعات ژنی کتابخانه‌های ژنی. یک جزء از هر کتابخانه انتخاب و با قطعات دیگر الحاق شده و رشته‌ی صفات را که نماینده پذیرنده ایمنی است، می‌سازد..... ۵۴
شکل ۳-۱۱: شبه‌کد الگوریتم شبکه‌ی ایمنی مصنوعی عمومی..... ۵۷
شکل ۳-۱۲: الگوریتم CLONALG..... ۶۰
شکل ۳-۱۳: جریان سلول‌ها از لایه‌های مختلف در الگوریتم MARIA..... ۶۵
شکل ۳-۱۴: شبه‌کد الگوریتم AISEC..... ۶۶
شکل۳-۱۵: الگوریتم WUM ارائه شده در [۱۴۳]..... ۶۹
شکل ۴-۱: فلوچارت الگوریتم پیشنهادی ۸۱
شکل ۴-۲: نمایش اعمال جهش هدایت‌شده در الگوریتم AISWUM (الف)..... ۹۷
شکل ۴-۲: نمایش اعمال جهش هدایت‌شده در الگوریتم AISWUM (ب) ۹۷
شکل ۴-۲: نمایش اعمال جهش هدایت‌شده در الگوریتم AISWUM (ج) ۹۸
شکل ۴-۲: نمایش اعمال جهش هدایت‌شده در الگوریتم AISWUM (د)..... ۹۸
شکل ۴-۲: نمایش اعمال جهش هدایت‌شده در الگوریتم AISWUM (ذ)..... ۹۹
شکل ۴-۲: نمایش اعمال جهش هدایت‌شده در الگوریتم AISWUM (ط) ۹۹
شکل ۴-۳: شبه کد الگوریتم AISWUM ۱۰۶
شکل ۵-۱: نمونه‌ای از مسیرهای ذخیره شده در سرور وب ابزارآلات موسیقی ۱۲۰
شکل ۵-۲: نمونه‌ای از مسیرهای ذخیره شده در سرور وب دانشگاه ساسکاچوآن..... ۱۲۱
شکل ۵-۳: توزیع آنتی‌بادی‌های که دقت و شمول آن‌ها در مقایسه با مجموعه آیتم‌های مکرر پایه از ۰.۴ بیشتر است $S_{PRC,CVG}(t,c)$ ۱۲۵
شکل ۵-۴: توزیع داده‌های ورودی که دقت و شمول آن‌ها در مقایسه با مجموعه آیتم‌های مکرر پایه از ۰.۴ بیشتر است $S'_{PRC,CVG}(t,c,0)$ ۱۲۵
شکل ۵-۵: الف)توزیع داده‌های ورودی با دقت بالاتر از ۰.۴ نسبت به مجموعه آیتم‌های پایه ۱۲۶
شکل ۵-۵: ب)توزیع آنتی‌بادی‌های با دقت بالاتر از ۰.۴ نسبت به مجموعه آیتم‌های پایه۱۲۶
شکل ۵-۶: الف)توزیع داده‌های ورودی با شمول بالاتر از ۰.۴ نسبت به مجموعه آیتم‌های پایه ۱۲۶
شکل ۵-۶: ب)توزیع آنتی‌بادی‌های با شمول بالاتر از ۰.۴ نسبت به مجموعه آیتم‌ها پایه ۱۲۶
شکل ۵-۷: توزیع آنتی‌بادی‌های که دقت و شمول آن‌ها در مقایسه با مجموعه آیتم‌های مکرر پایه از ۰.۴ بیشتر است $S_{PRC,CVG}(t,c)$. ۱۲۷
شکل ۵-۸: توزیع داده‌های ورودی که دقت و شمول آن‌ها در مقایسه با مجموعه آیتم‌های مکرر پایه از ۰.۴ بیشتر است $S'_{PRC,CVG}(t,c,0)$ ۱۲۷
شکل ۵-۹: الف)توزیع داده‌های ورودی با دقت بالاتر از ۰.۴ نسبت به مجموعه آیتم‌های پایه ۱۲۸
شکل ۵-۹: ب)توزیع آنتی‌بادی‌های با دقت بالاتر از ۰.۴ نسبت به مجموعه آیتم‌ها پایه . ۱۲۸

شکل ۵-۱۰: الف)توزیع داده‌های ورودی با شمول بالاتر از ۰.۴ نسبت به مجموعه آیتم‌های پایه ۱۲۸
شکل ۵-۱۰: ب)توزیع آنتی‌بادی‌های با شمول بالاتر از ۰.۴ نسبت به مجموعه آیتم‌های پایه ۱۲۸
شکل ۵-۱۱: $P(\Delta t)$ برای مجموعه داده‌ی اول ۱۲۹
شکل ۵-۱۲: $C(\Delta t)$ برای مجموعه داده‌ی اول ۱۲۹
شکل ۵-۱۳: $P(\Delta t)$ برای مجموعه داده‌ی دوم ۱۲۹
شکل ۵-۱۴: $C(\Delta t)$ برای مجموعه داده‌ی دوم ۱۲۹

فهرست جداول

جدول۲-۱. WUM، ارتباط بین کاربردها، تکنیک‌ها و منابع داده ۳۴
جدول۵-۱. برخی از پروفایل‌های به‌دست آمده توسط الگوریتم Scalable K-Means برای مجموعه داده‌ی ابزار آلات موسیقی ۱۲۲
جدول۵-۲. میزان رضایت‌مندی کاربران از صفحات پیشنهادشده به آن‌ها ۱۲۴
جدول۵-۳. برخی از پروفایل‌های به‌دست آمده توسط الگوریتم پیشنهادی برای مجموعه داده‌ی ابزار آلات موسیقی ۱۳۱
جدول۵-۴. مقایسه‌ی ویژگی‌های الگوریتم‌های متفاوت ۱۳۵

فهرست علائم اختصاری

علامت اختصاری	عبارت کامل مربوطه
WUM	Web Usage Mining
AIS	Artificial Immune System
AISWUM	Artificial Immune System for Web Usage Mining
URL	Uniform Resource Locator
GUI	Graphic User Interface
WWW	World Wide Web
HTTP	HyperText Transfer Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
ISP	Internet Service Provider
HPG	Hypertext Probability Garph
EA	Evolutionary Algorithm
TCR	T-Cell Receptor
AIN	Artificial Immune Network
AINE	Artificial Immune NEtwork with limited resources
ARB	Artificial Recognition Ball
TSP	Travelling Salesman Problem
AIRS	Artificial Immune Recognition System
AISEC	Artificial Immune System for Email Classification
AISIID	Artificial Immune System for Interesting Information Discovery
SOSDM	Self Organize Sparse Distributed Memory
HTML	HyperText Markup Language
UCI	University of California, Irvine
IZ	Influence Zone

۱

مقدمه

انگیزه و پیش‌زمینه‌ها

چالش‌ها

اهداف

ساختار پروژه

سر ارادت ما و آستان حضرت دوست — که هر چه بر سر ما می‌رود ارادت اوست

نظیر دوست ندیدم اگر چه از مه و مهر — نهادم آینه‌ها در مقابل رخ دوست

-- حافظ

فصل ۱: مقدمه

۱-۱ انگیزه و پیش‌زمینه‌ها

با گسترش وب و ازدیاد اطلاعات موجود در آن، وجود ابزارهای مناسب برای دسته‌بندی این داده‌ها، عرضه‌ی داده‌ها به کاربران بر اساس علایق و نیاز آن‌ها و تغییر نوع عرضه‌ی اطلاعات با توجه به ذائقه‌ی کاربران ضروری به نظر می‌رسد. هدف استخراج اطلاعات از داده‌های دسترسی به وب نیز آسان‌سازی استفاده کاربران از وب، دسترسی سریع و آسان آن‌ها به اطلاعات و کمک به طراحان و عرضه‌کنندگان اطلاعات است تا با کمترین هزینه، بهترین خدمات را به کاربران ارائه و بیشترین سود را نصیب خود کنند. هیچ کدام از این اهداف قابل بهره‌برداری نخواهند بود مگر به واسطه‌ی یک سیستم استخراج اطلاعات اتوماتیک.

برای استخراج اطلاعات از داده‌های دسترسی به وب از روش گوناگونی استفاده می‌شود که مهم‌ترین آن کشف قوانین ارتباطی است و مهم‌ترین و چالش‌برانگیزترین قسمت کشف قوانین ارتباطی، کشف مجموعه آیتم‌های مکرر است. مهمترین روش کشف مجموعه آیتم‌های مکرر، اپریوری است[۱] و بقیه‌ی متدهای کشف مجموعه آیتم‌های مکرر نیز بر اساس اپریوری طراحی شده‌اند[۱]. خصوصیات الگوریتم اپریوری به شرح زیر است:

اپریوری یک الگوریتم سطح-مبنا[۲] است که مجموعه آیتم‌های مکرر را سطح به سطح تولید می‌کند. یعنی در سطح اول مجموعه آیتم‌های سایز یک و به ترتیب مجموعه آیتم‌های با سایزهای بزرگ‌تر.

در هر سطح، مجموعه آیتم‌های مناسب جواب با استفاده از مجموعه آیتم‌های مکرر استخراج شده در سطوح قبلی تولید می‌شوند.

---

[۱] Aprioru

[۲] Level-wise

در هر سطح، تراکنش یکبار پویش می‌شود تا میزان معیار پشتیبانی[۱] هر مجموعه آیتم مناسب جواب تعیین شود.

همان‌طور که از خصوصیات اپریوری مشخص است، این الگوریتم الگوریتمی زمان‌بر و ایستا است و برای مجموعه داده‌های بزرگ غیرکارآمد می‌باشد. در کاربرد کاوش اطلاعات از داده‌های دسترسی به وب (Web Usage Mining)، تراکنش‌ها داده‌های دسترسی به وب هستند. وب دارای ویژگی‌هاییست که وجود ویژگی‌های خاصی را برای الگوریتم‌های مرتبط با آن می‌طلبد. واضح است که مدل‌هایی که برای کاوش اطلاعات از وب طراحی می‌شوند باید دارای قابلیت تطابق‌پذیری با زمان[۲] ، مقاومت[۳] و کارایی در مقیاس‌های بزرگ[۴] باشند [۲]. به دو دلیل باید تطابق‌پذیر باشند، اول اینکه ذائقه‌ی کاربران و نوع استفاده‌ی آن‌ها از اطلاعات، بسیار پویا و متغیر است و دوم اینکه اطلاعات موجود در وب، پیوسته در حال تغییر است. باید مقاوم باشند، زیرا داده‌های موجود استفاده‌ی کاربران از وب دارای اطلاعات نادرست و نویز بسیار است و باید کارآمد در مقیاس‌های بزرگ باشد، زیرا حجم کاربران وب بسیار بالاست و بنابراین روزانه حجم زیادی از داده‌های دسترسی به وب تولید می‌شود. با توجه به مشخصاتی که برای کار با داده‌هایی که از وب تولید می‌شوند ذکر شد، کمبود الگوریتمی که دارای ذاتی پویا و کارآمد برای داده‌های حجیم و خلوت مانند داده‌های دسترسی به وب باشد، برای به‌دست آوردن مجموعه آیتم‌های مکرر در این نوع داده‌ها محسوس است.

اما سیستم ایمنی طبیعی، دارای ویژگی‌هاییست که برای کاوش وب بسیار مناسب به نظر می‌رسد. مهمترین ویژگی این سیستم طبیعی، ذات پویای آن است که بسیار مشابه ذات پویای کاوش اطلاعات

---

[۱] Support

[۲] Adaptable

[۳] Robust

[۴] Scalable

وب است. مدل‌هایی که بر اساس سیستم ایمنی مصنوعی طراحی شده‌اند دارای خصوصیاتی هستند که برای کاربرد کاوش اطلاعات وب مناسب است [۳].

پیاده‌کردن الگوریتم‌های کامپیوتری بر اساس اصول ایمنی و اجزای آن، AIS، یک الگوی یادگیری ماشین[۱] در حال رواج گسترده است. AIS با الهام از سیستم ایمنی پستانداران و با استفاده از اجزاء و فرآیندهای مورد استفاده در این سیستم‌ها، الگوریتم‌هایی متفاوت تولید می‌کند. این الگوریتم‌ها تعدادی از خصوصیات مناسب سیستم ایمنی طبیعی را در خود جمع کرده و برای حل مسائل در حوزه‌های مختلفی به‌کار می‌روند [۳].

انگیزه‌های زیادی برای استفاده از سیستم ایمنی به عنوان الهام برای کاوش اطلاعات دسترسی به وب وجود دارد، این انگیزه‌ها عبارتند از تشخیص، تنوع، حافظه، خود تنظیمی و یادگیری [۴]. به علت طبیعت سیستم ایمنی طبیعی، در الگوریتم‌هایی که با الهام از این سیستم به‌وجود می‌آیند، عمومیت[۲] حفظ می‌شود و اطلاعاتی که بسیار کم در مجموعه داده‌ها ظاهر می‌شوند حذف می‌شوند. این خصوصیات باعث می‌شود الگوریتمی که بر مبنای اصول AIS به‌وجود می‌آید با تغییر ذائقه و کاربرد کاربران تطابق پیدا کند.

در این پروژه به طراحی، پیاده‌سازی و ارزیابی الگوریتم AISی خواهیم پرداخت که به منظور استخراج اطلاعات از داده‌های دسترسی به وب طراحی شده است. در این الگوریتم از فرآیندهایی از جمله شبکه‌ی ایمنی و تئوری خطر برای استخراج مجموعه آیتم‌هایی (URL) که مکرراً با هم در مجموعه‌ی داده‌های دسترسی به وب ظاهر می‌شوند، استفاده می‌شود. تعداد مجموعه آیتم‌هایی که قرار است توسط الگوریتم استخراج شوند، نامشخص است بنابراین الگوریتم در دسته الگوریتم‌های غیرنظارتی دسته‌بندی می‌شود. می‌توان از این الگوریتم به عنوان یک مدل خوشه‌بندی برای مجموعه داده‌های

---

[۱] Machin Learning

[۲] Generalization

بزرگ نیز استفاده کرد، زیرا این الگوریتم قادر به خلاصه کردن داده‌های ورودی است و بنابراین می‌توان به وسیله‌ی این الگوریتم مراکز خوشه‌ها را در داده‌های حجیم پیدا کرد.

۱-۲ چالش‌ها

وب کاوی به خودی خود، یک کار چالش برانگیز است. کتاب بازیابی اطلاعات مدرن [۵] تعدادی از مسائل چالش برانگیز مرتبط با وب کاوی را برشمرده است. برای دفاع اولیه از ادعای مناسب بودن AIS به عنوان راه‌حل برای حل مسئله‌ی استخراج مجموعه آیتم‌های مکرر از داده‌های دسترسی به وب، چالش‌های عنوان شده در کتاب بالا به همراه راه‌حل رفع آن از طریق AIS، به‌صورت زیر خلاصه می‌شود:

داده‌های توزیع شده: داده‌های وب، بر روی کامپیوترهای بیشماری است. سیستم ایمنی نیز به‌طور طبیعی، توزیع شده است. درست مانند اینترنت این خاصیت توزیعی باعث عدم توقف و تنوع سیستم است.

درصد زیاد داده‌های ناپایدار[۱]**:** کامپیوتر و داده‌های جدید به آسانی به اینترنت اضافه یا از آن کم می‌شوند؛ سلول‌های ایمنی نیز به‌طور پیوسته می‌میرند و تکثیر می‌شوند. توانایی هر دوی آن‌ها در کنار آمدن با این شرایط نشان می‌دهد که هر دو تطابق‌پذیر، انعطاف‌پذیر[۲] و مقاوم هستند.

حجم بالا: سایز وب باورنکردنی و در حال ازدیاد است، بنابراین کاوش آن کار سختی است. سیستم ایمنی نیز از تعداد بیشماری سلول تشکیل شده است. از آن‌جا که وظیفه‌ی هر سلول خاص است و هر سلول به‌طور مستقل عمل می‌کند، سیستم کارایی بالایی دارد.

کیفیت داده: به دلیل آسانی انتشار اطلاعات در وب و آسانی دسترسی به آن، کیفیت اطلاعات موجود در آن زیر سوال می‌رود. خطا و غفلت در اطلاعات وب عادی است. سیستم ایمنی در برابر نویز

---

[۱] Volatile

[۲] Resilient

مقاوم است، به‌طوری‌که هماهنگی کامل برای ایجاد پاسخ لازم نیست. چنین تحمل خطایی برای الگوریتمی که به منظور کاوش در داده‌های کم کیفیت طراحی شده است ضروریست. ویژگی‌های یادگیری سیستم ایمنی برای کاوش داده‌های بی‌کیفیت بسیار ارزشمند است. سیستم ایمنی به سرعت خصوصیات جدید مهاجمان را از طریق جهش یاد می‌گیرد.

سه نکته چالش برانگیز که محور اصلی هستند: ۱) حجم داده‌ها بسیار زیاد است ۲) داده‌ها بسیار متغیرند ۳) سعی در طراحی الگوریتمی است که برای کاربردهای بلادرنگ مناسب باشد.

از آنجا که حجم داده‌ها بسیار زیاد است، باید الگوریتمی طراحی شود که در مقیاس بالا کارآمد باشد، بنابراین یکی از نکات مهم که هنگام طراحی مدنظر بوده است کارایی الگوریتم برای داده‌های زیاد است.

چالش دوم، سروکار داشتن با داده‌های پویا است. همان‌طور که در بخش پیش هم عنوان شد، پویایی وب نه تنها به علت پویایی محتویات و ساختار وب، بلکه به علت تغییر ذائقه و علایق کاربران و در نتیجه الگوهای راهبری[۱] آن‌هاست. تغییر الگوهای دسترسی بر اساس ساعت دسترسی در یک روز، روز دسترسی در یک هفته و بر اساس الگوهای فصلی و حوادث خارجی دیگر، قابل مشاهده است. بنابراین این نوع کاوش با کاوش داده‌های ایستا بسیار متفاوت است و باید متدی برای طراحی استفاده شود که بتواند با چنین تغییراتی سازگار شود.

و چالش سوم هم طراحی الگوریتم با در نظر گرفتن کاربردهای بلادرنگ WUM، از قبیل دادن پیشنهاد به کاربر[۲]، شخصی‌سازی[۳] و ... است. برای رسیدن به این هدف، باید از روش‌های تکراری، تصادفی و امثالهم در طراحی الگوریتم حذر کرد زیرا این کاربردها، همه کاربردهایی بلادرنگ هستند و باید دارای زمان اجرای کمی باشد.

---

[۱] Navigational patterns

[۲] Recommendation

[۳] Personalization

۱-۳ اهداف

هدف این پروژه، طراحی الگوریتمی است بر اساس AIS برای استخراج مجموعه آیتم‌های مکرر یا به عبارت دیگر برای کشف مسیرهای پر رفت و آمدی که کاربران هنگام حضور در یک وب‌سایت می‌پیمایند. همان‌طور که در دو بخش قبل عنوان شد نکات ویژه‌ای مدنظر این پروژه است از قبیل کارایی در مقیاس‌های بالا و مناسب بودن از نظر هزینه‌ی زمانی، که کارهای پیشین برای کشف مجموعه آیتم‌های مکرر در کاربرد WUM، موفق به ارائه‌ی الگوریتم عمومی با چنین ویژگی‌هایی نبوده‌اند.

۱-۴ ساختار پروژه

ساختار پروژه به صورت زیر است. فصل دو و سه به ارائه‌ی اطلاعات پیش‌زمینه‌ی لازم، تخصیص پیدا کرده است. مرور بر ادبیاتی که در این دو فصل آمده است برای دادن اطلاعات به خواننده برای درک راحت‌تر فصل‌های بعدی و آشنایی با کارهای انجام شده و پیدا کردن جایگاه این پروژه در ادبیات موضوع است. از آن‌جا که در این پروژه به دو موضوع تا حد زیادی بدون شباهت یعنی وب‌کاوی و سیستم ایمنی مصنوعی می‌پردازیم (که البته در این مرحله به هم نامرتبط هستند!)، هنگام مرور بر ادبیات برای حفظ وضوح، این دو موضوع را از هم جدا کرده و به هر کدام در بخش مجزایی پرداخته شده است. بنابراین فصل‌های دو و سه ارتباط مفهومی چندانی با هم نخواهند داشت.

فصل دو به جزئیات وب‌کاوی و بطور اخص، کاوش داده‌های دسترسی به وب، می‌پردازد. با استفاده از رویکرد بالا به پایین از وب‌کاوی شروع کرده و سپس بحث را اختصاصی کرده و به WUM خواهیم پرداخت.

فصل سه، به سیستم ایمنی مصنوعی اختصاص دارد و به همراه فصل دو اطلاعات پیش‌زمینه‌ی لازم را برای خواننده به‌وجود می‌آورد. در این فصل به معرفی AIS از منظر یک راه حل محاسباتی پرداخته می‌شود. مقالاتی که به آن‌ها ارجاع داده شده است، شامل مفاهیم ایمونولوژیکی هستند که اغلب به

عنوان منبع الهام برای طراحی AIS استفاده می‌شوند. بنابراین این فصل به ارائه‌ی روش‌هایی اختصاص می‌یابد که توسط مهندسانی به‌وجود آمده‌اند که از بیولوژی برای ایجاد یک مدل قابل انعطاف و مورد استفاده، بهره گرفته‌اند.

در فصل چهار به ارائه‌ی الگوریتم پیشنهادی برای استخراج مجموعه آیتم‌های مکرر خواهیم پرداخت که فصل اصلی این پروژه است.

فصل پنج به ارزیابی الگوریتم ارائه شده و تعریف معیارهای تخمین اختصاص دارد. در این فصل نتایج آزمایشاتی که برای ارزیابی سیستم انجام شده است، ارائه خواهد شد و رفتار الگوریتم تفسیر خواهد شد.

و نهایتاً فصل شش که به جمع‌بندی و خلاصه کردن نتایج فصل‌های پیشین اختصاص دارد، و دیدی کلی بر مسئله ایجاد می‌کند.

۲

استخراج اطلاعات از داده‌های دسترسی به وب

مقدمه

داده کاوی و استخراج دانش

وب کاوی

کاوش اطلاعات از داده‌های دسترسی به وب، مسائل مرتبط و مروری بر تحقیقات انجام شده

جمع بندی

"We are what we repeatedly do."
--Aristotle, c.384 BC-322 BC

فصل ۲: استخراج اطلاعات از داده‌های دسترسی به وب

۲-۱ مقدمه

در این فصل به تعریف مفاهیم در زمینه وب کاوی خواهیم پرداخت، بعد از ارائه‌ی دیدی کلی بر مسئله‌ی وب کاوی، به کاوش در اطلاعات دسترسی به وب (WUM) خواهیم پرداخت. روش‌های موجود برای این نوع کاوش در ادبیات این مسئله عنوان شده و کاربردهای آن ذکر می‌شود.

۲-۲ داده کاوی و استخراج دانش

داده کاوی به صورت زیر تعریف می‌شود:

داده کاوی تحلیل داده‌های قابل مشاهده برای کشف ارتباطات غیرمنتظره و خلاصه کردن داده‌ها به‌صورتی بدیع است که برای دارنده‌ی اطلاعات مفید و قابل درک باشد [۶].

کاوش اطلاعات، حجم عظیمی از داده‌های خام را به فرمی تغییر می‌دهد که انسان بتواند آن‌ها را به راحتی بفهمد و برای تصمیم‌گیری بتواند از این اطلاعات استفاده کند. در کل، داده کاوی به دو شاخه‌ی مرتبط مدل کردن و کشف الگو تقسیم می‌شود. مدل کردن خلاصه کردن ساختارهای عظیم داده است. مثال‌هایی از چنین مدل کردنی عبارتند از:

- تحلیل جزءبندی
- تحلیل رگرسیون
- تجزیه‌ی سری‌های زمانی

فرآیند کشف دانش، می‌تواند در قدم‌های زیر خلاصه شود [۷]:

- پاک‌سازی داده
- انتخاب و تغییر شکل داده
- کاوش داده

- ارزیابی
- ارائه‌ی دانش

دو قدم اول، فاز پیش‌پردازش و دو قدم آخر فاز پس‌پردازش هستند. قدم کاوش داده، قسمتی از فرآیند کشف دانش است که از داده‌های خام دانش استخراج می‌کند. در این قدم از استراتژی‌های یادگیری ماشین، آمار و ... برای حل مسائل داده‌کاوی استفاده می‌شود. [۸] سه صفت مطلوب را که دانش استخراج شده باید دارا باشد به‌صورت زیر برمی‌شمرد:

۱) دقیق، ۲) قابل درک و ۳) جالب

البته درجه‌ی اهمیت هر کدام از این صفات بسته به کاربرد و کاربر دارد. برای مثال اگر کاربرد دانش استخراج شده، استفاده از آن توسط کاربر و تصمیم‌گیری استراتژیک بر اساس این دانش به‌وسیله‌ی کاربر باشد و نه استفاده از این دانش توسط ماشین، قابل درک بودن دانش استخراج شده اهمیت بیشتری دارد. در این کاربردها اگر دانش استخراج شده قابل فهم نباشد، هدف تحقق پیدا نخواهد کرد. در الگوریتم پیشنهادی در این پروژه، خروجی الگوریتم که مجموعه آیتم‌های مکرر است، توسط ماشین یا به عبارت دیگر توسط یک الگوریتم دیگر مورد استفاده قرار می‌گیرد، بنابراین قابل درک بودن در این الگوریتم نسبت به دو مورد دیگر یعنی دقیق و جالب بودن دارای اهمیت کمتری است. ضمن این‌که نتایج برای کاربر نیز غیرقابل درک نخواهد بود، زیرا خروجی مجموعه‌ای از مجموعه آیتم‌هایی است که میزان رخدادشان در داده‌های ورودی زیاد بوده است و فهم این دانش استخراج شده برای کاربر آسان است.

در مسائل داده کاوی، هر چه حجم داده‌ها بیشتر می‌شود، میل بیشتری برای کشف الگوهای مخفی در داده‌ها برای به‌دست آوردن سود تجاری و ... به‌وجود می‌آید. در قدم اصلی داده کاوی ممکن است از چندین الگوریتم داده کاوی استفاده شود. کار اصلی الگوریتم داده کاوی با توجه به نوع مسئله‌ی کشف دانش تغییر می‌کند اما دو نوع اصلی الگوریتم‌های داده کاوی، کلاس‌بندی و خوشه‌بندی است.

خوشه‌بندی به فرآیند تقسیم‌بندی داده به یک یا چند گروه به‌طوری‌که فاصله‌ی بین خوشه‌ها حداکثر و فاصله‌ی درون خوشه‌ها حداقل باشد، اطلاق می‌شود. در این دسته‌بندی، الگوریتم پیشنهادی در این پروژه در دسته‌ی خوشه‌بندی قرار می‌گیرد، زیرا در نهایت تعداد نامشخصی از مسیرهای استفاده‌ی کاربران به‌دست می‌آید که نماینده‌ی کل داده‌ها هستند، هر کدام از این مجموعه آیتم‌های کشف شده نماینده‌ی یک مسیر تردد پر رفت و آمد در داده‌های دسترسی به وب هستند و به عنوان مراکز خوشه‌ها استفاده می‌شوند و هر کدام از داده‌های ورودی به یکی از این مراکز متعلق می‌شود.

۲-۳ وب کاوی

وب، اکنون بزرگترین انبار داده‌ی بشر است. ایجاد وب برمی‌گردد به سال ۱۹۹۰ که یک کارمند CERN اولین برنامه برای حرکت روی اتصالات دو طرفه بین مجموعه‌ای از اسناد را با رابط کاربر گرافیکی[۱] نوشت. نام این نرم‌افزار را وب گسترده‌ی جهانی[۲] گذاشتند. اگرچه عمل هدایت بین اسناد به این طریق، جدید نبود ولی اضافه شدن رابط کاربر کار جدیدی بود. این رابط کاربر جدید با ابداع دیگری در CERN به هم آمیخت. این ابداع HTTP[۳] بود. در سال ۱۹۹۳ حجم ترافیک HTTP ده برابر شد [۹] و بعد از آن وب گسترده‌ی جهانی از اسم یک نرم‌افزار تبدیل شد به یک وب واقعی که با ارتباطات بین اسناد ابرمتن به وجود آمده است.

همان‌طور که گفته شد وب منبع عظیمی از داده است. . این داده یا محتویات وب است که از طریق میلیون‌ها صفحه‌ی وب که در اختیار عموم است به‌دست می‌آید و یا اطلاعات اتصالات بین صفحات وب و یا داده‌های دسترسی روزانه کاربران به صفحات وب که در سرورهای وب در فایل‌های متنی به

---

[۱] GUI

[۲] World Wide Web

[۳] HyperText Transfer Protocol

نام لاگ[۱] ذخیره می‌شوند. بر اساس داده‌های مورد استفاده در کاوش وب، سه شاخه‌ی مجزا در وب‌کاوی به‌وجود آمده است. این سه شاخه، عبارتند از کاوش محتویات وب[۲]، کاوش ساختار وب[۳]، و کاوش اطلاعات از داده‌های دسترسی به وب[۴] که مورد تحقیق این پروژه است و به همین جهت موضوع اصلی این فصل می‌باشد.

۲-۳-۱ کاوش محتویات وب

کاوش محتویات وب، به حوزه‌ای از کاوش وب گفته می‌شود که با اطلاعات خام موجود در صفحات وب سروکار دارد. منبع این داده‌ها، داده‌های متنی موجود در صفحات وب (کلمات و تگ‌ها) است. کاربردهای معمول این نوع کاوش وب، سازماندهی وب بر اساس محتوا و درجه‌بندی[۵] صفحات وب بر اساس محتوا است.

کاوش محتویات وب، از جهاتی شبیه به داده کاوی و کاوش متن است. به داده‌کاوی شبیه است زیرا تکنیک‌های داده کاوی زیادی در مسئله‌ی کاوش محتویات وب استفاده می‌شوند؛ و به کاوش متن شبیه است زیرا اکثر محتویات وب، متن هستند. ولی از جهاتی هم با داده کاوی متفاوت است زیرا داده‌های وب اکثراً نیمه‌ساختاریافته هستند در حالیکه داده کاوی با داده‌های بدون ساختار سروکار دارد. با کاوش متن نیز از آن جهت متفاوت است که متن موجود در وب نیمه ساخت‌یافته است ولی کاوش متن با داده‌های متنی بدون ساختار سروکار دارد.

---

[۱] Log

[۲] Web Content Mining

[۳] Web Structure Mining

[۴] Web Usage Mining

[۵] Ranking

۲-۳-۲ کاوش ساختار وب

کاوش وب بر اساس ساختار یا کاوش ساختاری وب[۱]، بخشی از کاوش وب است که با ساختار وب سروکار دارد. منبع داده در این‌گونه کاوش، اطلاعات ساختاری است که در صفحات وب موجود است (مانند لینک‌های درون صفحات). کاربردهای معمول این نوع کاوش وب، سازماندهی صفحات وب بر اساس لینک‌های درون صفحات و یا درجه‌بندی صفحات وب از طریق ترکیب محتوا و ساختار[۱۰] و مهندسی وارونه[۲] مدل‌های سایت‌های وب است.

۲-۳-۳ کاوش اطلاعات از داده‌های دسترسی به وب

استخراج اطلاعات از داده‌های دسترسی به وب (WUM)، بخشی از وب کاوی است که با استخراج دانش مفید از فایل‌های لاگ[۳] ذخیره شده در سرورها سروکار دارد. منبع داده در این نوع کاوش، همان‌طور که گفته شد معمولاً لاگ‌های متنی هستند که هنگام دسترسی کاربرها به سرورهای وب جمع‌آوری می‌شوند و ممکن است این اطلاعات در فرمت‌های استاندارد Common Log Format [۱۱]، Extended Log Format [۱۲] و LogLM [۱۳] در سرورها ذخیره شوند. کاربردهای معمول این نوع کاوش وب، کاربردهای مبتنی بر تکنیک‌های مدل کردن کاربر، مانند شخصی‌سازی وب، وب سایت‌های تطابق‌پذیر[۴] و کاربردهای اطلاعاتی هستند.

از آنجا که WUM مبحث اصلی این رساله است در بخش بعدی و در قالب مجزایی به ارائه‌ی مسائل مرتبط با این نوع کاوش وب پرداخته می‌شود.

---

[۱] Web Structure Mining

[۲] Reverse Engineering

[۳] Log

[۴] Adaptive Web sites

۲-۴ کاوش اطلاعات از داده‌های دسترسی به وب، مسائل مرتبط و مروری بر تحقیقات انجام شده

در سال‌های اخیر، تحقیقات در حوزه‌ی کاوش وب و به‌خصوص WUM گسترش زیادی یافته است. از زمانی که اولین مقالات در این زمینه منتشر شد یعنی اواسط ۱۹۹۰ تا به حال ۴۰۰ مقاله در زمینه‌ی کاوش وب منتشر شده است. تقریبا ۱۵۰ مقاله از ۴۰۰ مقاله، قبل از ۲۰۰۱ به چاپ رسیده است و حدودا ٪۵۰ از این مقالات درباره‌ی WUM بوده است. اولین کارگاه که منحصراً در زمینه‌ی WUM بوده است، WebKDD، در سال ۱۹۹۹ برگزار شده است. از سال ۲۰۰۰، مقالات منتشر شده در مورد WUM بیش از ۱۵۰ عدد بوده است که نشان دهنده‌ی افزایش جذابیت این موضوع است.

در این بخش که ارتباط مستقیم با موضوع این پروژه دارد ابتدا در مورد انواع متفاوت داده‌هایی که در اثر استفاده‌ی کاربران از وب (مسیرهایی که کاربر طی ملاقات سایت می‌پیماید) جمع‌آوری می‌شود، بحث خواهد شد. سپس پیش‌پردازش‌های لازم بر روی لاگ‌های جمع‌آوری شده بیان می‌شود [۱۰]. لاگ‌های وب برای مقاصد مختلف فیلتر می‌شوند، مثلا برای مرتب کردن داده‌های غیرلازم (مثلا دسترسی از طریق Web Spider)، برای تعیین نشست‌های کاربران[۱] (به‌وسیله‌ی کوکی[۲]ها)، برای ذخیره‌ی داده‌ها در یک پایگاه داده‌ی رابطه‌ای یا برای فراهم کردن یک ساختار مناسب برای مراحل بعدی کاوش. در قسمت‌های دیگر این فصل، تکنیک‌های کاوش مورد بررسی قرار خواهد گرفت و در ادامه به کاربردهای کاوش اطلاعات از داده‌های دسترسی به وب پرداخته می‌شود. در پایان با استفاده از یک جدول، تکنیک‌های مورد استفاده برای کاربردهای خاص و منبع داده‌های مورد استفاده برای تعدادی از کارهای انجام شده که در ادبیات این موضوع ثبت شده‌اند ارائه خواهد شد.

---

[۱] User sessions

[۲] Cookie

۲-۴-۱ منابع داده

داده‌های مورد استفاده برای کاوش اطلاعات از داده‌های دسترسی به وب از سه منبع اصلی به‌وجود می‌آید، این سه منبع عبارتند از: ۱) سرورهای وب، ۲) سرورهای پراکسی، ۳) مشتریان وب

۲-۴-۱-۱ سرور

سرورهای وب، قطعا پربارترین و معمول‌ترین منبع داده هستند. سرورها می‌توانند حجم زیادی از اطلاعات را در فایل‌های لاگ خود، جمع‌آوری کنند. این لاگ‌ها معمولاً حاوی اطلاعات پایه برای مثال: نام، IP میزبان دور، تاریخ، زمان درخواست، خط درخواستی[۱] که مستقیماً از مشتری آمده و ... هستند. این اطلاعات اغلب در فرمت استاندارد نمایش داده می‌شوند. این فرمت‌ها Common Log Format [۱۱]، Extended Log Format [۱۲] و LogML [۱۳] هستند. در برخی موارد برای ذخیره‌ی لاگ‌ها به جای فایل متنی از پایگاه داده استفاده می‌شود، دلیل اینکار بهبود پرس‌وجو در فایل‌های لاگ حجیم است [۱۴] و [۱۵].

هنگام استفاده از اطلاعاتِ لاگ سرورهای وب، مسئله‌ی مهم تعیین نشست‌های کاربران یا چگونگی گروه‌بندی همه‌ی درخواست‌های صفحه‌ی کاربران (یا رشته کلیک[۲]) است تا به این‌صورت بتوان مسیری را که کاربران هنگام راهبری در وب سایت طی می‌کنند، تعیین کرد. این کار معمولا بسیار مشکل است و به نوع اطلاعات موجود در فایل‌های لاگ مربوط است. یکی از رویه‌های معمول، استفاده از کوکی‌ها برای ردیابی توالی درخواست صفحه‌های کاربر است. (برای مروری بر استانداردهای کوکی‌ها به [۱۶] مراجعه شود.)

اگر کوکی‌ها موجود نباشند، می‌توان به کمک الگوریتم‌های اکتشافی متعددی [۱۷]، نشست‌های کاربران را بطور نسبتاً دقیقی تشکیل داد. توجه کنید که حتی اگر کوکی‌ها هم موجود باشند باز هم

---

[۱] Request line

[۲] Click stream

امکان تعیین کاملاً دقیق مسیر راهبری کاربران به دلیل وجود دکمه برگشت[1] که در سرور قابل ردیابی نیست، وجود ندارد [۱۸]. به غیر از لاگ‌های وب، رفتار کاربران در سرور از طریق TCP/IP packet sniffer نیز قابل ردیابی است. حتی با این روش نیز، تعیین نشست کاربران به آسانی ممکن نیست ولی این روش سودهایی دارد [۱۹] که در زیر به آن‌ها اشاره می‌کنیم. با استفاده از TCP/IP packet sniffer ۱) داده در زمان واقعی[2] جمع‌آوری می‌شود. ۲) اطلاعات وب سرورهای مختلف می‌تواند به راحتی با هم در یک فایل لاگ ادغام شوند. ۳) استفاده از کلیدهای خاص (مانند کلید توقف) قابل تشخیص است و بنابراین اطلاعاتی که در فایل‌های لاگ وجود ندارد، در دسترس قرار می‌گیرند.

با وجود همه‌ی این مزیت‌ها، از TCP/IP packet sniffer به ندرت استفاده می‌شود.

همان‌طور که در [۲۰] آورده شده است احتمالاً بهترین روش برای ردیابی داده‌های دسترسی به وب، دسترسی مستقیم به لایه‌ی کاربردی سرور[3] است. همان‌طور که در [۲۱] آورده شده است، متاسفانه این‌کار همیشه ممکن نیست. اول اینکه مسائلی مرتبط با کپی‌رایت کاربردهای سرور[4] وجود دارد. مهم‌تر اینکه، با استفاده از این رویکرد کاربردهای کاوش اطلاعات از داده‌های استفاده از وب باید برای سرورهای خاص، نوشته می‌شد و باید نیازهای ردگیری خاصی را مدنظر قرار می‌دادند.

در این پروژه منبع داده، فایل‌های لاگی است که در وب سرورها ذخیره می‌شوند.

۲-۴-۱-۲ پراکسی

خیلی از فراهم کنندگان سرویس اینترنت ISP، به مشتریان، سرویس‌های سرور پراکسی می‌دهند و از طریق نگهداری در حافظه‌ی نهانی[5]، سرعت هدایتشان را بیشتر می‌کنند. از جهات زیادی، جمع‌آوری

---

[1] Back

[2] Real time

[3] Server application layer

[4] Server applications

[5] Caching

داده‌ی راهبری در سطح پراکسی در اصل مشابه جمع‌آوری داده در سطح سرور است. اختلاف اصلی این دو مورد اینست که سرورهای پراکسی، داده را از گروهی از کاربرها که به گروه کثیری از سرورهای وب دسترسی دارند جمع‌آوری می‌کند. حتی در این مورد هم، تشکیل نشست‌ها مشکل است و همه مسیرهای راهبری کاربرها قابل تعیین نیستند. به هر حال زمانی‌که بین سرور پراکسی و مشتری‌ها، حافظه‌ی نهانی وجود ندارد (نگهداری در حافظه‌ی نهانی)، تعیین نشست کاربران آسان‌تر است.

۲-۴-۱-۳ مشتری

داده‌ی استفاده از وب در طرف مشتری نیز با استفاده از جاوا اسکریپت، اپلت‌های جاوا یا حتی مرورگرهای تغییریافته [۲۲] قابل ردگیری است. این تکنیک‌ها، از مشکلات ناشی از تعیین نشست کاربران و مشکلات ناشی از نگهداری در حافظه‌ی نهانی (مانند استفاده از دکمه‌ی برگشت)، جلوگیری می‌کند. به‌علاوه با این تکنیک‌ها، اطلاعات جزئی نیز درباره‌ی رفتارهای حقیقی کاربران فراهم می‌شود [۲۰]. به هر حال این تکنیک‌ها، اتکای زیادی بر همکاری کاربر می‌کنند و بنابراین مشکلات زیادی درباره‌ی مسائل حریم شخصی[۱] به‌وجود می‌آید.

۲-۴-۲ پیش پردازش

پیش‌پردازش داده، نقش مهمی در کاربردهای WUM دارد. مرجع [۲۳] اظهار می‌کند که اگرچه تکنیک‌های پیش‌پردازش در WUM بطور گسترده‌ای استفاده شده‌اند، مقالات در این مورد بسیار محدود است. مرجع کامل موجود مربوط به پیش‌پردازش برای WUM برمی‌گردد به سال ۱۹۹۹ [۲۴].

پیش‌پردازش داده‌های لاگ معمولاً پیچیده و زمان‌بر است و چهار مرحله دارد.

---

[۱] Privacy

۱) تمییز کردن داده‌ها[۱]،

۲) تعیین و تشکیل نشست کاربران،

۳) بازیابی اطلاعات در مورد محتوا یا ساختار صفحات و

۴) فرمت داده‌ها.

۲-۴-۲-۱ تمییز کردن داده‌ها

این مرحله شامل حذف همه‌ی داده‌های بدون مصرف از فایل‌های لاگ وب است [۲۳] و [۲۵]. از جمله‌ی این داده‌های بدون فایده می‌توان به درخواست کاربران برای محتویات صفحات گرافیکی (مانند تصاویر gif و jpg) و فایل‌هایی که درون صفحات وب وارد شده‌اند[۲] یا حتی نشست‌های راهبری که توسط ربات‌ها یا عنکبوت‌های وب انجام شده‌اند، اشاره کرد.

درخواست محتویات گرافیکی و فایل‌ها به راحتی قابل حذف است، ولی الگوی راهبری روبات‌ها و عنکبوت‌های وب باید صریحاً مشخص شوند. این کار معمولاً با مراجعه به اسم میزبان دور یا با مراجعه به عامل کاربر[۳] یا با چک کردن دسترسی به فایل robot.txt، انجام می‌شود. به هر حال برخی از روبات‌ها حقیقتاً یک عامل کاربر اشتباه، در درخواست HTTP می‌فرستند. در این موارد، الگوریتم‌های اکتشافی مبتنی بر رفتار راهبری برای جداکردن نشست روبات‌ها از نشست کاربران واقعی به‌کار می‌روند [۲۶][۲۷].

۲-۴-۲-۲ تعیین و تشکیل نشست کاربران

این مرحله تشکیل شده است از:

- تعیین نشست کاربران متفاوت از اطلاعات ضعیف موجود در فایل‌های لاگ.

---

[۱] Data cleaning

[۲] Include

[۳] User agent

- تشکیل دوباره‌ی مسیر راهبری کاربرها درون نشست‌های تعیین شده.

پیچیدگی این مرحله، بسته به کیفیت و کمیت اطلاعات موجود در فایل لاگ، متفاوت است [۲۸]. بیشتر مشکلاتی که در این بخش ایجاد می‌شود بر می‌گردد به نگهداری در حافظه‌ی نهانی که در سرورهای پراکسی یا مرورگرها اتفاق می‌افتد.

نگهداری در حافظه‌ی نهانی، در پراکسی باعث می‌شود که یک آدرس IP (که متعلق به سرور پراکسی است)، برای کاربران مختلف یکی شود و بنابراین استفاده از آدرس IP به عنوان یک مشخصه‌ی کاربر غیرممکن می‌شود. این مشکل را می‌توان تا حدی با کوکی‌ها [۲۹] ، با بازنویسی URL [۳۰] یا با مجبور کردن کاربر برای وارد کردن یک نام کاربری و کلمه‌ی عبور[۱] هنگام ورود به وب سایت [۲۳]، حل کرد.

کوکی تکه‌ای از اطلاعات است که توسط سرور وب به یک مرورگر وب فرستاده می‌شود. این اطلاعات درون کامپیوتر کاربران در یک فایل متنی ذخیره می‌شوند. کوکی‌ها ممکن است اطلاعات زیادی در مورد کاربرها دربرداشته باشند، بین این اطلاعات مربوط به کوکی‌ها، اطلاعات تعیین‌کننده‌ی نشست[۲] است که برای ما مطلوب است. این اطلاعات ممکن است، هر زمان که یک کاربر درخواست یک صفحه‌ی وب کند، توسط سرور وب درخواست شود و در یک لاگ وب به همراه صفحه‌ی درخواست شده، ذخیره گردد.

شرایطی وجود دارد که کوکی‌ها کمکی نمی‌کنند. برای مثال، برخی از مرورگرها از کوکی‌ها پشتیبانی نمی‌کنند. برخی دیگر از مرورگرها، به کاربران اجازه‌ی غیرفعال کردن کوکی‌ها را می‌دهند. در چنین شرایطی می‌توان از بازنویسی و اضافه کردن ID نشست در URL، برای ردگیری نشست کاربران استفاده کرد. بازنویسی URL شامل پیدا کردن همه لینک‌هایی که در آینده دوباره در مرورگر نوشته می‌شوند و بازنویسی آن‌ها به همراه ID نشست، می‌شود. برای مثال، یک لینک مانند:

---

[۱] Log in

[۲] Session identifier

<a href = “/store/catalog”>

به‌صورت زیر بازنویسی می‌شود:

<a href=”/store/catalog;jsessionid=DA32242SSGE2”>

که در آن ID نشست نیز وارد شده است. بنابراین هر زمان که یک کاربر روی یک لینک درون صفحه کلیک کند، فرم بازنویسی شده URL به سرور فرستاده می‌شود و در لاگ وب ذخیره می‌شود.

نگهداری در حافظه‌ی نهانی که توسط مرورگر وب انجام می‌شود، مسئله پیچیده‌تریست. لاگ‌های سرورهای وب نمی‌توانند شامل اطلاعاتی راجع به استفاده از دکمه‌ی برگشت باشند. این باعث می‌شود مسیرهای راهبری ناسازگار با واقعیت تولید شود. به هر حال با استفاده از اطلاعات اضافی در مورد ساختار وب‌سایت، هنوز هم می‌شود یک مسیر سازگار با استفاده از الگوریتمهای اکتشافی، تشکیل داد. برای مثال همان‌طور که در [۲۴] گزارش شده است اگر یک درخواست صفحه ایجاد شود و این درخواست مستقیماً به درخواست‌های قبلی صفحه لینک نداشته باشد، لاگ ارجاع دهنده چک می‌شود که درخواست از کدام صفحه آمده است. اگر صفحه در تاریخچه‌ی اخیر کاربر وجود داشته باشد، می‌توان فرض کرد کاربر از دکمه‌ی برگشت استفاده کرده است و سپس بر اساس این فرضیه امکان تشکیل یک مسیر راهبری سازگار و کامل وجود دارد.

برای حل هر دو مشکل نگهداری در حافظه‌ی نهانی در پراکسی و وب، آی بی ام[۱]، درون نرم افزار سورف‌اِید[۲] [۳۱]، یک جاوا اسکریپت به نام وب‌باگ[۳] معرفی کرده است که باید در هر صفحه وب وارد شود. هر وقت که یک صفحه‌ی وب بارگذاری شود، وب‌باگ یک درخواست به سرور می‌فرستد و درخواست یک تصویر ۱*۱ می‌کند. درخواست با پارامترهایی که مشخص کننده‌ی صفحه‌ی وب است و شامل اسکریپت و یک پارامتر تصادفی عددی است، تشکیل می‌شود. کل درخواست نمی‌تواند نه

---

[۱] IBM

[۲] SurfAid

[۳] WebBug

به‌وسیله پراکسی و نه به‌وسیله‌ی مرورگر در حافظه‌ی نهانی نگهداری شود، بلکه توسط سرور وب درون فایل لاگ ذخیره می‌شود و مسئله‌ی نگهداری در حافظه‌ی نهانی حل می‌شود [۲۳] [۳۱].

از آنجا که پروتکل HTTP دارای جزئیات نیست ، تعیین زمانی که کاربر وب سایت را ترک می‌کند از روی پروتکل HTTP امکان‌پذیر نیست و بنابراین تعیین زمان انتهای نشست کار آسانی نیست. به این مشکل، مشکل تعیین نشست[۱] می‌گویند.

مرجع [۱۷]، سه مدل اکتشافی برای تعیین انتهای نشست‌ها معرفی و این مدل‌ها را با هم مقایسه کرده است. دو تا از آن‌ها بر اساس زمان بین درخواست صفحه‌ی کاربران و یکی بر اساس اطلاعات در مورد ارجاع‌دهنده است. مرجع [۳۲]، یک مدل اکتشافی وفقی برای تعیین زمان اتمام نشست[۲] ارائه می‌دهد. مرجع [۲۴]، یک تکنیک برای نتیجه‌گیری از روی آستانه‌ی زمان اتمام برای وب‌سایت‌های خاص، ارائه می‌دهد. بقیه‌ی نویسندگان بر اساس تجربه، آستانه‌های مختلفی برای الگوریتم‌های اکتشافی بر مبنای زمان ارائه کرده‌اند. آستانه‌ی معمول برای اتمام نشست، ۳۰ دقیقه است که در [۲۲] ارائه شده است.

در این پروژه از ساده‌ترین روش ایجاد نشست‌ها استفاده شده است، به این‌صورت که از آستانه‌ی ۳۰ دقیقه‌ای برای هر نشست استفاده شده است. همچنین داده‌های لاگ در شرایطی جمع‌آوری شده‌اند که نگهداری در حافظه‌ی نهانی غیرفعال بوده است، بنابراین مطمئناً هر درخواست صفحه در فایل لاگ ذخیره می‌شود.

۲-۴-۲-۲ بازیابی ساختار و محتوا

اکثریت کاربردهای WUM، از URLهای ملاقات شده به عنوان منبع داده‌ی اصلی برای کاوش استفاده می‌کنند. ولی URL، منبع داده‌ی ضعیفی است. برای مثال، URL حامل اطلاعاتی راجع به

---

[۱] Sessionization

[۲] Adaptive time out heuristic

محتویات صفحه نیست. [۲۴]، اولین کاری بود که از اطلاعات محتوا برای غنی کردن داده‌های لاگ ذخیره شده در وب سرور استفاده کرد. در این مقاله، یک مرحله طبقه‌بندی کردن جدید اضافه شده است که در آن صفحات وب بر اساس نوع محتوا کلاس‌بندی می‌شوند. این اطلاعات اضافی بعداً هنگام کاوش لاگ‌های وب استفاده می‌شوند. اگر یک کلاس‌بند مناسب از قبل مشخص نباشد، تکنیک‌های کاوش ساختار وب برای ایجاد یک کلاس‌بند مناسب مورد استفاده قرار می‌گیرد. مانند موتورهای جستجو، صفحات وب بر اساس حوزه‌ی معنایی که در آن قرار می‌گیرند، توسط تکنیک‌های کاوش محتوای وب، کلاس‌بندی می‌شوند. این اطلاعات کلاس‌بندی، برای غنی‌کردن اطلاعات درون لاگ‌ها به‌کار می‌روند. برای مثال [۳۳]، پیشنهاد می‌کند تا از وب معنایی برای WUM استفاده شود. صفحات وب به آنتولوژی‌ها نگاشت می‌شوند تا به مسیرهایی که اغلب مشاهده می‌شوند معنا داده شود. وقتی صفحه‌ی وبی به ما داده می‌شود، باید قادر به استخراج اشیاء[۱] به عنوان موجودیت‌های معنایی که در صفحه هستند، باشیم. این کار ممکن است شامل استخراج خودکار و کلاس‌بندی اشیاء با انواع مختلف به کلاس‌هایی بر مبنای آنتولوژی‌های دامنه[۲] باشند. آنتولوژی‌های در این حوزه باید از قبل تعیین شوند و یا به‌طور اتوماتیک از داده‌های آموزش یاد گرفته شوند [۳۴]. در صورت وجود این توانایی، می‌توان داده‌های تراکنش را با موجودیت‌های معنایی‌ای که کاربر هنگام ملاقات یک سایت به آن‌ها دسترسی پیدا می‌کند، ترکیب کرد. مرجع [۳۵] به عنوان یک راه دیگر در کنار مسیرهای راهبری معمول کاربرها، یک مسیر مبتنی بر مفهوم ارائه می‌دهد. مسیر مبتنی بر مفهوم یک ایده (عمومیت) سطح بالا از مسیر معمول است که در آن مفاهیم معمول با استفاده از اشتراک مسیرهای خام کاربر و معیارهای شباهت، استخراج می‌شود. مرجع [۳۶]، استفاده از پی‌گیری[۳] اطلاعات برای بهبود نتایج مدل کردن کاربر را پیشنهاد می‌کند. ایده‌ی پی‌گیری اطلاعات از کاوش محتوای وب و کاوش ساختار

---

[۱] Domain-level structured

[۲] Underlying domain ontologies

[۳] Scent

وب اخذ شده است. پی‌گیری اطلاعات [۳۶]، به این‌صورت تعریف می‌شود: دریافت ذهنی و غیرکامل از مقدار و ارزش منابع اطلاعاتی که از نزدیک‌ترین نشانه‌ها مانند لینک‌های وب یا آیکون‌هایی که نماینده‌ی منابع محتوا[۱] هستند، به‌دست می‌آید.

مرجع [۳۷] نتایجی تجربی ارائه می‌دهد که نشان‌دهنده‌ی این نکته است که یک پیش‌پردازش مناسب بدون داشتن اطلاعات اضافی در مورد محتوا و ساختار وب سایت ممکن نیست و این اطلاعات فرآیند تحلیل الگو را تا حد زیادی بهبود می‌بخشد.

۲-۴-۲-۳ فرمت داده

این مرحله آخرین قدم پیش‌پردازش است، وقتی که مراحل قبل کامل شدند، داده به فرم مناسبی فرمت خواهد شد تا تکنیک‌های کاوش روی آن اعمال شود. مرجع [۳۸]، داده های استخراج شده از وب را با استفاده از شمای حقیقی کلیک[۲] در یک پایگاه داده‌ی رابطه‌ای ذخیره می‌کند تا به این‌صورت پرس‌وجو در لاگ آسان‌تر شود و به تبع آن کاوش الگوهای مکرر[۳]، راحت‌تر شود. مرجع [۱۴]، متدی بر اساس درخت امضاء[۴] برای ایندکس کردن لاگ ذخیره شده در پایگاه‌های داده برای پرس‌وجوی کارای الگو ارائه می‌دهد.

یک ساختار درختی به نام درخت وَپ[۵] نیز در [۳۹]، معرفی شده است تا توالی دسترسی به صفحات وب ثبت شود. این ساختار برای اجرای الگوریتم‌های کاوش توالی که توسط همان نویسنده ایجاد شده، بهینه شده است. مرجع [۴۰]، داده‌ی لاگ را در یک ساختار درختی دیگر به نام درخت اف‌بی‌پی[۶]،

---

[۱] Content sources

[۲] Click fact schema

[۳] Frequent patterns

[۴] Signature tree

[۵] Wap-tree

[۶] FBP-tree

ذخیره می‌کند و به این‌صورت کشف الگوهای توالی را بهبود می‌بخشد. مرجع [۴۱]، از ساختاری شبه مکعب، برای ذخیره‌ی اطلاعات نشست‌ها استفاده می‌کند و به این‌وسیله استخراج تکه‌های مکعب[۱]، را که توسط تکنیک‌های خوشه‌بندی استفاده می‌شود، بهبود می‌دهد.

ما در این پروژه ساختار داده‌ی پیچیده‌ای استفاده نخواهیم کرد، بلکه از یک آرایه‌ی باینری به طول کل URLهای موجود در داده‌های دسترسی استفاده شده است که در خانه‌های متناظر با URLهایی که در نشست مربوطه ملاقات شده‌اند یک و URLهای ملاقات نشده صفر ثبت می‌شود. همچنین مدت درنگ کاربر بر روی هر صفحه‌ی ملاقات شده و تعداد ملاقات هر صفحه در نشست در آرایه‌های جداگانه‌ای ثبت می‌شوند.

۲-۴-۳ تکنیک‌ها

بر عکس تحقیقات آکادمیک در زمینه‌ی کاوش اطلاعات از داده‌های دسترسی به وب که بیشتر تمرکز بر توسعه‌ی تکنیک‌های کشف دانشی دارد که مختص تحلیل داده‌های WUM طراحی شده‌اند، در بیشتر برنامه‌های تجاری، این نوع کاوش را با ادغام تکنیک‌های آماری انجام می‌دهند. بیشتر تحقیقات از یک یا ترکیبی از سه تکنیک زیر استفاده کرده‌اند: قوانین انجمنی، الگوهای ترتیبی[۲] و خوشه‌بندی[۴۳].

قوانین انجمنی بحث مربوط به این پروژه است که در بخش‌های بعد با تفضیل بیشتری به آن پرداخته می‌شود. ضمناً همان‌طور که در بخش‌های پیش‌تر به این نکته اشاره شد که الگوریتم پیشنهادی در کلاس مسائل خوشه‌بندی هم قرار می‌گیرد، بحث خوشه‌بندی را نیز با تفضیل کمی بیشتری دنبال خواهیم کرد.

---

[۱] Cube slice

[۲] Sequential patterns

۲-۴-۳-۱ قوانین انجمنی

این تکنیک جزو اولین تکنیک‌های کاوش داده است و در عین حال پر استفاده‌ترین تکنیک در کاوش اطلاعات از داده‌های دسترسی به وب.

قوانین انجمنی استلزاماتی به فرم $X \Rightarrow Y$ هستند، که در آن X بدنه‌ی قانون و Y سر قانون است و X و Y مجموعه‌ای از آیتم‌های متعلق به مجموعه‌ای از تراکنش‌ها و با شرط $X \cap Y = \phi$ هستند. قانون $X \Rightarrow Y$ بیان می‌کند که تراکنش‌هایی که دارای آیتم‌های موجود در X هستند احتمال داشتن آیتم‌های Y را هم دارند. قدرت یک قانون انجمنی بر اساس دو معیار پشتیبانی[۱] و اطمینان[۲] محاسبه می‌شود. پشتیبانی تعیین می‌کند که چند بار یک قانون در تراکنش صدق می‌کند. معیار پشتیبانی از تقسیم تعداد تکرار $X \cup Y$ بر تعداد کل تراکنش‌ها به‌دست می‌آید. اطمینان، تعیین می‌کند که چند بار آیتم‌های در Y در تراکنش‌هایی که شامل X نیز هستند ظاهر می‌شوند. این دو معیار به ترتیب به صورت رابطه‌های (۲-۱) و (۲-۲) نشان داده می‌شوند.

$$s(X \Rightarrow Y) = \frac{\sigma(X \cup Y)}{N} \qquad (۲-۱)$$

$$c(X \Rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)} \qquad (۲-۲)$$

که N در رابطه‌ی (۲-۱) تعداد کل تراکنش‌هاست.

معیار پشتیبانی، معیار مفیدی است زیرا میزان مهم بودن قانون را منعکس می‌کند و قوانینی که دارای میزان پشتیبانی کمی هستند به ندرت مشاهده می‌شوند و بنابراین احتمالاً تصادفی به‌وقوع پیوسته‌اند. قوانین دارای پشتیبانی کم، قوانین مفیدی نیستند.

---

[۱] Support

[۲] Confidence

اطمینان معیار مفید دیگریست که نشان می‌دهد پیشگویی انجام شده توسط آن قانون تا چه اندازه معتبر است. برای یک قانون به‌صورت $X \Rightarrow Y$، هر چه اطمینان بیشتر باشد، احتمال وجود آیتم Y در تراکنشی که X در آن حضور دارد، بیشتر است.

مسئله‌ی کاوش قوانین انجمنی به فرم زیر است [۱]:

اگر مجموعه‌ای از تراکنش‌ها T در دست باشد، همه‌ی قوانین با پشتیبانی بزرگ‌تر از آستانه‌ی کمینه‌ی *minsup* و اطمینان بزرگ‌تر از آستانه‌ی کمینه‌ی *minconf* به عنوان قوانین انجمنی در آن پایگاه داده انتخاب می‌شوند.

استراتژی معمولی که در اکثر الگوریتم‌های استخراج قوانین انجمنی به‌کار می‌رود، شکستن این مسئله به دو زیر مسئله است :

۱- ایجاد مجموعه آیتم‌های تکرارشونده[۱]: پیدا کردن همه‌ی مجموعه آیتم‌هایی که شرط آستانه‌ی از پیش تعیین شده‌ای را رعایت می‌کنند.

۲- تولید قوانین: استخراج قوانین با درجه اطمینان بالا (قوانین قوی) از مجموعه آیتم‌های تکرارشونده‌ی به‌دست آمده در مرحله قبل.

مرحله‌ی ایجاد مجموعه آیتم‌های تکرار شونده، مرحله‌ی زمانگیر و چالش برانگیزیست و تکنیک‌های متفاوتی برای ایجاد این مجموعه آیتم‌ها به‌وجود آمده است. مشهورترین روش برای ایجاد مجموعه آیتم‌های تکرار شونده، اپریوری[۲] است که بر مبنای مقدار پشتیبانی، آیتم‌ها را انتخاب می‌کند.

از آن‌جا که اپریوری مهم‌ترین الگوریتم برای پیدا کردن مجموعه آیتم‌های مکرر است و الگوریتم‌های دیگری که طراحی شده‌اند نیز تغییر یافته‌ی اپریوری هستند، به این الگوریتم با جزئیات بیشتری پرداخته می‌شود.

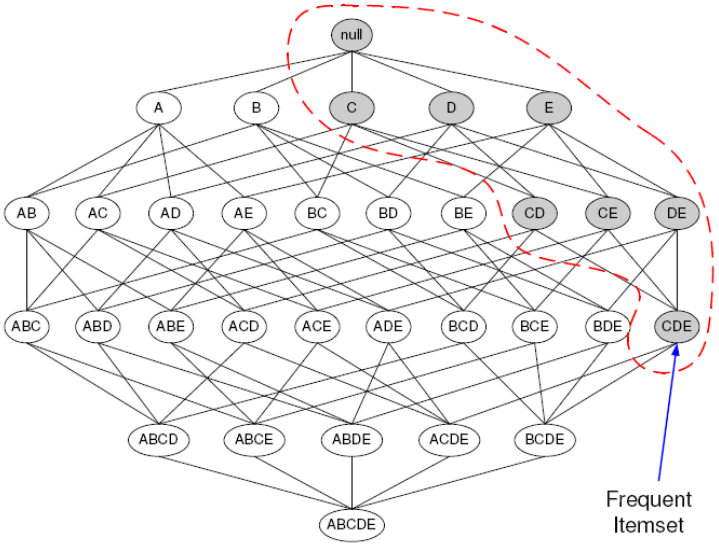
---

[۱] Frequent Itemsets

[۲] Apriori

اصل اپریوری: اگر یک مجموعه آیتم مکرر باشد، همه‌ی زیرمجموعه‌های آن نیز مکرر خواهند بود. به‌وسیله‌ی این اصل، فضای جستجوی نمایی را هرس کرده و مجموعه آیتم‌های مکرر را پیدا می‌کنند (شکل ۲-۱).

الگوریتم اپریوری: الگوریتم به این‌صورت است که ابتدا، هر آیتم یک کاندیدا برای یک مجموعه آیتم یک‌آیتمه محسوب خواهد شد. از این به بعد مجموعه آیتم n آیتمه به صورت مجموعه آیتم-nعضوی نشان داده خواهد شد. یک آستانه‌ی کمینه‌ی پشتیبانی *minsup* تعیین خواهد شد و کاندیداهایی که میزان پشتیبانی آن‌ها کمتر از *minsup* باشد از لیست کاندیداها حذف می‌شوند. آیتم‌های باقی‌مانده برای تولید مجموعه‌آیتم‌های-۲عضوی استفاده می‌شوند. در این حالت از m مجموعه‌آیتم-۱عضوی مکرر $^{m}C_2$ کاندیدای مجموعه آیتم-۲عضوی تولید خواهد شد. همین‌طور پیداکردن مجموعه آیتم‌های-۳عضوی و بیشتر ادامه پیدا می‌کند. بر اساس اصل اپریوری، تنها مجموعه آیتم‌هایی نگهداری می‌شوند که زیرمجموعه‌های آنها مکرر هستند.



شکل ۲-۱ نمایش اصل اپریوری. اگر $\{C,D,E\}$ مکرر باشد، همه‌ی زیرمجموعه‌های این مجموعه‌ی مکرر نیز مکرر خواهد بود.

شبه‌کد الگوریتم اپریوری در شکل ۲-۲ مشاهده می‌شود. فرض کنید C_k و F_k به‌ترتیب نشان‌دهنده‌ی مجموعه‌ای از مجموعه آیتم‌های کاندیدا و مجموعه آیتم‌های مکرر با سایز k باشد.

یک توصیف سطح بالا از الگوریتم در زیر آمده است:

- در ابتدا، پایگاه داده یک‌بار پیمایش می‌شود تا مجموعه‌ی F_1، مجموعه‌ی همه‌ی مجموعه آیتم‌های مکرر-۱عضوی به‌دست آید.
- الگوریتم به‌طور تکراری، مجموعه‌آیتم‌های مکرر بزرگ‌تر را با ۱) تولید مجموعه آیتم‌های کاندیدا با استفاده از تابع apriori-gen، (قدم ۵)، ۲) شمارش پشتیبانی برای هر کاندیدا با پیمایش پایگاه داده (قدم‌های ۶-۱۰) و ۳) تعیین مجموعه آیتم‌های مکرر با مقایسه‌ی پشتیبانی هر کاندیدا با آستانه‌ی *minsup* (قدم ۱۲) تولید می‌کند.
- الگوریتم زمانی‌که هیچ مجموعه آیتم مکرر جدیدی تولید نشود $F_k = \phi$، به اتمام می‌رسد (قدم ۱۳).

خصوصیات کلی الگوریتم اپریوری به شرح زیر خلاصه می‌شود:

اپریوری یک الگوریتم سطح-مبنا[۱] است که مجموعه آیتم‌های مکرر را سطح به سطح تولید می‌کند. یعنی در سطح اول مجموعه آیتم‌های یک عضوی و به ترتیب مجموعه آیتم‌های با تعداد عضوهای بیشتر. در هر سطح، مجموعه آیتم‌های کاندیدا با استفاده از مجموعه آیتم‌های مکرر استخراج شده در سطوح قبلی تولید می‌شوند.

در هر سطح، تراکنش یکبار پویش می‌شود تا میزان معیار پشتیبانی[۲] هر مجموعه آیتم کاندیدا تعیین شود.

---

[۱] Level-wise

[۲] Support

```
1: k = 1.
2: F_k = { i | i ∈ I ∧ σ({i})/N ≥ minsup}.    {Find all frequent 1-itemsets}
3: repeat
4:    k = k + 1.
5:    C_k = apriori-gen(F_{k-1}).    {Generate candidate itemsets}
6:    for each transaction t ∈ T do
7:       C_t = subset(C_k, t).    {Identify all candidates that belong to t}
8:       for each candidate itemset c ∈ C_t do
9:          σ(c) = σ(c) + 1.    {Increment support count}
10:      end for
11:   end for
12:   F_k = { c | c ∈ C_k ∧ σ(c)/N ≥ minsup}.    {Extract the frequent k-itemsets}
13: until F_k = ∅
14: Answer = ⋃ F_k.
```

شکل ۲-۲ شبه‌کد الگوریتم اپریوری

زمانی‌که قانون‌های انجمنی را به WUM اعمال می‌کنیم، هدف پیداکردن ارتباط بین صفحات وبی است که مکرراً با هم در نشست‌های کاربران ظاهر می‌شوند. این قوانین در کاربرد WUM عموماً به صورت زیر هستند:

A.html, B.html => C.html

که بیان می‌کند، اگر یک کاربر صفحه‌ی A.html و B.html را ملاقات کرده باشد، احتمال این‌که در همان نشست، همان کاربر صفحه C.html را نیز ملاقات کند، زیاد است. این نوع نتایج، برای مثال در [۱۵] و [۴۳] با استفاده از الگوریتم اپریوری تغییر یافته [۴۲] ایجاد شده‌اند.

مرجع [۴۴]، معیارهای مطلوب برای ارزش‌گذاری بر قوانین انجمنی استخراج شده از داده‌های دسترسی به وب را ایجاد و ارزیابی کرده است. مرجع [۴۵]، یک تکنیک ترکیب شده از قوانین انجمنی و منطق فازی برای استخراج قوانین انجمنی فازی از لاگ‌های وب، استفاده کرده است. همان‌طور که از ادبیات این زمینه مشخص است، الگوریتمی که بتواند از داده‌های حجیم و خلوت، مجموعه آیتم‌های مکرر را استخراج کند وجود ندارد و الگوریتم‌های موجود همه برپایه‌ی الگوریتم اپریوری ایجاد شده‌اند که به هیچ‌وجه برای داده‌های حجیم و کاربردهای بلادرنگ قابل استفاده نیستند زیرا هم هزینه زمانی

و هم هزینه‌ی حافظه‌ی بالایی دارند. الگوریتمی که در این پروژه ارائه می‌شود از زاویه‌ای در دسته‌ی الگوریتم‌هایی قرار می‌گیرد که برای کشف مجموعه آیتم‌های مکرر به کار می‌روند.

۲-۴-۳-۲ الگوهای ترتیبی

الگوهای ترتیبی برای کشف زیر توالی‌های مکرر بین حجم زیادی از داده‌ی ترتیبی به‌کار می‌روند. در WUM، الگوهای ترتیبی برای پیدا کردن الگوهای راهبری ترتیبی که مکرراً در نشست‌های کاربران مشاهده می‌شود، به‌کار می‌روند. الگوی ترتیبی معمولی، فرمی به شکل زیر دارد [۴۶]: ۷۰٪ کاربرها که اول صفحه‌ی A.html و سپس B.html را ملاقات کرده‌اند، صفحه‌ی C.html را نیز در همان نشست ملاقات کرده‌اند.

الگوهای ترتیبی ممکن است از لحاظ نحوی[۱] مشابه به قوانین انجمنی باشند. در حقیقت الگوریتم‌های استخراج قوانین انجمنی برای کاوش الگوهای ترتیبی نیز بکار می‌روند. به هر حال، الگوهای ترتیبی شامل زمان هم می‌شوند. به این‌صورت که در کدام نقطه از توالی، یک حادثه رخ داده است. در مثال بالا، صفحه A، B و C متوالیاً یکی بعد از دیگری در نشست کاربر ظاهر شدند. در مورد قوانین انجمنی، اطلاعاتی در مورد توالی حوادث وجود ندارد. دو کلاس اصلی از الگوریتم‌ها برای استخراج الگوهای ترتیبی وجود دارد. یکی شامل روش‌هایی مبتنی بر کاوش قوانین انجمنی است و دیگری شامل روش‌هایی مبتنی بر استفاده از ساختارهای درختی و زنجیره‌های مارکوف برای نمایش مسیر راهبری. برخی از الگوریتم‌های شناخته شده برای کاوش قوانین انجمنی برای استخراج الگوهای ترتیبی تغییر یافته‌اند. برای مثال [۴۴] و [۴۷]، از اپریوری‌آل[۲] و جی‌اس‌پی[۳]، که دو نسخه‌ی توسعه‌یافته از الگوریتم اپریوری هستند [۴۲]، استفاده کرده‌اند. مرجع [۳۹]، ادعا می‌کند که

---

[۱] Syntactically

[۲] AprioriAll

[۳] GSP

الگوریتم‌های مربوط به قوانین انجمنی زمانی‌که برای به‌دست آوردن الگوهای ترتیبی طولانی استفاده شوند، کارا نیستند که این مشکل هنگام کار با لاگ‌های وب بیشتر بروز می‌کند. از این رو [۳۹]، یک الگوریتم دیگر که در آن ساختار درختی درخت وَپ برای نشان دادن الگوهای راهبری استفاده می‌شود، ارائه کرده است. الگوریتم وَپ‌ماین[۱] [۳۹] و ساختار داده‌ی درخت وَپ، به‌طور خاص برای کاوش الگوهای دسترسی به وب ایجاد شده‌اند. وَپ‌ماین از بقیه‌ی الگوریتم‌های شبیه اَپریوری مانند جی‌اس‌پی بهتر عمل می‌کند. ساختارهای درختی در [۴۰] نیز استفاده شده‌اند. مرجع [۴۷]، مقایسه‌ای بین سه الگوریتم متفاوت برای کشف الگوهای ترتیبی در داده‌های WUM، ارائه کرده است. این سه الگوریتم عبارتند از ۱) پی‌اس‌پی‌پلاس[۲] که تکامل یافته‌ی جی‌اس‌پی است و بر اساس تولید کاندیدا و اکتشافات تستی[۳] عمل می‌کند، ۲) فیری‌اسپن[۴] [۴۸]، که بر اساس ترکیبی از کاوش ترتیب‌های مکرر و کاوش الگوهای مکرر تکامل یافته و ۳) پِرِفیکس‌اسپن[۵] [۴۹]، که از رویکردی مبتنی بر تصویر داده[۶]، استفاده می‌کند. نتیجه‌ی مقایسه‌ی انجام شده در [۴۷] نشان می‌دهد که پِرِفیکس‌اسپن از بقیه‌ی دو الگوریتم بهتر عمل می‌کند و حتی در توالی‌های طولانی کارایی خوبی نشان می‌دهد. مرجع [۵۰]، یک متد پیوندی[۷] ارائه می‌دهد که در آن داده‌ها در یک پایگاه داده بر اساس شمایی به نام شمای حقیقی کلیک[۸] ذخیره می‌شوند و یک گرامر احتمالی ابرمتن (HPG) با پرس‌وجوی پایگاه داده تولید می‌شود. HPGها تراکنش‌های بین صفحات وب را از طریق مدلی که

---

[۱] WAP-mine

[۲] PSP+

[۳] Test heuristic

[۴] FreeSpan

[۵] PrefixSpan

[۶] Data projection

[۷] Hybrid

[۸] Click fact

شباهت‌های زیادی با زنجیره‌ی مارکوف دارد، نمایش می‌دهد. الگوهای ترتیبی مکرر، از طریق جستجوی اول سطح بر روی HPG کاوش می‌شوند. HPG اولین بار در [۵۱] پیشنهاد شد و بعدها در [۵۰] توسعه پیدا کرد.

۲-۴-۳-۳ تکنیک‌های خوشه‌بندی

تکنیک‌های خوشه‌بندی، در جستجوی گروهی از آیتم‌های مشابه بین حجم عظیمی از داده هستند. این کار بر اساس ایده‌ی کلی تابع فاصله که شباهت بین گروه‌ها را محاسبه می‌کند، انجام می‌شود. خوشه‌بندی به‌طور گسترده‌ای در WUM برای گروه کردن نشست‌های مشابه انجام شده است [۳۵] [۴۱] [۵۲] [۲۱].

مرجع [۵۳] اولین مقاله‌ای بود که پیشنهاد کرد تمرکز WUM باید بر گروهی از نشست‌های کاربر باشد نه بر یک نشست کاربر. مرجع [۵۳] همچنین اولین مقاله‌ای بود که خوشه‌بندی را برای تعیین خوشه‌هایی از نشست‌های مشابه استفاده کرد. مرجع [۳۵] گراف مشابهت را در تلفیق با زمان سپری شده در صفحات وب برای تخمین شباهت گروه در خوشه‌بندی مبتنی بر مفهوم پیشنهاد کرده است. مرجع [۵۴] از هم‌راستایی[۱] توالی برای اندازه‌گیری شباهت استفاده کرده است و [۵۳] از توابع باور[۲]. مرجع [۵۵] از الگوریتم ژنتیک برای توسعه‌ی نتایج خوشه‌بندی از طریق بازخورد کاربر، استفاده کرده‌است. مرجع [۵۶] AIS فازی را ارائه می‌دهد که از تکنیک‌های سیستم ایمنی مصنوعی و خوشه‌بندی برای توسعه‌ی پروفایل‌های کاربران که از طریق خوشه‌بندی تولید شده‌اند، استفاده کرده است، در این کار از شبکه‌ی ایمنی با آنتی‌بادی‌های فازی برای خوشه‌بندی پروفایل‌های کاربران استفاده شده است.

---

[۱] Alignment

[۲] Belief functions

مرجع [۵۲]، خوشه‌بندی چندحالته[۱] را به‌کار برده است. مرجع [۵۷] کاربردی از خوشه‌بندی ماتریسی را برای داده‌های WUM ارائه کرده است. مرجع [۵۸]، کاوش قوانین انجمنی و خوشه‌بندی را در یک متد به نام افراز ابرگراف قوانین انجمنی[۲]، ترکیب کرده است. در این متد، قوانین انجمنی برای استخراج الگوهای مکرر از نشست‌های کاربران استفاده می‌شود و سپس الگوهای مکرر برای ساختن یک گراف که در آن ۱) نودها صفحات وب ملاقات شده هستند. ۲) لبه ها دو یا تعداد بیشتری نود را به هم متصل می کنند البته در صورتی‌که الگوی مکرری وجود داشته باشد که صفحات (نودها) مربوطه را در بر داشته باشد. ۳) لبه‌ها بر اساس ارتباط الگوهایی که نودها را به هم متصل کرده‌اند، وزن‌دهی می‌شوند.

توجه کنید که قوانین بالا، یک ابرگراف را تعریف می‌کند، زیرا یک لبه می‌تواند بیش از دو نود را به هم متصل کند. ابرگراف بطور بازگشتی در خوشه‌ها دسته‌بندی می‌شود تا گروه‌های مطلوب رفتار کاربران را تعیین کند.

در این قسمت لازم است به سه روش خوشه‌بندی که برای خوشه‌بندی مجموعه داده‌های بسیار بزرگ طراحی شده‌اند و در ادامه‌ی پروژه از آن‌ها استفاده می‌شود، اشاره‌ای کرد. BIRCH [۵۹] متد خوشه‌بندی است که برای مجموعه داده‌های بسیار بزرگ طراحی شده است. این مدل خوشه‌بندی از ساختار داده‌ی خاصی به نام درخت سی‌اف[۳]استفاده می‌کند، نسبت به نویز مقاوم نیست و داده‌ها را بدون نویز تلقی می‌کند. از خصوصیات این الگوریتم اینست که با هر مقدار حافظه‌ای قابل اجراست و پیچیدگی I/O آن کمی بیشتر از یک‌بار اسکن داده‌ها است. DBSCAN [۶۰] یک الگوریتم خوشه‌بندی برای مجموعه داده‌های بزرگ همراه با نویز است. در این الگوریتم خوشه‌ها بر مبنای

---

[۱] Multi-modal

[۲] Association rule hyper graph partitioning

[۳] C F Tree

چگالی تشکیل می‌شوند. در این الگوریتم نیز از ساختار داده‌ی خاصی به نام درخت آر اِستار[1] استفاده می‌شود. الگوریتم SKM [۶۱] نیز الگوریتمی است که بر مبنای افراز کردن داده‌ها عمل می‌کند. این الگوریتم نسبت به نویز مقاوم نیست و در این الگوریتم نیز از ساختار داده‌ی خاصی استفاده می‌شود. از خصوصیات این الگوریتم پیچیدگی زمانی خطی آن است.

الگوریتم ارائه شده در این پروژه را می‌توان از زاویه‌ی دید دیگری در دسته الگوریتم‌های خوشه‌بندی قرار داد، به این‌صورت که مجموعه آیتم‌های مکرری که در نهایت با استفاده از الگوریتم پیشنهادی تولید می‌شوند را می‌توان مراکز خوشه‌های موجود در داده‌ها در نظر گرفت.

۲-۴-۴ از تکنیک تا کاربرد

وقتی با یک کار WUM روبرو می‌شویم، مشکل اصلی انتخاب مناسب‌ترین تکنیک برای مسئله‌ی مورد بحث است. مانند همه‌ی کاربردهای دیگر، برای مسائل WUM نیز یک پاسخ یکتا برای چنین سوالی وجود ندارد. در دو بخش قبلی، تکنیک‌های مورد استفاده برای WUM را شرح دادیم. عموماً همه‌ی تکنیک‌های معرفی شده برای هر چهار کاربرد WUM قابل استفاده هستند. چهار کاربرد WUM عبارتند از:

- شخصی‌سازی وب
- پیش‌بینی نیاز کاربران به منظور اجابت درخواست‌ها، قبل از صادر شدن درخواست[2] یا اجابت سریع درخواست از طریق ذخیره در حافظه‌ی نهانی
- پشتیبانی برای طراحی وب‌سایت‌ها[3]
- تجارت الکترونیک

---

[1] R*Tree

[2] Pre-fetching

[3] Support to design

برای مثال در [۶۲]، یک تکنیک خوشه‌بندی سلسله مراتبی برای گروه‌بندی کاربران وب سایت بر اساس علایقشان و سپس شخصی‌سازی محتویات وب بر اساس گروهی که کاربر به آن تعلق دارد، ارائه شده است. در [۲۳]، هر دو تکنیک الگوهای ترتیبی و خوشه‌بندی برای شخصی‌سازی محتوای وب به‌کار رفته‌اند. خوشه‌بندی برای گروه‌بندی نشست کاربران مشابه استفاده می‌شود، در حالی‌که الگوهای ترتیبی با زنجیره‌های مارکوف برای پیش‌بینی رفتار کاربر، به‌کار می‌رود.

مرجع [۶۳]، از قوانین انجمنی برای سیستم توصیه‌کننده، استفاده می‌کند. در [۴۳] قوانین انجمنی برای حل مشکل نگهداری در حافظه‌ی نهانی و از قبل‌آوردن صفحات مورد درخواست استفاده می‌شود. برای حل همین مشکل در [۶۴] از کلاس‌بندی مبتنی بر الگوهای ترتیبی استفاده می‌شود. مرجع [۴۶] از الگوهای ترتیبی برای تولید وب‌سایتهای وفق‌پذیر پویا استفاده می‌کند. در [۶۵]، کلاس‌بندها برای کلاس‌بندی صفحات وب بر اساس مسیر کاربرها و سپس صفحات کلاس‌بندی شده برای تشخیص ساختار وب سایت استفاده می‌شوند. مرجع [۶۶]، نشان می‌دهد چگونه تکنیک‌های مختلف کاوش داده، مانند قوانین انجمنی، الگوهای ترتیبی، کلاس‌بندی و خوشه‌بندی برای دسته‌بندی مشتریان[۱] و پروفایل‌کردن در کاربردهای تجارت الکترونیک به‌کار می‌روند. در جدول۱، کاربردهای WUM خلاصه شده و تکنیک‌هایی که در تحقیقات مرور شده، استفاده شده‌اند، ذکر شده است. توجه کنید که ارتباط مشخصی بین تکنیک و کاربرد وجود ندارند و همه‌ی تکنیک‌ها را برای همه‌ی کاربردها می‌توان استفاده کرد. الگوریتمی که در این پروژه ارائه می‌شود، یک الگوریتم عمومی است و بنابراین قابل اعمال برای تمامی کاربردهای WUM است و از آن‌جا که با یک‌بار عبور از داده‌ها الگوریتم الگوهای تکرار شونده را استخراج می‌کند و بنابراین بلادرنگ است، برای کاربردهایی از قبیل پیشنهاددهی بسیار مناسب به نظر می‌رسد.

[۱] Customer segmentation

جدول۲-۱ WUM، ارتباط بین کاربردها، تکنیک‌ها و منابع داده

Paper	Application	Technique	Data
[۲۳]	Personalization	Clustering, Association rules	Web Server
[۶۷]	Personalization	Clustering	Web Server
[۱۸]	Personalization		Client
[۶۸]	Personalization	Fuzzy Clustering	Web Server
[۶۲]	Personalization	Clustering	Web Server
[۶۳]	Personalization	Association Rules	Web Server
[۵۸]	Personalization	Association Rules, Clustering	Web Server
[۵۳]	Personalization	Clustering	Web Server
[۶۹]	Personalization	Clustering, Sequential Patterns	Web Server
[۷۰]	Caching	Classifiers, Association Rules, Sequential Patterns	Proxy Server
[۷۱]	Caching	Association Rules	Web Server
[۶۴]	Caching	Association Rules, Classifiers	Web Server
[۷۲]	Caching	Association Rules	Proxy Server
[۷۳]	Caching	Markov Models	Web Server
[۱۰۳]	Caching	Association Rules	Web Server
[۱۰۴]	Design	Markov Models	Web Server
[۷۶]	Design	Classifiers, Sequential Patterns	Web Server
[۶۵]	Design	Classifiers	Web Server
[۷۷]	Design	Sequential Patterns	Web Server
[۷۸]	Design	Sequential Patterns	Web Server
[۷۹]	Design	Markov Models	Web Server
[۲۰]	E-commerce		Web Server
[۶۶]	E-commerce	Classifiers, Association Rules, Sequential Patterns	Web Server
[۸۰]	E-commerce	Clustering	Web Server
[۵۴]	E-commerce	Fuzzy Logic, Clustering, Genetic Algorithm	Web Server

۲-۵ جمع‌بندی

در این فصل دیدی کلی بر وب کاوی ایجاد شده و بطور اخص به مرور روش‌های استخراج اطلاعات از داده‌های دسترسی به وب پرداخته و به نوع و منبع داده‌ها در این کاربرد اشاره شد. روش‌های پاک‌سازی و پیش‌پردازش‌های لازم برای داده‌های دسترسی به وب توصیف شد. روش‌های کاوش اطلاعات که در ادبیات این موضوع موجود هستند نام برده شد. الگوریتم اپریوری که مهمترین الگوریتم کشف مجموعه آیتم‌های تکرارشونده است، توضیح داده شد و در نهایت مختصراً به کاربردهای استخراج اطلاعات از داده‌های دسترسی به وب پرداخته شد.

۳

سیستم ایمنی مصنوعی

مقدمه

سیستم ایمنی مصنوعی: الگویی الهام گرفته شده از بیولوژی ایمنی

سیستم ایمنی مصنوعی

چهارچوب سیستم ایمنی مصنوعی

سیستم ایمنی مصنوعی برای داده‌کاوی و کاربردهای دیگر

جمع‌بندی

"If one way be better than another, that you may be sure is nature's way."
--Aristotle, c.384 BC-322 BC

فصل ۳: سیستم ایمنی مصنوعی

۳-۱ مقدمه

در این بخش به تشریح سیستم ایمنی مصنوعی و طبیعی می‌پردازیم و اصولی را که در ادامه در این رساله استفاده کرده‌ایم، تشریح خواهیم کرد. این بخش را با ارائه‌ی خواص سیستم ایمنی طبیعی که انگیزه‌ی الهام گرفتن از آن شده است، شروع کرده و سپس به الگوریتم‌های ایمنی خواهیم پرداخت. این اطلاعات برای درک سیستم ایمنی مصنوعی ضروریست. در ادامه به سیستم مصنوعی الهام گرفته شده از سیستم ایمنی طبیعی خواهیم پرداخت و در نهایت بخش را با مروری بر کارهای انجام شده به‌وسیله‌ی AIS در داده‌کاوی به اتمام می‌رسانیم.

۳-۲ سیستم ایمنی مصنوعی: الگویی الهام گرفته شده از بیولوژی

به طور کلی، سیستم‌های ایمنی مصنوعی (AIS) جزء الگوریتم‌های الهام گرفته شده از بیولوژی هستند. همان‌طور که از نام آن‌ها برمی‌آید، این نوع الگوریتم‌ها، الگوریتم‌هایی کامپیوتری هستند که اصول و ویژگی‌های آنها نتیجه‌ی بررسی در خواص وفقی[۱] و مقاومت[۲] نمونه‌های بیولوژیکی است. نمونه‌ی چنین الگوریتم‌هایی، شبکه‌های عصبی [۸۱] که از مغز الهام گرفته شده است؛ بهینه‌سازی کلونی مورچه‌ها [۸۲] که از اصول رفتاری مورچه‌ها برای حل مسائل استفاده می‌کند؛ الگوریتم‌های تکاملی [۸۳] [۸۴]، که از اصول تئوری تکامل داروین برای حل مسائل استفاده می‌کند و سیستم ایمنی مصنوعی که از اصول و پردازش‌های سیستم ایمنی طبیعی برای حل مسائل استفاده می‌کند، هستند.

---

[۱] Adaptability

[۲] Robustness

در بین الگوریتم‌های الهام گرفته شده از طبیعت، الگوریتم انتخاب کلونی AIS اغلب با الگوریتم‌های ژنتیک [۸۵] مقایسه می‌شود. هر دوی این الگوریتم‌ها (انتخاب کلونی و الگوریتم ژنتیک) تحت نام کلی الگوریتم‌های تکاملی قرار می‌گیرند. الگوریتم‌های تکاملی (EAs) از اساس تکامل داروین استفاده می‌کنند. در این الگوریتم‌ها، جواب‌های بالقوه به فرم فردهای جمعیتی که در یک محیط بسته به سمت جواب متکامل می‌شوند، نشان داده می‌شوند.

انواع متفاوتی EA وجود دارد، اما همه‌ی آن‌ها مشترکاتی دارند. در [۸]، این مشترکات به‌صورت زیر بیان شده‌اند.

- جمعیتی از افراد، که هر کدام نماینده‌ی یک کاندیدای جواب است.
- یک متد انتخاب بر اساس معیار کیفیت کاندیدای جواب که یک فرد جمعیت است.
- تولید افراد جدید با یک متد ارث‌بری از افراد موجود. این افراد جدید، با اعمال عملگرهای احتمالی بر افراد فعلی جمعیت، ایجاد می‌شوند.

۳-۳ ایمنی

حین مرور ادبیات، به این نکته اشاره کردیم که در ابتدای به‌وجود آمدن AIS، نویسندگان مقلات در این زمینه، خود را موظف به توضیح مفصل سیستم ایمنی طبیعی می‌کردند و بیشتر این توضیحات، پیچیده‌تر از حد مورد نیاز بوده است. در این زیربخش به اصول ایمنی اشاره خواهیم کرد ولی اصول در سطحی بیان خواهد شد که به فهم ادامه پروژه کمک کند و به توضیح جزئیات اضافه نخواهیم پرداخت.

این بخش را با خلاصه‌ی کوتاهی از سیستم ایمنی ذاتی شروع می‌کنیم. سیستم ایمنی ذاتی در سیستمهای مصنوعی به‌طور گسترده استفاده نشده‌اند ولی از آن‌جا که به عملکرد سیستم ایمنی اکتسابی کمک می‌کنند و بر روی آن تاثیر می‌گذارند و در این پروژه نیز از این سیستم استفاده می‌شود، مروری کوتاه بر آن لازم به نظر می‌رسد.

۳-۳-۱ ایمنی ذاتی

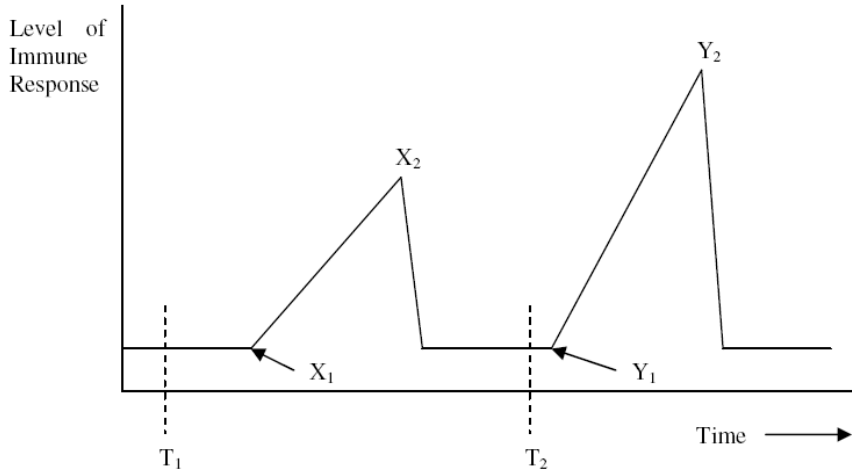
سیستم ایمنی ذاتی [۸۶]، همان‌طور که از نامش برمی‌آید با گذر زمان تغییر نمی‌کند و به همین دلیل در AIS به‌طور گسترده مورد استفاده قرار نگرفته است. به هر حال، این روند خصوصاً از زمان ارائه‌ی تئوری خطر [۸۷]، تغییر کرده است. افزایش مقالاتی مانند [۸۸] [۸۹] [۹۰] [۹۱] [۹۲] این تغییر روند را اثبات می‌کنند.

سیستم ایمنی ذاتی، برای تشخیص و حمله به تعداد کمی از مهاجمان معمول، تنظیم شده‌است. سیستم ایمنی ذاتی بیشتر پاتوژن‌ها را (که مهاجمان بالقوه مضرّ هستند) در برخورد اول منهدم می‌کند و بنابراین سیستم ایمنی ذاتی در موارد کمی مورد نیاز خواهد بود. سیستم ایمنی اکتسابی، برای ایجاد واکنش در برابر مهاجمان احتیاج به وقت دارد و بنابراین وظیفه‌ی سیستم ایمنی ذاتی است که در برابر مهاجمان به سرعت واکنش نشان دهد و حمله را تحت کنترل بگیرد تا سیستم ایمنی اکتسابی بتواند پاسخ موثری ایجاد کند. سلول‌های سیستم ایمنی ذاتی اغلب به عنوان سلول‌های عرضه‌کننده‌ی آنتی‌ژن عمل می‌کنند؛ به این معنی که این سلول‌ها موجودیت‌هایی را که ممکن است پاتوژن باشند تشخیص داده و آن‌ها را به روش مناسبی به سلول‌های سیستم ایمنی اکتسابی عرضه می‌کنند. بنابراین سلول‌های سیستم ایمنی، نقشی حیاتی در ایجاد پاسخ اکتسابی دارند.

۳-۳-۲ ایمنی اکتسابی

ایمنی ذاتی زمانی‌که فعال می‌شود، برای چند روزی فعال می‌ماند. در حالی‌که ایمنی اکتسابی زمانی‌که فعال شود برای هفته‌ها فعال می‌ماند. وظیفه‌ی ایمنی اکتسابی است که زمانی‌که ایمنی ذاتی از پا در آمده یا به هر جهت ناکاراست پاتوژن‌ها را از بین ببرد. ناکارا بودن ایمنی ذاتی به این علت است که ایمنی ذاتی نمی‌تواند پاسخی خاص برای یک پاتوژن مهاجم ایجاد کند در این حالت سیستم ایمنی اکتسابی وارد عمل می‌شود. بر خلاف ایمنی ذاتی، پاسخ ایمنی اکتسابی خاص است و ایمنی اکتسابی حافظه دارد و بنابراین پاتوژنی که یک‌بار حمله کرده و سیستم برای آن پاسخی تولید کرده است را به

یاد می‌آورد و در برخوردهای بعدی، برای مقابله با آن پاتوژن پاسخ سریع‌تری تولید می‌کند. (شکل ۳-۱)



شکل ۳-۱: پاسخهای اولیه و ثانویه‌ی ایمنی. آنتی‌ژن ناآشنا در زمان t1 پاسخ X2 را تولید می‌کند ولی به تاخیر بین t1 وX1 توجه کنید. همان آنتی‌ژن که در زمان T2 وارد شده است تقریبا فورا در Y1 پاسخ ایجاد شده است و پاسخ Y2 از پاسخ X2 قویتر است.

اولین نشانه‌های وجود چنین سیستم اکتسابی، در آزمایشات واکسیناسیون که توسط جنر[۱] در سال ۱۷۹۰ انجام شد، دیده شد. بعد از آن تئوری‌های زیادی برای توضیح مشاهدات آزمایش واکسیناسیون داده شد تا ۱۸۹۰ که بهرینگ[۲] و کیتاساتو[۳] نشان دادند که ایمنی که با واکسیناسیون به‌وجود می‌آید به علت حضور المان‌های محافظ در خون به نام آنتی‌بادی است. بعد از آن بورنت[۴] انتخاب کلونی یا اصول توسعه‌ی کلونی را ارائه داد [۹۳]. در این مدل، هر سلول (سلول B) معرف یک آنتی‌بادی است، وقتی سلول به وسیله‌ی یک آنتی‌ژن تحریک شود، شروع به تکثیر و ترشح آنتی‌بادی می‌کند.

---

[۱] Jener

[۲] Behring

[۳] Kitasatto

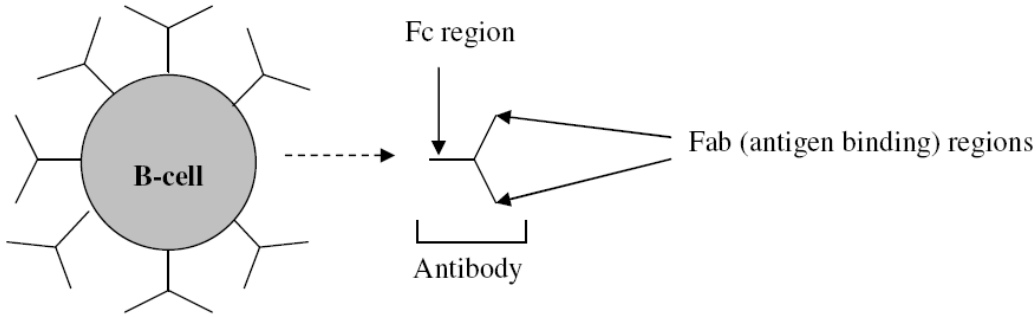
[۴] Burnett

امروزه پذیرفته شده که دو نوع سلول در سیستم ایمنی اکتسابی وجود دارد که به آن‌ها لنفوسیت گفته می‌شود. این دو نوع سلول، سلول B و سلول T نام دارند. این دو نوع سلول، تفاوت عملکرد کمی دارند. لنفوسیت‌ها دارای پذیرنده‌ای هستند که در سطح این سلول‌های ایمنی قرار دارند. هر مولکولی که قادر به اتصال به یکی از این پذیرنده‌ها باشد، آنتی‌ژن نام دارد. همین اتصال است که باعث فعال شدن لنفوسیت‌ها می‌شود. هماهنگی بین پذیرنده و آنتی‌ژن لازم نیست کامل باشد. قدرت اتصال بین آنتی‌بادی و آنتی‌ژن با نام میل پیوندی معروف است. اگر این میل پیوندی از حد آستانه‌ای بیشتر باشد، سلول ایمنی که اتصال به آن صورت گرفته، فعال می‌شود.

۳-۳-۲-۱ سلول‌های B و آنتی‌بادی‌ها و انتخاب کلونی

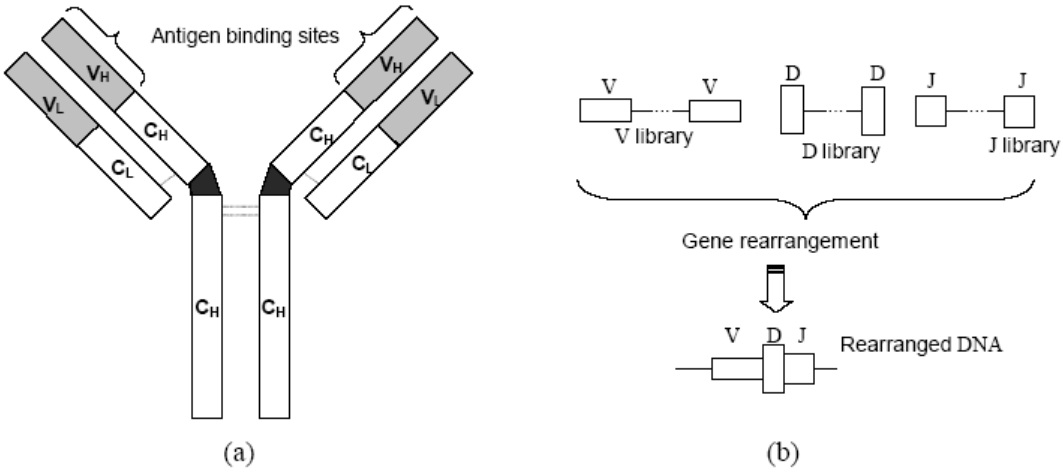
مانند همه سلول‌های ایمنی، سلول‌های B در مغز استخوان ساخته می‌شوند. یک سلول B طبیعی حامل 10^5 آنتی‌بادی (پذیرنده) در سطح خود است. هر کدام از این آنتی‌بادی‌ها دارای شکلی هستند که در ساختار ژنتیکی سلول B است و بنابراین همه شکلی مشابه به سلول B دارند. بنابراین همه‌ی آنتی‌بادی‌هایی که توسط یک سلول B تولید شده‌اند به مجموعه‌ی مشابهی از الگوهای مولکولی متصل می‌شوند. آنتی‌بادی‌های سلول B دو ارزشی و دو عملکردی[۱] هستند. آنها دو ارزشی‌اند زیرا می‌توانند از طریق دو بازوی نواحی Fab به دو آنتی‌ژن متصل شوند (شکل ۳-۲) و دو عملکردی‌اند زیرا به‌علاوه‌ی اتصال از طریق نواحی Fab به الگوهای آنتی‌ژن می‌توانند از طریق قسمت Fc به پذیرنده‌های خاص روی سطح سلول‌های ایمنی دیگر نیز متصل شوند.

[۱] Bi-functional



شکل ۳-۲ دیاگرام سلول B. آنتی‌بادی‌های زیادی در سطح سلول B دیده می‌شود.

یک آنتی‌بادی (Ab)، یا یک ایمونوگلوبین (Ig)، از دو زنجیر سبک مشابه (L) و دو زنجیر سنگین (H) مشابه، تشکیل شده است.



شکل ۳-۳ مولکول آنتی‌بادی و ژنومش. (a) منطقه متغیر (منطقه V) مسئول تشخیص آنتی‌ژن و منطقه ثابت (منطقه C) مسئول اعمال متعددی مانند تثبیت مکمل. (b) پروسه بازآرایی که منجر به شکل‌گیری منطقه متغیر زنجیره‌ی سنگین مولکول آنتی‌بادی می‌شود: تکه‌های ژن (دقیقاً یکی از هر کتابخانه‌ی ژنی) به ترتیب به هم الحاق می‌شوند. سپس محصول نهایی به مولکول آنتی‌بادی کارا ترجمه می‌شود. V، D، J ، کتابخانه‌های منحصربه‌فردی هستند که در تولید پاسخ ایمنی شرکت می‌کنند.

اتصال به آنتی‌ژن دو نقش بازی می‌کند، نقش اصلی، برچسب زدن به آنتی‌ژن به عنوان یک مخرب است تا بقیه‌ی سلول‌های سیستم ایمنی آن را تخریب کنند به این عملیات آپسونایز یا آماده‌ی مرگ

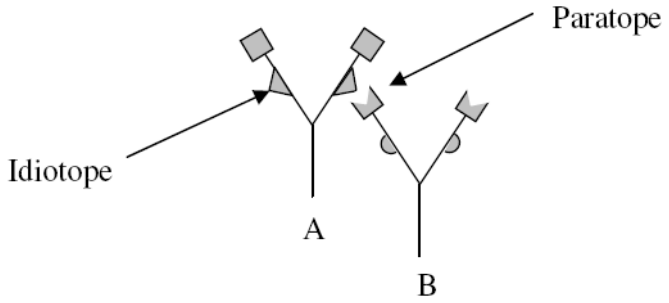
کردن می‌گویند، این کار با اتصال نواحی Fab آنتی‌بادی به آنتی‌ژن صورت می‌گیرد. ناحیه‌ی Fc دست‌نخورده باقی می‌ماند تا اتصال با سلول‌های دیگر ایمنی مانند ماکروفاژها صورت پذیرد.

زمانی که یک سلول B، به یک الگوی آنتی‌ژنی متصل شود (یعنی آنتی‌ژن را شناسایی کند)، سلول B تکثیر شده و تعداد زیادی سلول B یکسان تولید می‌شود. حدود ۱۲ ساعت طول می‌کشد تا یک سلول B رشد کرده و به دو سلول تبدیل شود. بعد از تحریک شدن، دوره تکثیر حدود یک هفته طول می‌کشد. بنابراین از یک سلول، ۲ به توان ۱۴ (حدود ۱۶۰۰۰) سلول مشابه تولید می‌شود. به هر حال، هر چه میل پیوندی بین سلول B و آنتی‌ژن بیشتر باشد، نرخ تکثیر بیشتر خواهد بود. بنابراین سلولهای B با میل پیوندی بالاتر، کلونی‌های بیشتری تولید می‌کنند. به این فرآیند، اصل انتخاب کلونی می‌گویند. این فرآیند شبیه به فرآیند انتخاب طبیعی داروین است. اصل انتخاب کلونی در AIS الگوریتم خاص خود را دارد [۹۴]. بعد از تکثیر، سلول‌های B شروع به بالغ شدن می‌کنند. این فرآیند در سه مرحله صورت می‌گیرد: دگرگونی ایزوتایپ، بلوغ میل پیوندی و تصمیم گیری بین حافظه یا پلاسما شدن سلول B. معمولاً AIS با دگرگونی ایزوتایپ درگیر نیست ولی دو مرحله‌ی بعدی مراحل مهمی در AIS به شمار می‌روند. ابرجهش سوماتیک که قسمتی از بلوغ پیوندی است، می‌تواند قسمت مهمی از سیستم ایمنی مصنوعی باشد. ابرجهش سوماتیک در قسمت کد ژنتیکی که مربوط به شکل پذیرنده‌ی آنتی‌بادی است، رخ می‌دهد. جهش ممکن است میل پیوندی آنتی‌بادی را افزایش یا کاهش دهد. سلولB تنها در صورتی در این مرحله به تکثیر ادامه می‌دهد که جهش آن باعث افزایش میل پیوندی آن شده باشد و بنابراین آنتی‌بادی مداوماً توسط آنتی‌ژن‌ها تحریک شود. یک مکانیزم بازخورد مثبت وجود دارد که سعی در اعمال فشار انتخاب شدید، برای ایجاد سلول‌های B بهتر دارد. به این فرآیند بلوغ پیوندی می‌گویند. قدم بعدی فرآیند بلوغ پیوندی انتخاب بین حافظه یا پلاسما شدن سلول B است. سلول‌های پلاسما، سازندگان آنتی‌بادی هستند و در حجم بسیار زیادی آنتی‌بادی ترشح می‌کنند و بنابراین عمر زیادی نیز ندارند. حالت دیگر تبدیل شدن سلول B به سلول حافظه است. بیشتر سلول‌های حافظه، دارای میل پیوندی بالاتر از میانگین با آنتی‌ژن مربوطه دارند و بنابراین

کاندیدای اصلی به خاطر سپردن این آنتی‌ژن برای آینده هستند. این سلول‌های حافظه به‌علاوه‌ی تنظیم شدن برای یک نوع خاص آنتی‌ژن، در آینده برای فعال شدن احتیاج به تحریک کمتری دارند بنابراین سرعت و کارایی پاسخ ایمنی را وقتی پاتوژن برای بار دوم به بدن حمله می‌کند، زیاد می‌کند (شکل ۳-۱).

۳-۳-۲-۲ شبکه‌های ایمنی و حافظه

در این بخش به تئوری شبکه‌ی ایمنی خواهیم پرداخت. در این تئوری، که توسط جرن[۱] در سال ۱۹۷۴ [۹۵] پیشنهاد شد، سیستم ایمنی، سیستمی پویاتر در نظر گرفته شده است. تئوری شبکه‌ی ایمنی (یا شبکه‌ی ایدیوتیپیک) پیشنهاد می‌کند که سیستم ایمنی حتی در غیاب محرک دارای رفتاری پویا است.



شکل ۳-۴ آناتومی یک آنتی‌بادی بر اساس تئوری شبکه‌ی ایمنی جرن. در این شکل آنتی‌بادیB با اتصال از طریق پاراتوپش به ایدیوتوپ آنتی‌بادیA تحریک می‌شود. آنتی‌بادی A با این عمل سرکوب (تحریک منفی) نیز می‌شود.

جرن این فرضیه را ایجاد کرد که در سیستم ایمنی، هر مولکول آنتی‌بادی می‌تواند توسط مجموعه‌ای از مولکول‌های آنتی‌بادی دیگر تشخیص داده شود. برای توضیح این مسئله، جرن فرض کرد که هر آنتی‌بادی شامل دو ناحیه‌ی به نام‌های پاراتوپ[۲] و ایدئوتوپ[۱] است (شکل ۳-۴). این مناطق لزوماً

---

[۱] Jern

[۲] Paratope

دارای فرم یکسان نیستند اما ایدیوتوپ باید الگویی را که توسط آنتی‌ژن بیان می‌شود داشته باشد. بنابراین یک آنتی‌بادی با اتصال پاراتوپش به ایدوتوپ مکملش روی یک آنتی‌بادی دیگر تحریک می‌شود. تحریکی که بر اثر این اتصال به‌وجود می‌آید باعث تکثیر آنتی‌بادی می‌شود و فرزندانی با پذیرنده‌ی مشابه به‌وجود می‌آیند و در صورتی‌که سلول‌های والد بمیرند، اطلاعات آن‌ها از بین نمی‌رود. برعکس این مسئله هم صادق است یعنی در صورتی‌که آنتی‌بادی از طریق ایدیوتوپش به آنتی‌بادی دیگری متصل شود، سرکوب یا تحریک منفی می‌شود. بنابراین بر اساس این تئوری، به سیستم ایمنی مانند یک شبکه‌ی به‌هم متصل از سلول‌ها نگریسته می‌شود که یکدیگر را تحریک و سرکوب می‌کنند تا حافظه‌ی ایمنی ایجاد کنند. استفاده از این تئوری در سیستم ایمنی مصنوعی به تولید سیستم‌هایی چون [۹۶] منجر شده است.

۳-۳-۲-۳ تئوری خطر: فعال شدن بر اثر دو سیگنال

تئوری خطر در سال‌های اخیر توسط متزینگر[۲] [۹۷] [۹۸] ارائه شده است. این تئوری طبیعت و عملکرد پاسخ ایمنی را به طریقی متفاوت از دید کلاسیک توضیح می‌دهد. در دید کلاسیک تشخیص خودی و غیرخودی چنین توجیه می‌شود که سلول‌های ایمنی نمی‌توانند به میزبان خود حمله کنند زیرا در فرآیند بلوغ سلول‌هایی که در برابر سلول‌های خودی فعال می‌شوند بر طبق انتخاب منفی حذف می‌شوند. ولی به این توجیه ایراداتی وارد است زیرا مشاهدات نشان می‌دهد که شاید در برخی مواقع لازم باشد بدن به خودش حمله کند و یا سیستم ایمنی لازم باشد به سلول‌هایی که می‌داند خودی نیستند، حمله نکند. برای مثال موارد زیر که توسط متزینگر به عنوان مشاهدات نقض کننده‌ی توجیه کلاسیک، بیان شده است:

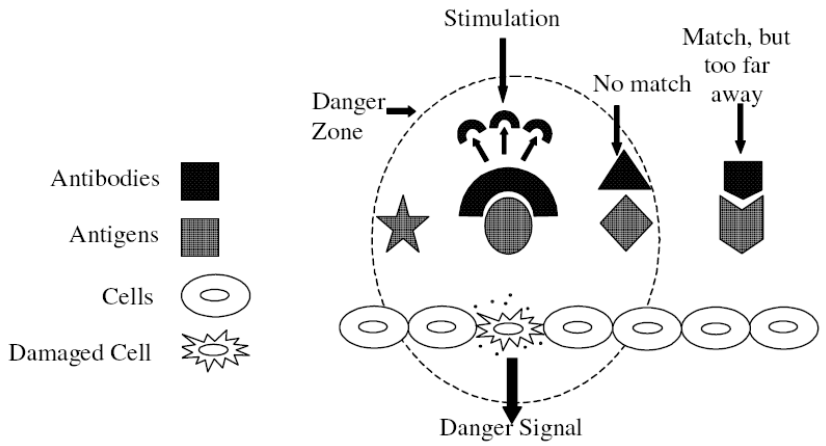
---

[۱] Idiotope

[۲] Metzinger

- واکنش ایمنی نسبت به باکتری‌های موجود در روده و غذای انسان، ایجاد نمی‌شود در صورتیکه هر دو موجودیت‌های بیگانه (غیرخودی) هستند.
- بدن انسان در طول عمر، در حال تغییر است و بنابراین سلول خودی نیز تغییر می‌کند. سوالی که مطرح می‌شود اینست که آیا دفاع علیه غیرخودی که قبلاً یادگرفته شده است ممکن است در آینده واکنش در برابر سلول خودی شود؟
- تناقض دیگر، بیماری‌های خودایمنی هستند و انواع مشخصی از تومورها که سیستم ایمنی با آنها مقابله می‌کند (که هر دوی موارد مقابله با خودی است) و پیوندهای موفقیت آمیز که نمونه‌ای از حمله نکردن سیستم ایمنی علیه غیرخودی است.

متزینگر می‌گوید که یک راه محتمل دیگر برای توضیح تحریک سیستم ایمنی برای پاسخ، واکنش به تحریکی باشد که بدن آن را مضر تشخیص می‌دهد. بنابراین ایده‌ی اصلی در این تئوری اینست که سیستم ایمنی به غیرخودی واکنش نشان نمی‌دهد بلکه به خطر پاسخ می‌دهد. از نظر مفهومی این نظریه تغییر بسیار کمی ایجاد می‌کند اما یک الگوی کاملا جدید وارد حوزه‌ی ایمونولوژی می‌شود [۹۹]. این مدل به سلولهای ایمنی و غیرخودی اجازه می‌دهد در کنار هم حضور داشته باشند، حالتی که در دید کلاسیک وجود ندارد. در هر حال، زمانی‌که حمله‌ای صورت گیرد، سلول‌هایی که غیرطبیعی می‌میرند سیگنالی به نام سیگنال خطر ایجاد می‌کنند [۱۰۰]، که در محدوده‌ی کوچکی حول سلول پراکنده می‌شود و به این محدوده، محدوده‌ی خطر گفته می‌شود. فقط در این ناحیه است که سیستم ایمنی فعال می‌شود و در برابر آنتی‌ژن‌هایی که در این ناحیه هستند واکنش نشان می‌دهد.



شکل ۳-۵ مدل تئوری خطر

یک راه دیگر نگاه کردن به تئوری خطر، نگرش مدل دو سیگناله است که توسط برتچر[۱] و کوهن[۲] [۱۰۱] ارائه شده است. در این مدل، دو سیگنال، یکی تشخیص آنتی‌ژن (سیگنال یک) و دیگری کمک تحریک[۳] (سیگنال دوم) است.

کمک سیگنال، سیگنالی است که چنین مفهومی دارد: این آنتی‌ژن حقیقتاً بیگانه است، یا در تئوری خطر، این آنتی‌ژن حقیقتاً خطرناک است. اگر به تئوری خطر این چنین بنگریم، می‌توانیم عملکرد این تئوری را تحت سه قانون زیر عنوان کنیم [۱۰۲]:

- قانون ۱: اگر سیگنال یک و دو با هم دریافت شدند، فعال شو. اگر سیگنال یک را در غیاب سیگنال دو دریافت کردی، بمیر. اگر سیگنال دو را بدون سیگنال یک دریافت کردی، به سیگنال دریافتی توجه نکن.
- قانون ۲: سیگنال دو را فقط از سلول‌های عرضه کننده‌ی آنتی‌ژن قبول کن.
- قانون ۳: بعد از فعال شدن (سلول‌های فعال شده احتیاج به سیگنال دو ندارند)، بعد از مدت کوتاهی به حالت معمول تغییر حالت بده.

---

[۱] Bertcher

[۲] Kohen

[۳] Co-Stimulation

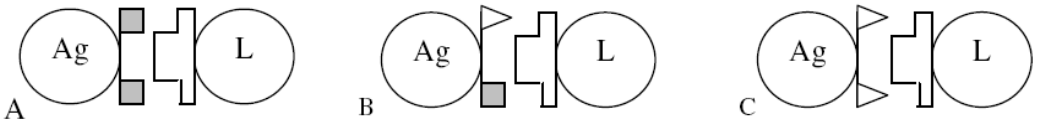
وارد توضیحات جزئی‌تر در این زمینه نخواهیم شد و در بخش سیستم ایمنی مصنوعی به کاربردهای این تئوری در سیستمهای ایمنی مصنوعی خواهیم پرداخت.

۳-۳-۳ ایمنی اکتسابی و مهندسی

در بخش‌های قبل، اصول ایمنی مورد نیاز برای درک AIS مرور شدند. درک راه استفاده از اصول ایمنی در سناریوهای مهندسی با توجه به اطلاعات داده شده در متن کار سختی است. بنابراین در دو سه پاراگراف بعدی به صحبت در این مقوله می‌پردازیم.

مهمترین مفهوم در سیستم ایمنی اینست که لنفوسیت ممکن است از طریق مولکول پذیرنده‌ای که در سطح خود دارد و دارای شکل خاصی است به سلول دیگری متصل شود. مولکول‌های پذیرنده‌ی سطحی دارای فرم‌های متفاوتی هستند و بسته به فرمی که دارند به الگوهایی با فرم مکمل با خود متصل می‌شوند و قدرت این اتصالات نیز با یکدیگر متفاوت است. اگر هماهنگی زیاد یا مکملی با درجه‌ی بالا بین شکل مولکول‌های پذیرنده‌ی سطحی و آنتی‌ژن وجود داشته باشد، قدرت اتصال بین دو سلول، بالا خواهد بود. از همین مفهوم، قدرت تشخیص الگوی سیستم ایمنی که در AIS دارای اهمیت بسیاری است به‌وجود می‌آید. این مفهوم در شکل ۳-۶ نشان داده شده است.

این مکانیزم تشخیص الگو به همراه مکانیزم‌های دیگر موجود در سیستم ایمنی مانند انتخاب کلونی و شبکه‌ی ایمنی برای حل مسائل استفاده می‌شوند.



شکل ۳-۶ قدرت اتصال بین لنفوسیت (L) و آنتی‌ژن (Ag)

۳-۴ سیستم ایمنی مصنوعی

سیستم ایمنی مصنوعی نوعی الگو برای یادگیری ماشین است. یادگیری ماشین، توانایی کامپیوتر برای انجام یک کار با یادگیری داده‌ها یا از روی تجربه است. سیستم ایمنی مصنوعی در [۳] به‌صورت زیر تعریف شده است:

سیستم‌های وفقی که با الهام از ایمونولوژی نظری و توابع، اصول و مدل‌های ایمنی مشاهده شده به‌وجود آمده‌اند و برای حل مسائل مورد استفاده قرار می‌گیرند.

دی‌کاسترو[۱] و تیمیس[۲] تعریف بالا را برای AIS برگزیده‌اند و سه نکته را برشمردند که در هر الگوریتم ایمنی مصنوعی باید لحاظ شود:

۱- در هر الگوریتم ایمنی مصنوعی، حداقل باید یک جزء ایمنی مانند لنفوسیت‌ها وجود داشته باشد.

۲- در هر الگوریتم ایمنی مصنوعی باید ایده‌ای برگرفته از بیولوژی نظری یا تجربی استفاده شود.

۳- الگوریتم ایمنی مصنوعی طراحی شده باید به حل مسئله‌ای کمک کند.

بر اساس این سه ضابطه، دی‌کاسترو و تیمیس، اولین الگوریتم‌های ایمنی مصنوعی را در سال ۱۹۸۶ طراحی کردند. در همان سال فارمر[۳] [۱۰۳] مدلی برای تئوری شبکه‌ی ایمنی ارائه کرد و بر اساس این مدل اعلام کرد که "سیستم ایمنی قادر به یادگیری، به خاطر سپردن و تشخیص الگوست." بعد از ادعای فارمر، توجه به AIS به عنوان یک مکانیزم یادگیری ماشین شروع شد. بعد از آن به تدریج AIS، در زمینه‌های مختلف وفق‌پذیر و جذاب بودن خود را نشان داد. سیستم ایمنی علاوه بر توانایی تشخیص الگو، صفات دیگری از قبیل یادگیری، حافظه، خود سازماندهی و از منظر مهندسی، خصوصیاتی دیگری مانند تشخیص بی‌قاعدگی، تحمل خطا، توزیع پذیری و مقاومت بالا نیز دارد که در صورتی‌که AIS به‌طور صحیح ایجاد شود، AIS هم دارای این ویژگی‌ها خواهد بود.

---

[۱] De Castro

[۲] Timmis

[۳] Farmer

۳-۵ چهارچوب سیستم ایمنی مصنوعی

دی‌کاسترو و تیمیس [۳] [۱۰۴]، یک چهارچوب مهندسی برای سیستم ایمنی مصنوعی ارائه کرده‌اند. هسته‌ی این چهارچوب برای مهندسی سیستم ایمنی مصنوعی شامل سه عامل است:

۱- فرم نمایش اجزاء سیستم.

۲- مجموعه‌ای از مکانیزم‌ها برای سنجش ارتباطات بین اجزاء با محیط و یکدیگر (یک تابع شباهت).

۳- شیوه‌ی سازگاری

این شیوه‌ی نگاشت یک مسئله‌ی مهندسی به AIS یک رویکرد لایه لایه است. بر اساس دامنه‌ی مسئله‌ی مورد بحث، نوع نمایش و به تبع آن انتخاب تابع پیوند تعیین می‌شود. ما نیز از این چهارچوب برای نگاشت مسئله‌ی خود به AIS استفاده خواهیم کرد. بنابراین در بخش بعد به توضیح لایه‌های مختلف این چهارچوب می‌پردازیم.

۳-۵-۱ طرز نمایش اجزاء

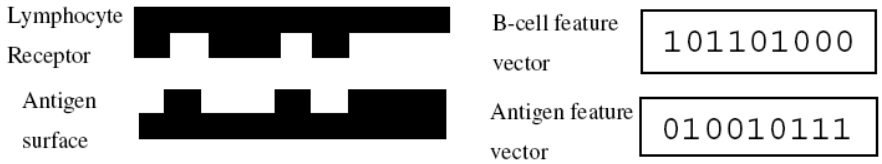
برای ایجاد یک AIS مناسب یکی از مهم‌ترین موارد، انتخاب بهترین نوع نمایش داده است. این نمایش در حقیقت شبیه‌سازی پذیرنده‌ی سلول در بیولوژی است که در AIS شکل این پذیرنده‌ی سلول با مجموعه‌ی ویژگی‌های داده تعیین می‌شود و AIS با تغییر مجموعه‌ای از این لنفوسیت‌های مصنوعی به‌صورت تکاملی درگیر است.

همان‌طور که توضیح داده شد، سیستم ایمنی طبیعی شامل سلول‌های ایمنی است که این سلول‌های ایمنی دارای پذیرنده در سطح خود هستند. مانند سلول B که روی سطح خود دارای آنتی‌بادی و سلول T که دارای پذیرنده‌ی TCR است. فرم این پذیرنده‌ها را اطلاعات ژنتیکی آن‌ها می‌سازد. فرم یک سلول ایمنی مصنوعی را نیز اطلاعات آن می‌سازد، به این‌صورت که ویژگی‌ها یا صفات داده در یک بردار کدگذاری می‌شوند (شکل ۳-۷). در سیستم ایمنی مصنوعی، فرم کدگذاری آنتی‌ژن و

آنتی‌بادی یکسان است و بنابراین می‌توان یک معیار شباهت یا فاصله بین آنها محاسبه کرد. در مورد شکل ۳-۷، هر دو آنتی‌ژن و آنتی‌بادی به‌صورت باینری تعریف شده‌اند.

هر سلول مصنوعی، نماینده‌ی یک نقطه در فضای جواب یا فضای جستجو است؛ که در زیست‌شناسی به این فضا، فضای تجسمی[۱] می‌گویند. مفهوم فضای تجسمی توسط پرلسون[۲] و استر[۳] [۱۰۵] ایجاد شد. پرلسون و استر فرم یک آنتی‌بادی m را به‌صورت مجموعه‌ای از L پارامتر فیزیکی (طول، پهنا، قد، شارژ الکتریکی و ... در مکان اتصال) تعریف کردند. بنابراین یک نقطه در فضای L بعدی نشان‌دهنده‌ی فرم پذیرنده یا همان آنتی‌بادی است.

از دید ریاضیات، شکل عمومی یک مولکول(m)، یک آنتی‌بادی(Ab) یا آنتی‌ژن(Ag)، می‌تواند با استفاده از یک مجموعه از اعداد حقیقی $< m = m_1, m_2, ..., m_l >$ نمایش داده شود، که نماینده یک نقطه در فضای L بعدی حقیقی است ($m \in S^L \subseteq R^L$ ، S نماد فضای تجسمی و L بعد آن است). اگر سیستم ایمنی N آنتی‌بادی داشته باشد، فضای تجسمی S آن سیستم ایمنی شامل N نقطه خواهد بود. می‌توان انتظار داشت که این نقاط در یک حجم محدود از فضا V قرار گیرند، زیرا تنها هر ویژگی حدی بالا و پایینی دارد و مقادیر ویژگی در بازه‌ی محدودی قرار می‌گیرد. برای اینکه آنتی‌بادی بتواند به الگویی متصل شود، باید نواحی مکملی با آن الگو داشته باشد، ولی این هماهنگی لازم نیست کامل باشد.



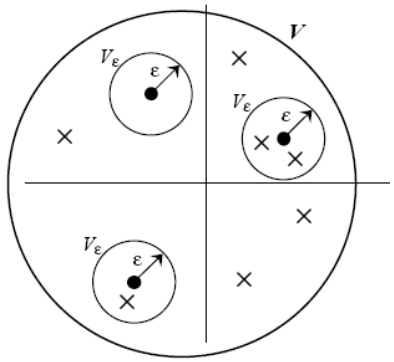
شکل ۳-۷ شباهت بین پذیرنده‌ی سلول B با بردار ویژگی سلول ایمنی مصنوعی. بردار ویژگی سلول‌های مصنوعی، برای مثال می‌توانند بولین باشند و نشان دهنده‌ی وجود یا عدم وجود یک کلمه در یک سند.

---

[۱] Shape space

[۲] Perelson

[۳] Oster



شکل ۳-۸. درون فضای تجسمی S، فضایV وجود دارد که در آن آنتی‌بادی (•) و آنتی‌ژن (×) قرار گرفته‌اند. فرض بر اینست که یک آنتی‌بادی همه‌ی آنتی‌ژن‌هایی که مکمل آن‌هاست و درون فضای V_ε قرار دارند را می تواند شناسایی کند.

مفهوم فضای تجسمی در AIS مترادف فضای جواب است. مجموعه‌ای از L پارامتر پتانسیل جواب بودن برای یک مسئله‌ی محاسباتی دارد. بنابراین می‌توان آنتی‌بادی را به‌صورت $Ab =< Ab, Ab_{2,...,} Ab_L >$ برای نشان دادن یک نقطه در فضا توصیف کرد. به این نوع نمایش بردار ویژگی یا رشته‌ی صفات گفته می‌شود. بر اساس مسئله‌ی مورد نظر آنتی‌بادی ممکن است مجموعه‌ای از

- اعداد (اعداد حقیقی، صحیح یا ...)
- سمبول‌ها، که از الفبای محدودی ایجاد شده باشند مانند اعداد باینری.

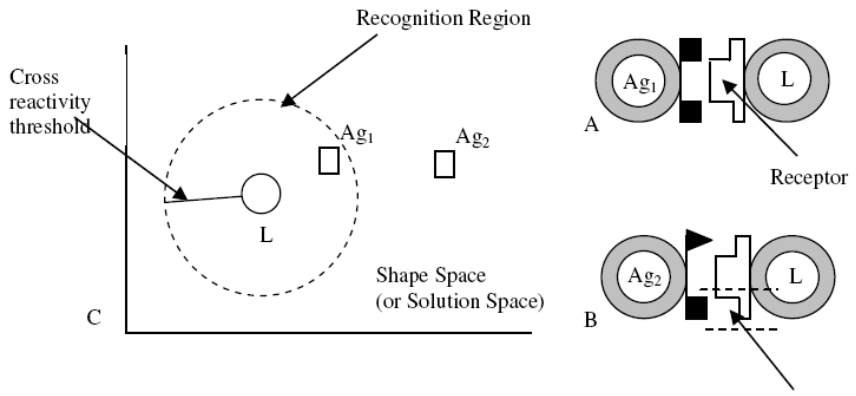
باشد.

۳-۵-۲ میل پیوندی

معیار پیوند مصنوعی از مفهوم اتصال بین پذیرنده‌ی لنفوسیت به الگوی آنتی‌ژنی برای تعیین درجه‌ی هماهنگی بین داده و نمونه‌ی کاندیدای جواب به وجود آمده است. برای اتصال به یک الگوی آنتی‌ژنی لازم است یک پذیرنده‌ی طبیعی قسمت اعظمی از بردار ویژگی‌اش با آنتی‌ژن هماهنگ باشد.

فرض بر اینست که هر آنتی‌بادی همه‌ی آنتی‌ژن‌هایی که در محدوده‌ی کوچک اطرافش هستند را می‌بیند. شعاع این محدوده با پارامتر ε مشخص می‌شود و به این منطقه، منطقه تشخیص می‌گویند

که دارای حجم V_ε است. الگوهای آنتی‌ژنی مشابه نواحی همسایه در فضای تجسمی را اشغال می‌کنند و بسته به مقدار ε ممکن است توسط یک فرم آنتی‌بادی شناسایی شوند. یک تابع ریاضیاتی باید تعریف شود تا تعیین کند که آیا دو آیتم در منطقه‌ی تشخیص یکدیگر قرار دارند یا خیر و اگر قرار دارند، میزان هماهنگی آن‌ها با یکدیگر چقدر است یا به عبارت دیگر با چه نیرویی به یکدیگر پیوند خورده‌اند. به این تابع معمولاً معیار فاصله یا تابع پیوند می‌گویند و این تابع معیار شباهتی بین دو سلول محاسبه می‌کند. تابع پیوند، نگاشتی از دو رشته‌ی صفات (بردار ویژگی) به عددی ایجاد می‌کند که آن عدد نشان‌دهنده‌ی درجه‌ی هماهنگی بین دو رشته است. شکل ۳-۸ نشان‌دهنده‌ی توضیحات فوق است.



شکل ۳-۹ منطقه‌ی تشخیص و آستانه‌ی فعالیت ضربدری. (A) نشان‌دهنده‌ی لنفوسیت (L) در پیوند مستحکمی با آنتی‌ژن (Ag1) است. آنتی‌ژن (Ag2) شباهت کمتری با لنفوسیت (L) دارد.

می‌توان به میل پیوندی به عنوان یک معیار کلی نگریست که کیفیت یک سلول در مجموعه سلول‌های مصنوعی را نسبت به الگوی خارجی مشخص می‌کند. اگر مقداری که توسط تابع پیوند محاسبه می‌شود بزرگ‌تر از حد آستانه‌ی ε باشد، گفته می‌شود آنتی‌ژن درون منطقه‌ی تشخیص سلول ایمنی قرار دارد یا به عبارت دیگر لنفوسیت می‌تواند آنتی‌ژن را شناسایی کند. اگر مقدار این آستانه مشخص باشد و به کمک رابطه‌ی (۳-۱) محاسبه‌ی تعداد کمینه‌ی آنتی‌بادی‌های مورد نیاز برای پوشش کامل فضای تجسمی امکان‌پذیر خواهد بود.

$$N = k^L \qquad (۳-۱)$$

که N سایز آنتی‌بادی‌های مورد نیاز، k سایز الفبای مورد استفاده (در فضای همینگ $k=2$) و L طول رشته‌ی صفات است.

اگرچه در سیستم ایمنی طبیعی، سلول‌هایی که فرم مکمل دارند به هم متصل می‌شوند ولی در AIS، معمولا پیوند میان دو رشته صفات متناسب با شباهت بین آن دو رشته خواهد بود.

بنابراین قدرت پیوندی بین یک آنتی‌ژن و یک آنتی‌بادی بستگی به فاصله آن‌ها دارد که می‌توان این فاصله را از طریق معیارهای فاصله‌ی بین دو رشته (دو بردار) محاسبه کرد، اگر مختصات یک آنتی‌بادی $< ab, ab_{2,...,} ab_L >$ و مختصات یک آنتی‌ژن $< ag_1, ag_2, ..., ag_L >$ باشد، و صفات به‌صورت عددی باشند می‌توان از فاصله‌ی (D) اقلیدسی (رابطه‌ی (۳-۲)) یا فاصله‌ی منهتن(رابطه‌ی (۳-۳)) بین آن‌ها استفاده کرد .

$$D = \sqrt{\sum_{i=1}^{L} (ab_i - ag_i)^2} \qquad (۳-۲)$$

$$D = \sum_{i=1}^{L} \left| ab_i - ag_i \right| \qquad (۳-۳)$$

یک معیار فاصله‌ی دیگر فاصله‌ی همینگ است. فاصله‌ی همینگ تعداد مکان‌هایی است که در دو رشته‌ی با طول برابر دارای مقادیر متفاوت هستند. رابطه‌ی (۳-۴) معیار فاصله همینگ را نشان می‌دهد.

$$D = \sum_{i=1}^{L} \delta = \begin{cases} 1(ab_i \neq ag_i) \\ 0(othrwise) \end{cases} \qquad (۳-۴)$$

در این بخش به معیارهای شباهت و تفاوت بین دو سلول آنتی‌بادی و آنتی‌ژن پرداختیم. در بخش بعد به فرآیندهایی ارتباطی بین سلول‌های مصنوعی خواهیم پرداخت.

۳-۵-۳ پردازش‌ها

بعد از ارائه‌ی طرز نمایش سلول‌ها و ایجاد مفهوم شباهت بین لنفوسیت‌ها و آنتی‌ژن‌ها، اکنون نوبت به بحث در مورد فرآیندهایی می‌رسد که جمعیت لنفوسیت‌ها را به روش‌های متفاوت تغییر می‌دهند. این فرآیندها از متدهای طبیعی تغییر جمعیت لنفوسیت‌ها در طی زمان اقتباس شده‌اند و بر اساس این

متدها، اصولی که بر اساس آن‌ها جمعیت سلول‌های مصنوعی نسبت به شرایط متغیر و تحریکات موجود، تغییر و تحول پیدا می‌کنند، شکل می‌گیرد. بنابر همین فرآیندهای ارتباطی بین سلول‌هاست که صفاتی چون یادگیری و تطابق‌پذیری در AIS به‌وجود می‌آیند.

الگوریتم‌های AIS متنوعی از روی فرآیندهای طبیعی ایجاد شده‌اند که طراحان AIS بسته به نوع مسئله‌ی پیش‌رو، بهترین فرآیند را انتخاب می‌کنند. این فرآیندها در ۵ دسته‌ی زیر قرار می‌گیرند:

۱- مغز استخوان

۲- انتخاب منفی

۳- شبکه‌ی ایمنی

۴- انتخاب کلونی

۵- سیگنال خطر

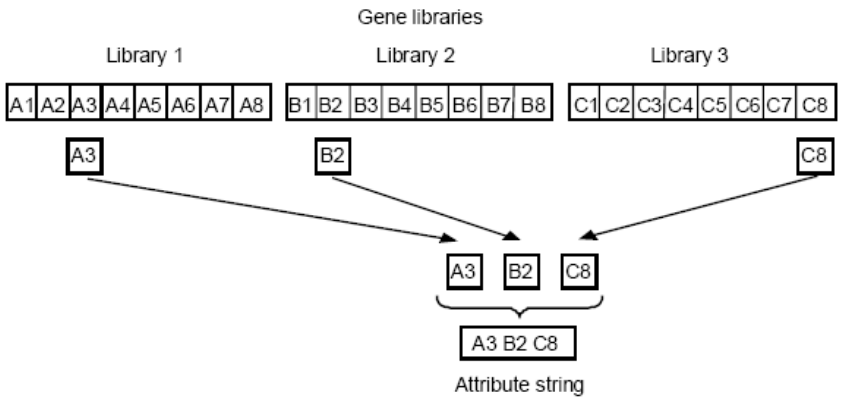
برخی از این الگوریتم‌ها با یکدیگر همپوشانی دارند. برای مثال در شبکه‌ی ایمنی از فرآیند کلون کردن استفاده می‌شود که اصل انتخاب کلونی است. در همه‌ی مدل‌ها لازم است مجموعه‌ی اولیه‌ای از سلول‌ها ایجاد شود که بنابراین در همه‌ی مدل‌ها از مدل مغز استخوان استفاده می‌شود.

در بخش‌های بعد به توضیح فرآیندهای مغز استخوان، شبکه‌ی ایمنی، انتخاب کلونی و سیگنال خطر خواهیم پرداخت.

۳-۵-۳-۱ مغز استخوان

سلول‌ها و مولکول‌های ایمنی در مغز استخوان تولید می‌شوند. ژن‌هایی که برای رمزکردن این مولکول‌ها استفاده می‌شوند در کتابخانه‌های مجزایی ذخیره شده‌اند. رمزکردن این مولکول‌ها از طریق اتصال قطعات ژنی مختلف که به‌صورت تصادفی از هر کدام از کتابخانه‌ها انتخاب می‌شوند، صورت می‌گیرد. مدل‌های مغز استخوان برای ایجاد رشته‌های صفات که نماینده‌ی پذیرنده‌های ایمنی هستند استفاده می‌شوند. توجه کنید که تا به حال، تفاوتی بین سلول و پذیرنده‌ی آن قائل نشدیم، هر دوی آنها توسط رشته صفات نشان داده می‌شوند.

ساده‌ترین مدل مغز استخوان، مدلی است که با استفاده از تولیدکننده‌ی عدد تصادفی، رشته‌ی صفات با طول L در S^L ایجاد می‌کند. در حالت فضای تجسمی تصادفی، کافیست بازه‌ای که m در آن تعریف شده‌است را تعریف کنیم. مثلا $m \in [0,1]^L$. در حالت فضای تجسمی همینگ، رشته‌ای که m را نشان می‌دهد باید از المان‌هایی تشکیل شود که به الفبای از قبل تعیین شده‌ای متعلق باشد. مثلاً $m \in \{0,1\}$ برای رشته‌های باینری.



شکل ۳-۱۰ فرآیند ایجاد یک مولکول آنتی‌بادی از ترکیب قطعات ژنی کتابخانه‌های ژنی. یک جزء از هر کتابخانه انتخاب و با قطعات دیگر الحاق شده و رشته‌ی صفات را که نماینده پذیرنده ایمنی است، می‌سازد.

از الگوریتم‌های مغز استخوان در AIS، برای مقداردهی اولیه‌ی جمعیت یا برای جایگزینی سلول‌هایی که به هر دلیلی از جمعیت حذف شده‌اند استفاده می‌شود. بر خلاف الگوریتم‌های انتخاب منفی، انتخاب کلونی و شبکه‌ی ایمنی این الگوریتم، کل جمعیت را تغییر نمی‌دهد. هدف چنین فرآیندی تولید بردارهای ویژگی تصادفی است. دو محدودیتی که معمولاً برای تولید این بردارها اعمال می‌شود عبارتند از:

۱- طول بردار باید مجاز باشد.

۲- عناصر بردار باید در محدوده‌ی الفبای تعریف شده یا محدوده‌ی مقادیر مجاز باشد.

الگوریتم مغز استخوان به‌صورت بالا، ساده‌ترین الگوریتم مغز استخوان است.

۳-۵-۳-۲ شبکه‌ی ایمنی

در بخش ۳-۳-۲-۳ به توضیح تئوری شبکه‌ی ایمنی پرداختیم. شبکه‌ی ایمنی مصنوعی (AIN) یک مدل محاسباتی الهام گرفته شده از محیط زیستی است که از ایده‌ها و مفاهیم تئوری شبکه‌ی ایمنی که ارتباطات بین سلول‌های B (تحریک و سرکوب یکدیگر)، تکثیر و جهش است، استفاده می‌کنند. مدل‌های متفاوت شبکه‌ی ایمنی برای حل مسائل در زمینه‌های مختلف از قبیل تحلیل داده، تشخیص الگو و بهینه‌سازی طراحی شده است. در این بخش به توضیح شبکه‌ی ایمنی مصنوعی کلی خواهیم پرداخت.

شبه کد الگوریتم شبکه‌ی ایمنی در شکل ۳-۱۰ نشان داده شده است.

الگوریتم، مجموعه‌ای از آنتی‌ژن‌ها را به عنوان ورودی دریافت کرده (مجموعه‌ی A) و خروجی الگوریتم، شبکه‌ی ایمنی متشکل از مجموعه‌ای از سلول‌های B و ارتباطات بین آن‌ها خواهد بود.

اولین قدم ایجاد یک مجموعه‌ی اولیه از سلول‌های B است (مجموعه‌ی B)، بنابراین در این قسمت از الگوریتم شبکه‌ی ایمنی، از الگوریتم‌های مغز استخوان استفاده می‌شود. در برخی الگوریتم‌ها از زیر مجموعه‌ای از آنتی‌ژنها به عنوان مجموعه‌ی اولیه‌ی آنتی‌بادی استفاده می‌کنند و برخی مدل‌ها خود به‌طور تصادفی سلول‌های B اولیه را ایجاد می‌کنند.

در مرحله‌ی بعد یک فرآیند تکراری انجام می‌شود که این فرآیند با عرضه‌ی مجموعه‌ی آنتی‌ژن‌ها به شبکه شروع می‌شود. برای هر آنتی‌ژن و هر سلول B، میزان تحریک محاسبه می‌شود. میزان تحریک به‌صورت زیر (رابطه‌ی (۳-۵)) نشان داده می‌شود.

$$f^{A}_{stimulation} : A \times B \rightarrow \Re \qquad (۳-۵)$$

در بیشتر مدل‌ها، تحریک تابعی از معیار پیوند است. در چنین حالتی، تحریک به‌صورت زیر (رابطه‌ی (۳-۶)) خواهد بود.

$$f^{A}_{stimulation}(a,b) = g(f_{affinity}(a,b)) \qquad (۳-۶)$$

که $f_{affinity} : B \cup A \times B \cup A \to \Re$ و $g : \Re \to \Re$. $f_{affinity}$ شباهت بین سلول‌های ایمنی را در فضای تجسمی محاسبه می‌کند. g مقدار تحریکی را که به‌وسیله‌ی یک آنتی‌ژن ایجاد می‌شود محاسبه می‌کند.

در قدم بعد، سلول‌های B با یکدیگر ارتباط برقرار می‌کنند. این ارتباطات با محاسبه‌ی تاثیر تحریک و سرکوب آنتی‌بادی‌ها بر یکدیگر محاسبه می‌شود. این تاثیرها به‌صورت زیر نشان داده می‌شود.

$$f^{B}_{stimulation} : B \times B \to \Re \quad , \quad f^{B}_{\sup pression} : B \times B \to \Re \qquad (۷-۳)$$

مانند محاسبه‌ی تحریک آنتی‌ژن/سلولB، تحریک سلولB/سلولB (و سرکوب) نیز به‌صورت تابعی از پیوند سلولB/سلولB محاسبه می‌شود. کل تحریک $F : B \to \Re$ سلول B از جمع تاثیرات آنتی‌ژن و ارتباطات شبکه به‌دست می‌آید:

$$F(b) = \sum_{a \in A} f^{A}_{stimulation}(a,b) + \sum_{b' \in B, b' \notin b} f^{B}_{stimulation}(b',b) + \sum_{b' \in B, b' \notin b} f^{B}_{\sup pression}(b',b), b \in B \qquad (۸-۳)$$

بر اساس تحریک کلی، برخی از سلول‌های B انتخاب و $f_{cloning}(b)$ کپی از هر سلول B انتخاب شده، ایجاد می‌شود. این کپی‌ها با نرخ ثابتی تحت جهش قرار می‌گیرند. برخی از مدل‌ها از این نرخ به عنوان احتمال این‌که سلول B برای اعمال جهش بر آن انتخاب شود یا نه استفاده می‌کنند و برخی از این نرخ برای تعیین تعداد جهش‌هایی که باید بر سلول B اعمال شود، استفاده می‌کنند.

در مرحله‌ی متاداینامیک برخی از سلول‌های B که تحریک نشدند از شبکه حذف می‌شوند. سلول‌های B جدید به‌طور تصادفی تولید و به شبکه اضافه می‌شوند. در نهایت هنگامی که شرط خاتمه برقرار شد، شبکه‌ی به‌وجود آمده، جواب خواهد بود.

1: *initialization*
1.1: assign B an initial set of B-cells
1.2: initialize network structure L
2: repeat until a stop criterion is met
2.1: *antigen presentation:*
▷ *Antigen/B-cell affinity*
2.1.1: calculate $f_{affinity}(a,b)$ for all $a \in A, b \in B$
▷ *Antigen/B-cell stimulation*
2.1.2: calculate $f^{A}_{stimulation}(b,a)$ for all $a \in A$ and $b \in B$
2.2: *B-cell interaction:*
▷ *B-cell/B-cell stimulation/suppression*
2.2.1: calculate $f^{B}_{stimulation}(b',b)$ and $f^{B}_{suppression}(b',b)$ for all $b, b' \in B$
2.3: *affinity maturation:*
▷ *Total stimulation*
2.3.1: calculate $F(b) := \sum_{a \in A, b' \in B, b' \neq b} f^{A}_{stimulation}(a,b) + f^{B}_{stimulation}(b',b) + f^{B}_{suppression}(b',b),\ b \in B$
2.3.2: create $f_{cloning}(b)$ clones of the B-cell b and mutate them
2.3.3: calculate stimulation of all new B-cells
2.4: *metadynamics:*
▷ *deletion/creation of B-cells and links*
2.4.1: update network structure L
▷ *Return immune network*
3: return (B, L)

شکل ۳-۱۱ شبه‌کد الگوریتم شبکه‌ی ایمنی مصنوعی عمومی

۳-۵-۳-۳ انتخاب کلونی

با اقتباس انتخاب کلونی و به‌کار گرفتن آن در AIS، می‌توان برخی از خصوصیات سیستم ایمنی طبیعی از قبیل تطابق و یادگیری را در AIS به‌وجود آورد. به یاد می‌آورید که وظیفه‌ی سلول B این است که به آنتی‌ژن متصل شود تا زمانی‌که آن آنتی‌ژن توسط سلول‌های ایمنی دیگر از بین برود. ولی در مواقعی هیچ سلول B وجود ندارد که پیوندی قوی با آنتی‌ژن برقرار کند. بنابراین سلول B فعال شده، فرآیند تکثیر و جهش پذیرنده را شروع می‌کند، که به این فرآیند، فرآیند انتخاب کلونی گفته می‌شود. فشار انتخاب شدیدی در حین این فرآیند تکثیر باعث ماکزیمم کردن پیوند با آنتی‌ژن خواهد شد و بنابراین پاسخ ایمنی قوی‌تر می‌شود. در AIS یک سلول ایمنی فعال شده به همین صورت برای ایجاد پاسخ مناسب به یک داده‌ی جدید، تغییر پیدا می‌کند. بعد از فعال شدن، سلول مصنوعی شروع به تکثیر با نرخی متناسب با عکس مقدار پیوند با آنتی‌ژن می‌کند.

هدف هر دوی این فرآیندها، سوق دادن تدریجی سلول‌های جمعیت در فضای جواب به سمت آنتی‌ژن است. چنین فرآیند تطبیقی در الگوریتم‌های تکاملی، متداول است. بعد از فعالیت تعداد کمی از کلون‌ها که دارای پیوند زیاد با آنتی‌ژن‌ها هستند باقی‌مانده و به شکل سلول حافظه در می‌آیند.

باید به این نکته توجه شود که در سیستم طبیعی، جهش، شکل پذیرنده‌ی سلول B را تغییر می‌دهد و بنابراین حقیقت که تنها پذیرنده‌هایی که توانایی شناسایی آنتی‌ژن‌ها را دارند تکثیر می‌شوند، فشار انتخاب شدیدی بر جمعیت اعمال می‌شود.

به نظر می‌آید جهش در سلول‌های B هدفمند است. به این معنی که اگر هماهنگی بین آنتی‌ژن و پذیرنده بیشتر باشد، نرخ جهش بر روی آن پذیرنده کمتر خواهد بود. بنابراین سلول‌هایی که دارای پذیرنده‌های با میل پیوندی بالا هستند بیشتر تکثیر می‌شوند و بنابراین میانگین میل پیوندی جمعیت با آنتی‌ژن بالا می‌رود. به این فرآیند تکثیر و جهش، بلوغ پیوندی گفته می‌شود.

دو اصل در الگوریتم انتخاب کلونی مهم است:

۱- سلول‌های زیادی ممکن است برای تکثیر شدن متناسب با میزان پیوندشان انتخاب شوند.

۲- نرخ جهش هر کلون نسبت عکس با میزان پیوند آن با آنتی‌ژن دارد.

زمانی‌که تکثیر اتفاق می‌افتد، جهش هم بر روی کلون‌ها اتفاق می‌افتد. جهش مهم است زیرا در جمعیت تنوع ایجاد می‌کند و باعث جستجو برای جواب می‌شود. جهش احتمال اینکه جمعیت به یک نقطه‌ی بهینه‌ی محلی همگرا شود را کم می‌کند. توجه کنید که نرخ جهش در یک الگوریتم انتخاب کلونی نسبت عکس با میزان پیوند با آنتی‌ژن دارد. معمول‌ترین روش جهش، جهش تصادفی است. جهش تصادفی یک نقطه از بردار ویژگی را تصادفی انتخاب کرده و مقدار آن را با مقدار مجاز دیگری جایگزین می‌کند. نوع دیگر جهش، جهش وارونه[۱] نام دارد. در این نوع جهش، دو نقطه در بردار ویژگی تصادفی انتخاب می‌شود و مقادیر این دو نقطه با هم جابه‌جا می‌شوند.

---

[۱] Inversive

هنگام اعمال جهش تصادفی، باید به نوع نمایش آنتی‌بادی نیز توجه کرد. زیرا با توجه به الفبای مورد استفاده باید هنگام ایجاد جهش، از الفبای مجاز مقدار جدیدی جایگزین مقدار قبلی شود. هنگام ایجاد جهش در برداری که حاوی اعداد است، باید مقدار کمینه و بیشینه‌ی مجاز برای اعداد در نظر گرفته شود. بنابراین هنگام انجام جهش باید توجه داشت که بردار معتبر باقی بماند.

کُلونال‌جی[۱] الگوریتمی است که توسط دی‌کاسترو[۲] و وون زوبن[۳] [۹۴] طراحی شده است. این الگوریتم، یک الگوریتم عمومی انتخاب کلونی است که به‌طور گسترده توسط طراحان زیادی [۱۰۶] [۱۰۷] [۱۰۸] مورد استفاده قرار گرفته است. کُلونال‌جی برای کاربردهایی از قبیل تشخیص الگو و بهینه‌سازی مورد استفاده قرار می‌گیرد. به دلیل عمومی بودن و در برداشتن همه‌ی اصول الگوریتم انتخاب کلونی، شبه‌کد این الگوریتم را در شکل ۳-۱۱ برای آشنایی با انتخاب کلونی در این بخش ارائه می‌کنیم.

1. **Initialise**: Create a random population of individuals
2. **Antigenic Presentation**: For each antigenic pattern, do
   - 2.1. **Affinity Evaluation**: Present antigen to each member of population and determine affinity.
   - 2.2. **Clonal selection and expansion**: Select *n* highest affinity elements of population. Clone these with rates proportional to affinity.
   - 2.3. **Affinity maturation**: mutate all clones with rates inversely proportional to affinity and add them to population
   - 2.4. **Memory**: keep element of population with highest affinity to antigen
   - 2.5. **Metadynamics**: replace the *m* lowest affinity elements of population with new ones.
3. **Cycle**: Repeat step 2 until stopping criterion is met

شکل ۳-۱۲ الگوریتم کُلونال‌جی

---

[۱] CLONALG

[۲] De Castro

[۳] Von Zuben

۳-۵-۳-۴ تئوری خطر

به نظر می‌رسد تئوری خطر می‌تواند با تمرکز توجه روی وقایع داخلی یا خارجی که خطرناک هستند و باعث بروز مشکل در سیستم ایمنی می‌شوند، زیرمجموعه‌ی کوچکی(نسبت به مجموعه‌ی همه‌ی سلول‌های بیگانه‌ای که وارد سیستم ایمنی می‌شوند) از سلول‌های بیگانه را انتخاب و برای آن‌ها پاسخ ایمنی ایجاد کند. بنابراین با استفاده از این تئوری در سیستم‌های ایمنی مصنوعی می‌توان هزینه‌ی محاسباتی الگوریتم‌ها را تا حد زیادی کاهش داد.

سیگنال خطر، بسته به نوع مسئله تعاریف متفاوتی دارد. همان‌طور که در [۱۰۹] اشاره شده است خطر می‌تواند در سیستم‌های تشخیص بی‌قاعدگی، یک رفتار اشتباه در سیستم‌های تشخیص تقلب، یک داده‌ی جعلی، در داده‌کاوی یک داده‌ی جذاب و ... باشد.

از آنجا که سیگنال خطر اخیراً مورد توجه قرار گرفته است، الگوریتم‌های AIS زیادی از این تئوری بهره‌برداری نکرده‌اند. معمولا سیگنال خطر به همراه الگوریتم‌های انتخاب منفی، شبکه‌ی ایمنی یا انتخاب کلونی استفاده می‌شود.

اولین استفاده از این تئوری برای تشخیص خودی، غیرخودی بوده است. در [۱۰۹] ادعا شده است که در صورتی‌که در کاربردی احتیاج به تشخیص خودی از غیرخودی باشد، به دلیل مشکلات ذکر شده برای انتخاب منفی، استفاده از تئوری خطر سودمند خواهد بود. بعد از تصمیم به استفاده از تئوری خطر توجه به نکات زیر در استفاده از این تئوری به طراح کمک شایانی خواهد کرد:

- خطر، یک اصطلاح وابسته به کاربرد است و سیگنال شاید اصلاً ربطی به خطر نداشته باشد.
- سیگنال مناسب ممکن است مثبت یا منفی باشد.
- منطقه‌ی خطر در بیولوژی مکانی است. در کاربردهای سیستم ایمنی مصنوعی می‌تواند معیار دیگری جایگزین شود. برای مثال محدوده‌ی زمانی ممکن است، استفاده شود.

و ...

کاربردی که تئوری خطر در آن بسیار مورد استفاده قرار گرفته است، تشخیص بی‌قاعدگی است.

برای استفاده این تئوری خطر در شبکه‌ی ایمنی مصنوعی، الگوریتم مشخصی وجود ندارد، بسته به کاربرد باید مفهوم خطر در سیستم تعیین و سپس بر اساس این مفهوم، سیگنال خطر بر اثر ورود برخی آنتی‌ژنها آزاد شود. در برخی کاربردها، طراحان سعی کردند در سیستم، سلول‌های عرضه‌کننده‌ی آنتی‌ژن را تعریف کنند که نقش زیادی در ایجاد سیگنال خطر در سیستم ایمنی طبیعی دارند. در [۱۱۰] برای استفاده از تئوری خطر از سلول‌های دندریتیک که یک نوع از سلول‌های سیستم ایمنی ذاتی هستند، استفاده شده است. در بخش بعد به معدود کاربردهای تئوری خطر در داده‌کاوی خواهیم پرداخت.

۳-۶ سیستم ایمنی مصنوعی برای داده‌کاوی و کاربردهای دیگر

از زمان آغاز بحث AIS، این سیستم برای اهداف متنوعی به‌کار گرفته شده است [۱۱۱] [۱۱۲]. می‌توان اکثر این کاربردها را تحت سه عنوان بزرگ دسته‌بندی کرد. این سه عنوان عبارتند از:

۱- تشخیص بی‌قاعدگی

۲- بهینه‌سازی

۳- داده‌کاوی (کلاس‌بندی، خوشه‌بندی و ...)

هدف اصلی در این پروژه، داده‌کاوی است که در بخش‌های بعدی به این مقوله خواهیم پرداخت.

۳-۶-۱ داده‌کاوی با سیستم ایمنی مصنوعی

معمولاً برای کاربردهای داده‌کاوی، الگوریتم‌های مبتنی بر شبکه‌ی ایمنی مصنوعی و انتخاب کلونی مورد استفاده قرار می‌گیرند. این الگوریتم‌ها معمولاً شامل فرآیندهای هماهنگی الگو، به‌علاوه‌ی مکانیزم‌های یادگیری و تطابق‌پذیری طبیعی هستند. این ویژگی‌ها دست‌به‌دست هم داده و سیستم‌های داده‌کاوی قابل ملاحظه‌ای را به‌وجود می‌آورند.

۳-۶-۱-۱ راه‌حل‌های شبکه‌ی ایمنی

اولین کار در این زمینه توسط هانت[۱] و کوک[۲][۱۱۳] انجام شد. آن‌ها ادعا کردند برتری AIS در این است که یک شبکه‌ی غیرخطی تطابقی است که کنترل در آن غیرمتمرکز است و قادر به فراموش کردن اطلاعات کم اهمیت نیز می‌باشد. در الگوریتم آن‌ها، سلول‌های B، با استفاده از معادلات شبکه‌ی ایمنی فارمر تحریک و سرکوب می‌شوند.

اگر در شبکه، N آنتی‌بادی مختلف با تمرکز $\{c_1, c_2, ..., c_n\}$ و M آنتی‌ژن مختلف با تمرکز $\{y_1, y_2, ..., y_n\}$ وجود داشته باشد. نرخ تغییر تمرکز آنتی‌بادی با زمان، توسط مدل شبکه‌ی ایمنی فارمر از رابطه‌ی (۳-۹) به‌دست می‌آید.

$$\frac{dc_i}{dt} = k_1[\sum_{j=1}^{N} m_{j,i} c_i c_j - k_2 \sum_{j=1}^{N} m_{i,j} c_i c_j + \sum_{j=1}^{M} m_{j,i} c_i y_j] - k_3 c_i \qquad (۳-۹)$$

اولین عبارت، تحریک آنتی‌بادی i توسط آنتی‌بادی j، عبارت دوم، سرکوب آنتی‌بادی i توسط آنتی‌بادی j و سومین عبارت تاثیر تحریکی آنتی‌ژن بر شبکه و چهارمین عبارت نرخ ثابت مرگ سلول است. k_1، k_2 و k_3 ضرایب ثابت هستند.

این AIS برای اولین بار بر روی یک مسئله‌ی ساده‌ی تشخیص الگو، سنجیده شد و سپس برای کلاس‌بندی توالی‌های DNA تست شد که نتایج آن، شبیه یا حتی بهتر از رویکردهایی از قبیل نزدیک‌ترین همسایه[۳] و شبکه‌های عصبی بود.

به منظور اثبات کارایی الگوریتم‌های مبتنی بر شبکه‌ی ایمنی برای مسائل داده‌کاوی کلاسیک، نیل[۴] و همکاران [۱۱۴] الگوریتمی مبتنی بر شبکه‌ی ایمنی به نام جی‌سیس[۵] طراحی کرده و این الگوریتم را

---

[۱] Hunt

[۲] Cook

[۳] Nearest Neighbor

[۴] Neal

[۵] JISYS

برای تشخیص تقلب[۱] بکار بردند. این کار توسعه‌ای بر کار هانت و کوک بود و نتایج به‌دست آمده نشان داد که جی‌سیس می‌تواند همه‌ی الگوهای جعلی را در مجموعه‌ی داده‌های وام و رهن که برای تست الگوریتم مورد استفاده قرار گرفته بود، تشخیص دهد.

در [۱۱۵] نتایج استفاده از شبکه‌ی ایمنی برای یادگیری بدون نظارت ارائه شده است. در این الگوریتم از جمعیتی از سلول‌های B استفاده و تحریک هر یک از آن‌ها به‌وسیله‌ی میزان هماهنگی آن‌ها با دیگر سلول‌های شبکه و تحریک آنتی‌ژنیک محاسبه شده است. الگوریتم با استفاده از داده‌های فیشرآیریس[۲] تست شده است [۱۱۶]. اگرچه نتایج کلاس‌بندی خوب گزارش شده است ولی مشاهده شد که سایز جمعیت به‌صورت غیر قابل کنترلی رشد کرده و تنها سلول‌های مشابه در شبکه به‌هم متصل می‌شوند. این کار در [۱۱۷] توسعه داده شد و الگوریتمی به نام AINE به‌وجود آمد. در این الگوریتم مفهومی به نام ARB جایگزین سلول B شد. ARBها در این الگوریتم برای به‌دست آوردن منابع محدودی رقابت می‌کنند و ARBهایی که آخر هر حلقه‌ی تکرار الگوریتم بدون منبع می‌مانند از جمعیت حذف می‌شوند. در [۱۱۸] AINE با داده‌های محک دیگری تست شد. در مقایسه با AINE دی‌کاسترو و وون زوبن سیستمی مبتنی بر شبکه‌ی ایمنی به نام aiNet ارائه کردند. aiNet یک ابزار خوشه‌بندی بدون نظارت است. تعریف خود طراحان از aiNet به این شرح است: *aiNet یک گراف وزن‌دار ناهمبند است که از نودهایی به نام آنتی‌بادی و اتصالات بین نودها به نام لبه، که هر لبه دارای وزن یا نیروی اتصالات است، تشکیل شده است[۱۱۱]*. تحلیل پیچیدگی برای الگوریتم aiNet انجام شده و نشان داده شده است که مرتبه‌ی زمانی aiNet $O(m^2)$ است که m تعداد نهایی سلول‌های حافظه است. کارایی الگوریتم برای سه مسئله‌ی محک تست شده است و علی‌رغم نتایج خوب مشاهده شده، مقایسه‌ای با کارهای موجود انجام نشده است.

---

[۱] Fraud detection

[۲] Fisher Iris

بقیه‌ی کارهای انجام شده و الگوریتم‌های پیشنهادی مبتنی بر شبکه‌ی ایمنی، بر اصول الگوریتم‌های AINE و aiNet بنا شده‌اند. از این قبیل می‌توان به کار انجام شده توسط ناساروئی[۱] [۱۱۹] که بر مبنای AINE ایجاد شده اشاره کرد در این الگوریتم هر آنتی‌بادی شعاع تاثیر مخصوص به خود دارد و در هر مرحله، این شعاع تاثیر، به‌روز می‌شود.

۳-۶-۱-۲ راه حل‌های انتخاب کلونی

الگوریتم‌های انتخاب کلونی راه‌حل‌های رقابتی هستند که برای طیف گسترده‌ای از مسائل از قبیل بهینه‌سازی [۱۲۰] و مسئله‌ی فروشنده دوره‌گرد (TSP) [۹۴] مورد استفاده قرار گرفته‌اند. به علت طبیعت تکاملی این الگوریتم‌ها و بنابراین شباهت آن‌ها با الگوریتم‌های تکاملی، از الگوریتم‌های انتخاب کلونی بیشتر برای مسائل بهینه‌سازی استفاده شده است [۱۲۱] [۱۲۲] [۱۲۳].

انتخاب کلونی در زمینه‌ی داده‌کاوی مخصوصاً کلاس‌بندی نیز سودمندی خود را به اثبات رسانده است. یکی از الگوریتم‌های کلاس‌بندی مبتنی بر انتخاب کلونی، الگوریتمی به نام AIRS برای تشخیص الگو است [۱۲۴-۱۳۹] که در سال ۲۰۰۴ توسط واتکینز[۲] [۱۳۰] اصلاح شده است.

الگوریتم انتخاب کلونی برای داده‌کاوی از AINE که الگوریتم شبکه‌ی ایمنی است و در بخش پیش توضیح داده شد، الهام گرفته است [۱۳۱] [۱۳۲]. این الگوریتم، الگوریتمی مبتنی بر انتخاب کلونی برای خوشه‌بندی است.

الگوریتم ماریا[۳] که مبتنی بر انتخاب کلونی است برای غلبه بر مشکل بی‌ثباتی الگوریتم AINE به‌وجود آمده است. از آن‌جا که سیستم ایمنی ذاتاً چندین لایه است، ماریا نیز سه لایه‌ی سلول به شرح زیر دارد:

---

[۱] Nasroui

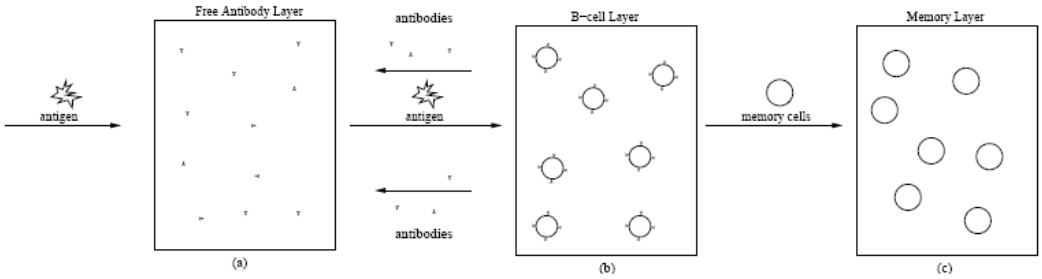
[۲] Watkins

[۳] MARIA

۱- لایه‌ی آنتی‌بادی آزاد: آنتی‌بادی‌های آزاد، آنتی‌بادی‌هایی هستند که در فضای تجسمی برای یافتن الگوی آنتی‌ژنی که به عنوان ورودی وارد سیستم می‌شود، جستجوی کلی انجام می‌دهند. هر آنتی‌بادی آزادی که میل پیوندی کافی با این الگو داشته باشد در لایه‌ی سلول B قرار می‌گیرد.

۲- لایه‌ی سلول B: در این لایه، جستجوی مخصوصی برای یافتن الگوی آنتی‌ژنی انجام می‌شود. نقش این لایه، یادگیری الگوی آنتی‌ژنی است. سلول‌ها در لایه‌ی سلول B در صورت تحریک شدن تحت بلوغ پیوندی قرار می‌گیرند. اگر آنتی‌ژن هیچ آنتی‌بادی‌ای را تحریک نکرد، یک پاسخ اولیه ایجاد می‌شود، به این‌صورت که داده‌ی درون آنتی‌ژن در یک سلول B کپی و به جمعیت سلول B اضافه می‌شود.

۳- لایه‌ی سلول حافظه: این لایه شامل الگوهایی است که در دو لایه‌ی قبل به‌وجود آمده‌اند. سلول‌هایی که در این لایه ذخیره می‌شوند تاثیری بر فرآیند یادگیری ندارند. این لایه خروجی ندارد.



شکل ۳-۱۳ جریان سلول‌ها از لایه‌های مختلف در الگوریتم ماریا

سلول‌ها در همه لایه‌ها دارای سن هستند و با گذر زمان سنشان زیاد می‌شود و زمانی‌که در تعداد مشخصی تکرار تحریک نشوند از جمعیت حذف می‌شوند. این مکانیزم کنترل جمعیت باعث عدم ازدیاد جمعیت و بطور همزمان باعث افزایش کیفیت جواب می‌شود؛ زیرا سلول‌هایی میل به بقا دارند که مفید هستند. تست‌های متعددی برای کارایی این الگوریتم اجرا شده و نشان داده شده است که الگوریتم به نرخ فشرده‌سازی داده‌ی خوبی می‌رسد و در عین حال این فشرده‌سازی باعث از بین رفتن اطلاعات نمی‌شود (مشکلی که در AINE وجود دارد).

در مقاله‌ی دیگری [۱۳۳] متدی متفاوت نسبت به الگوریتم پیشین برای تولید قوانین فازی برای کلاس‌بندی، ارائه شده است. نام این الگوریتم ایف‌رِیز[۱] است. این الگوریتم از انتخاب کلونی برای بهینه‌سازی جمعیتی از قوانین فازی (برای کلاس‌بندی) استفاده می‌کند. هر قانون در این الگوریتم یک آنتی‌بادی است و نمونه‌های آموزش که باید کلاس‌بندی شوند، آنتی‌ژن‌ها هستند. پیوند کل جمعیت با داده‌های آموزش در نسل‌های مختلف با تکامل جمعیت قوانین فازی افزایش می‌یابد، تا جایی که به میزان پیوند بیشینه برسد. نتایج تست بر روی مجموعه داده‌های عمومی با نتایج حاصل از C4.5 که یک الگوریتم شناخته شده برای کشف قوانین کلاس‌بندی است، قابل مقایسه هستند.

۳-۶-۱-۳ راه‌حل‌های تئوری خطر

تئوری خطر به دلیل نوظهور بودن در مقالات معدودی استفاده شده و در سال‌های اخیر است که مورد توجه قرار گرفته است. از جمله‌ی کاربردهای تئوری خطر در داده‌کاوی می‌توان به دو کاربرد ارائه شده توسط سِکِر[۲] و همکاران در [۱۳۴] و [۲] اشاره کرد. سکر در این مقاله به ایجاد دو سیستم ایمنی مصنوعی می‌پردازد. سیستمی برای کلاس‌بندی ایمیل (AISEC) و سیستمی دیگر برای کاوش صفحات وبی که از نظر کاربر بدیع، غیرمنتظره و متعجب‌کننده هستند (AISIID). عقیده‌ی طراح بر اینست که استفاده از تئوری خطر برای کاربردهای یادگیری پیوسته برای داده‌های متغیر و حجیم مناسب است. بنابراین در هر دو کاربرد از تئوری خطر استفاده شده است. در AISEC سیگنال خطر از بازخورد واکنش کاربر نسبت به ایمیل کلاس‌بندی شده تولید می‌شود. شبه‌کد این الگوریتم در شکل ۳-۱۳ آمده است.

---

[۱] IFRAIS

[۲] Secker

```
PROGRAM AISEC
    train(training set)
    WAIT until (an e-mail arrives or a user action is intercepted)
        ag ← convert e-mail into antigen
        IF(ag requires classification)
            classify(ag)
            IF(ag is classified as uninteresting)
                move ag into user accessible storage
            ELSE
                allow e-mail to pass through
        IF(user is giving feedback on ag)
            update_population(ag)
```

شکل ۳-۱۴ شبه‌کد الگوریتم AISEC

کاربرد دیگر تئوری خطر، توسعه‌ی SOSDM در [۱۳۵] است. در این مقاله، کمیتی به نام رضایت برای هر آنتی‌بادی محاسبه می‌شود. اگر سطح کلی رضایت سیستم از حد آستانه‌ای کمتر باشد، سیستم با تولید یک آنتی‌بادی جدید به این حالت پاسخ می‌دهد. با استفاده از این ایده‌ی محاسبه‌ی رضایت، در این سیستم مرگ طبیعی و غیرطبیعی نیز مدل می‌شود.

در [۱۳۶] نیز سیستمی برای پیشنهاد مقالات خبری بر اساس تئوری خطر و الگوریتم انتخاب کلونی طراحی شده است. در این سیستم سیگنال خطر، واکنش‌های کاربر از قبیل خواندن یا چاپ کردن مقاله است.

۳-۶-۲ سیستم‌های یادگیری پیوسته مبتنی بر سیستم‌های ایمنی مصنوعی دیگر

۳-۶-۳ وب و داده‌کاوی با سیستم ایمنی مصنوعی

مروری بر کارهای انجام شده، نشان می‌دهد که پروژه‌های کمی در زمینه‌ی کاربردهای کاوش متن و کاوش وب با استفاده از سیستم‌های ایمنی مصنوعی تعریف و انجام شده است. اولین مقاله‌ای که به کاربرد کاوش متن به‌وسیله‌ی AIS اشاره کرد [۸۸] است. در این مقاله‌ی نظری، بطور مختصر به ارائه‌ی کاربرد AIS در کاوش متن پرداخته شده است. در این سیستم هر سند دارای مجموعه‌ای از

ویژگی‌هاست و آنتی‌بادی‌های AIS با ویژگی‌های اسناد جور می‌شوند. هنگامی‌که کاربر صریحاً یا ضمنی، تمایل خود را به سندی نشان دهد، سیگنال خطر تولید می‌شود. این مقاله، اولین مقاله‌ای بود که در آن از عبارت "جذاب" برای کاوش اسناد با AIS استفاده شد.

دومین مقاله که آن هم به همکاری کایزر[۱] نوشته شده [۱۳۷] اولین کار در زمینه کلاس‌بندی اسناد با AIS است و بعد از آن در [۱۳۸] [۱۳۹] [۱۴۰] از الگوریتم انتخاب کلونی برای کلاس‌بندی اسناد استفاده شده است. وظیفه‌ی سیستم در این مقالات قرار دادن اسناد در دو دسته‌ی اسناد مرتبط با موضوع و اسناد نامرتبط با موضوع است. هر سلول، نشان‌دهنده‌ی حضور یا عدم حضور ویژگی‌های خاص (کلمات) با استفاده از نمایش باینری است. از یک الگوریتم استخراج ویژگی، برای تبدیل یک سند HTML به بردار ویژگی بولین، استفاده شده است. هر سلول دارای یک آستانه‌ی هماهنگی نیز هست که این عدد بین ۰ و ۱ است و وزن سلول در جمعیت را مشخص می‌کند. به منظور پاداش برای سلول‌هایی که کلاس یک سند را با اطمینان زیادی تعیین می‌کنند، آستانه‌ی پایین‌تر در نظر گرفته می‌شود. الگوریتم با کلاس‌بندی صفحات مجموعه داده‌ی صفحات وب سیسکیل[۲] و وبرت[۳] از انبار داده UCI که مجموعه‌ای از داده‌های استانداردی است که کاوشگران داده برای تست الگوریتم‌هایشان استفاده می‌کنند، تست شده است [۱۱۶]. این مجموعه داده شامل صفحات وب است که در چهار مقوله کلاس‌بندی می‌شوند. وظیفه‌ی سیستم‌های طراحی شده توسط AIS تعیین موضوع یک صفحه‌ی دیده نشده است.

در [۱۴۱]، نتایج اولیه‌ای برای کلاس‌بندی دو کلاسه و چندکلاسه‌ی اسناد ارائه شده است. نتایج نشان می‌دهد که تفاوت زیادی در دقت مسئله‌ی چهارکلاسه با مسئله‌ی دوکلاسه وجود ندارد. معیاری برای

---

[۱] Cayzer

[۲] Syskill

[۳] Webert

سود اطلاعاتی به منظور استخراج n ویژگی بهتر (کلمات دارای بار اطلاعاتی بیشتر) استفاده شد و رشته‌ی ویژگی هر سلول شامل یک نمایش باینری از وجود یا عدم وجود این کلمات است.

مقاله‌ی [۱۴۲] نیز با فیلتر کردن اسناد به‌وسیله‌ی AIS سروکار دارد. در این مقاله از تئوری شبکه‌ی ایمنی جرن[۱] برای ارائه‌ی علایق کاربران استفاده شده است. علاوه بر کلاس‌بندی، مقالاتی نیز در زمینه خوشه‌بندی اسناد وجود دارد. [۱۴۳] یک مثال از این نوع است. [۱۴۴] نیز از aiNet برای خوشه‌بندی اسناد استفاده کرده‌اند. در این مقاله نیز از نمایش باینری استفاده شده است و تقریباً ۱۵٪ ویژگی‌ها بر اساس تغییرات فرکانس کلمات انتخاب شده‌اند. تحلیل نتایج روی ۲۰ داده‌ی گروه‌های خبری یوزنت[۲] نشان می‌دهد که aiNet خوشه‌های فشرده‌ای مخصوصاً برای داده‌های حجیم و نویزی ایجاد می‌کند.

کارهای انجام شده توسط تیکراس[۳] و گرین‌اسمیت[۴] هر دو کاوش محتویات وب است. هر دو داده‌های وب را کلاس‌بندی می‌کنند ولی تاکید آن‌ها بیشتر بر کلاس‌بندی اسناد است که این اسناد از وب استخراج شده‌اند. در مورد کار [۱۴۴] نیز همین مسئله صادق است. ولی همه‌ی این کارها مشترکاتی با سیستم پیشنهاد وب‌سایت که توسط موریسون[۵] و آیکلین[۶] در [۱۴۵] ارائه شده است، دارند. در این مقاله اگرچه از شبکه‌ی ایمنی استفاده شده است ولی یک کلاس‌بندی باینری انجام می‌شود، به این صورت که به جای نسبت دادن کلاس به یک نمونه‌ی ناشناخته، سیستم پیشنهادی مشابه به "سیستم ایمنی اطلاعاتی" ایجاد می‌کند.

---

[۱] Jern

[۲] Usenet

[۳] Twycross

[۴] Greensmith

[۵] Morrison

[۶] Aicklin

به جزء کاوش محتویات وب و کاوش اسناد، یک مقاله نیز در مورد استخراج اطلاعات از داده‌های دسترسی به وب توسط کاربران یا WUM وجود دارد [۱۴۶]. در این مقاله بر اساس شبکه‌ی ایمنی مصنوعی AINE [۱۴۷] سیستمی برای کاوش اطلاعات دسترسی به وب و استخراج مجموعه اقلام مکرر طراحی شده است. الگوریتم استفاده شده در این مقاله در شکل ۳-۱۴ آورده شده است.

```
Fix the maximum number of resources (N_max) to be
allocated (the total number of B cells allocated to all
ARBs in the network);
Initialize AIN (Artificial Immune Network): select
ARB population using a cross section of the input
data;
Load antigen population = remaining training data;
Repeat
    Present antigen set to each ARB in network;
    Compute ARB stimulation level;
    Allocate B cells to ARBs based on stimulation
    level;
    Remove weakest ARBs (the ones that get
        allocated zero B cells) from population;
    WHILE total # cells is still > N_max {
        Remove weakest ARBs from network
    }
    If not Termination condition {
        Clone and mutate remaining ARBs;
        Integrate new ARBs into AIN;
    }
} Until Termination condition.
```

شکل ۳-۱۵ الگوریتم WUM ارائه شده در [۱۴۷]

۳-۷ جمع بندی

در این بخش به مروری بر سیستم ایمنی طبیعی و مصنوعی پرداختیم. بعد از ارائه‌ی برخی از اصول ایمنی‌شناسی، از چهارچوب کلی برای مهندسی AIS برای توضیح لایه‌های مختلف نمایش، میل پیوندی و فرآیندهای مورد نیاز، استفاده کردیم. سپس کارهای مرتبط با موضوع را با تمرکز بر کاربردهای داده‌کاوی مرور کردیم.

در این بخش و بخش ۲، شالوده‌ی کار برای ادامه‌ی پروژه بنا نهاده شد. در ادامه‌ی پروژه، از اصول ایمنی ارائه شده در این فصل برای ارائه‌ی کار پیشنهادی برای استخراج مجموعه اقلام مکرر به منظور استخراج اطلاعات از داده‌های دسترسی به وب استفاده خواهیم کرد.

۴

الگوریتم پیشنهادی برای استخراج مجموعه آیتم‌های مکرر از داده‌های دسترسی به وب

مقدمه

چرا سیستم ایمنی؟

سیستم ایمنی مصنوعی برای استخراج اطلاعات از داده‌های استفاده از وب

الگوریتم پیشنهادی

جمع بندی

"The secret to creativity is, knowing how to hide your sources!"
--Albert Einstein

فصل ۴: الگوریتم پیشنهادی برای استخراج مجموعه آیتم‌های مکرر از داده‌های دسترسی به وب

۴-۱ مقدمه

در بخش‌های پیشین انگیزه‌ی استفاده از سیستم ایمنی مصنوعی برای کاوش در داده‌های استفاده از وب تشریح شد. همچنین با الگوریتم‌های موجود در سیستم ایمنی مصنوعی و با نوع عملکرد کلی و اجزاء در این نوع سیستم‌ها آشنا شدیم. در این بخش به اعمال سیستم ایمنی مصنوعی در کاربرد استخراج اطلاعات از داده‌های دسترسی به وب خواهیم پرداخت و جزئیات سیستمی که برای این منظور پیشنهاد می‌کنیم، ارائه خواهد شد. وظیفه‌ی این سیستم پیشنهادی که به آن سیستم ایمنی مصنوعی برای استخراج اطلاعات از داده‌های دسترسی به وب (یا بطور اختصار AISWUM) می‌گوییم، پیدا کردن مسیرهای پر رفت و آمدی است که کاربران درون یک وب‌سایت بر اساس علایق یا نوع استفاده‌ای که دارند، طی می‌کنند. برای مثال در مورد وب‌سایت یک دانشگاه مسیرهایی خاص در زمان‌های مختلف قابل شناسایی است، مانند مسیری که کاربرها برای انتخاب واحد (که در زمان‌های خاصی از سال تحصیلی انجام می‌شود) می‌پیمایند؛ یا مسیر ثبت‌نام برای آزمون‌های خاص که از طریق سایت انجام می‌شود و یا صفحات مرتبط با یک دانشکده، یک درس خاص و یا یک استاد خاص. بنابراین هدف، پیدا کردن مسیرهای پر رفت و آمد در یک وب‌سایت است. این مسیرها از چندین URL که هر URL خود مرتبط با یک صفحه‌ی وب است، تشکیل شده‌اند. به مجموعه URLهایی که کاربر در مسیر خود در یک نوبت مراجعه به سایت می‌پیماید، یک نشست[۱] گفته می‌شود. در سیستم پیشنهادی از ترکیب مفهوم جدید تئوری خطر و شبکه‌ی ایمنی برای استخراج مجموعه اقلام مکرر به منظور استخراج اطلاعات از داده‌های دسترسی به وب استفاده خواهیم کرد. بنابراین سیستم ایمنی

[۱] Session

مصنوعی مورد استفاده از دو بخش اصلی تشکیل شده است. بخش اول تعیین معتبر بودن نشست ایجاد شده توسط کاربر (خطرناک بودن آنتی‌ژن ورودی) و بخش دوم یادگیری نشست‌های کاربران و تعیین مجموعه‌ی صفحات وبی است که مکرراً با یکدیگر در یک نشست ظاهر شده‌اند (ارائه‌ی آنتی‌ژن به شبکه‌ی ایمنی و فرآیند یادگیری آنتی‌ژن‌های ورودی و ایجاد آنتی‌بادی‌هایی با بیشترین قدرت مقابله با آنتی‌ژن‌ها).

در سیستم‌هایی که پیش از این برای استخراج اطلاعات از داده‌های دسترسی به وب ایجاد شده است، بخش پیش‌پردازش جدا از بخش کشف دانش است و البته با وجود بخش پیش‌پردازش، باز هم داده‌ها بسیار نویزی هستند و این نویز بر دانش استخراج شده تاثیر می‌گذارد؛ چراکه در اکثر این سیستم‌ها بخش پیش‌پردازش فقط برای ایجاد نشست‌ها استفاده می‌شود و در این بخش دانش نهفته در نشست‌ها، در نظر گرفته نمی‌شود. در حالی‌که دانش موجود در هر یک از نشست‌ها می‌تواند به تشخیص معتبر بودن و حتی میزان معتبر بودن آن‌ها کمک کند. در سیستم پیشنهادی علی‌رغم تحمل نویز بالا، فازی برای حذف داده‌های نویزی (حذف آنتی‌ژن‌هایی که باعث ایجاد تغییر مثبت در سیستم نمی‌شوند) ایجاد شده است. علاوه بر این، سیستمی که ارائه خواهد شد برای داده‌های پویا و کاربردهایی که احتیاج به یادگیری پیوسته دارند و کاربردهای بلادرنگ، بسیار مناسب است؛ زیرا این الگوریتم به‌صورت افزایشی طراحی شده است. بنابراین برای استخراج اطلاعات از داده‌های دسترسی به وب که ذاتاً پویاست، این الگوریتم، بسیار مناسب عمل می‌کند.

۴-۲ چرا سیستم ایمنی؟

به علت برخی از خصوصیاتی که در ذات سیستم ایمنی وجود دارد، استفاده از سیستم ایمنی مصنوعی برای استخراج اطلاعات از داده‌های دسترسی به وب، بسیار مناسب به نظر می‌رسد. در کار انجام شده توسط [۳] این خصوصیات مفصلاً توصیف شده‌اند. از جمله‌ی این خصوصیات بارز سیستم ایمنی مصنوعی که در کاربرد استخراج اطلاعات داده‌های دسترسی به وب با اهمیت است:

تشخیص الگو[۱]: توانایی تشخیص الگوهای شبیه به الگوهای نمونه‌های آموزش. این خصوصیت از خصوصیات مهم در هر الگوریتم تشخیص الگویی است که در سیستم ایمنی طبیعی با دقت بالایی انجام می‌شود.

تنوع[۲]: مانند سیستم ایمنی، وب نیز دارای تنوع زیادی است. وب دارای فرمت‌های اطلاعاتی بسیار متفاوت و متنوع، از متن معمولی گرفته تا صفحات گرافیکی است. سیستم ایمنی نیز شامل انواع متفاوتی از سلول‌هاست که هر کدام مختص انجام کار خاصی هستند.

خاصیت توزیعی[۳]: برتری سیستم‌های توزیعی نسبت به سیستم‌های دیگر نه تنها تحمل خطا، بلکه امکان پردازش موازی و بنابراین رسیدن به زمان اجرای کمتر می‌باشد. در یک سیستم وب کاوی که قدرت پردازش و ذخیره‌سازی اهمیت زیادی دارد، این ویژگی، ویژگی ارزشمندی است.

خودسازماندهی[۴]: یک سیستم ایمنی مصنوعی با طراحی خوب، بطور اتوماتیک برای تطابق با داده‌های نهفته تغییر می‌کند. در سیستم‌های وب کاوی، به علت پویایی بسیار زیاد وب، این نکته اهمیت بسیار زیادی دارد.

تحمل نویز[۵]: به علت سهولت ایجاد صفحات وب و دسترسی به وب، سوالاتی مبتنی بر کیفیت اطلاعات مربوط به وب به‌وجود می‌آید. بنابر همین سهولت استفاده، خطا در داده‌های وب بسیار زیاد است. سیستم ایمنی نیز سیستمی با تحمل نویز بالا است، به این صورت که در سیستم ایمنی تطابق کامل بین آنتی‌ژن و آنتی‌بادی برای ایجاد پاسخ در برابر آنتی‌ژن لازم نیست. به علت وجود تابع پیوند (affinity function) در الگوریتم‌های سیستم ایمنی مصنوعی و جمعیت متنوع آنتی‌بادی‌ها که هر

---

[۱] Pattern recognition

[۲] Diversity

[۳] Distributivity

[۴] Self-organization

[۵] Noise tolerance

کدام از آن‌ها با وجه متفاوتی از نمونه‌ی ورودی (آنتی‌ژن) مطابق می‌شوند، AIS پتانسیل بالایی برای فیلتر کردن داده‌های نویزی و آشکار کردن مفاهیم نهفته‌ی زیرین دارد. چنین تحمل نویزی برای الگوریتم‌های کاوش داده‌های با کیفیت کم ضروری است. در سطوح بالاتر نیز، توپولوژی، محتوا و علایق کاربران همواره در حال تغییر است. توانایی تطابق با این تغییرات یک ویژگی مهم برای سیستم‌های وب کاوی است. همه روزه دستگاه‌های کامپیوتر متعدد و داده‌های جدید به اینترنت اضافه و از آن کم می‌شوند. سلول‌های ایمنی هم رفتار مشابهی دارند، با نرخ ثابتی می‌میرند و با نرخ ثابتی تکثیر می‌شوند. نشان داده شده است که AIS تطابق‌پذیر، مقاوم و منعطف است.

از بین تمام خصوصیاتی که ذکر شد، خاصیت تطابق[۱]، تحمل خطا[۲]، مقاومت[۳] و توانایی آن در مقیاس‌های بزرگ[۴]، خصوصیات اصلی در این پروژه هستند. همانطور که قبلاً هم بیان شد، الگوریتم‌های AIS تحمل بالایی در برابر خطا دارند و تحمل خطا مفهومی مهم در کاربرد استخراج اطلاعات از داده‌های دسترسی به وب است، با این وجود از تئوری خطر نیز برای حذف جلساتی (آنتی‌ژن‌هایی) که معتبر نیستند (مضر هستند) برای کارایی بالاتر محاسبات، در الگوریتم ارائه شده استفاده خواهیم کرد.

۴-۳ سیستم ایمنی مصنوعی برای استخراج اطلاعات از داده‌های استفاده از وب

AISWUM الگوریتمی است که بر اساس سیستم ایمنی مصنوعی برای تعیین مسیرهای پر رفت و آمدی که کاربران درون یک وب‌سایت (بر اساس علایق یا نوع استفاده‌ای که دارند) می‌پیمایند،

---

[۱] Adaptability

[۲] Noise tolerance

[۳] Roboustness

[۴] Scalability

طراحی شده است. داده‌های خامی که این اطلاعات از آن‌ها استخراج می‌شود داده‌های موجود در لاگ‌های وب است که در سرور وب مربوطه ذخیره می‌شوند.

به هر URL در یک نشست، آیتم و به مجموعه‌ای از URLها، مجموعه آیتم گفته می‌شود.

در سیستم پیشنهادی، از دو مکانیزم موجود در سیستم ایمنی، یعنی تئوری خطر و شبکه‌ی ایمنی استفاده می‌شود. از تئوری خطر برای تعیین نشست‌های معتبر و مطلوب کاربران و از شبکه‌ی ایمنی برای یادگیری پیوسته‌ی مجموعه آیتم‌های مکرر (مسیرهای پر رفت‌وآمد و مهم پیموده شده توسط کاربران وب‌سایت) استفاده می‌شود. پروسه‌ی AISWUM به‌صورت زیر خلاصه می‌شود:

همان‌طور که قبلاً هم شرح داده شد در این سیستم ایمنی مصنوعی، آنتی‌ژن‌ها نشست‌هایی هستند که بر اثر بازدید کاربران از وب‌سایت به‌وجود می‌آید و هر نشست مجموعه‌ای از URL صفحات وب ملاقات شده توسط کاربران است. در ابتدای الگوریتم به تعداد ماکزیمم آنتی‌بادی‌هایی که در الگوریتم تعیین می‌شود از نشست‌های معتبر انتخاب و به عنوان آنتی‌بادی وارد سیستم می‌شوند و بقیه‌ی نشست‌ها به عنوان آنتی‌ژن یکی یکی وارد سیستم می‌شوند.

AISWUM از مفاهیمی چون فرکانس بازدید از صفحه، مدت درنگ کاربر در یک صفحه و میزان شباهت مفهومی بین صفحات یک نشست استفاده کرده و تعیین می‌کند که آیا این نشست، نشست معتبری هست یا خیر، بنابراین همان‌طور که در [۱۰۹] آمده است در کاربردهای مختلف، عبارت خطر تعبیرهای متفاوتی می‌تواند داشته باشد، در الگوریتم AISWUM عبارت خطر به عبارت معتبر، تغییر پیدا کرده است. به اینصورت برای شناسایی آنتی‌ژن احتیاج به دو سیگنال است، که سیگنال اول، سیگنال اعتبار نشست (سیگنال خطر) و دومین سیگنال، پیوند آنتی‌ژن با آنتی‌بادی است. از این طریق نشست‌هایی که در فاز پیش‌پردازش به‌صورت دقیقی ایجاد نشده‌اند (جلسات از طریق تابع مکاشفه‌ای[۱] زمانی با زمان ماکزیمم ۳۰ دقیقه در هر نشست، تشکیل شده‌اند) و یا نشست‌هایی که به‌وسیله‌ی کاربرانی که با قصد خاصی وارد وب سایت نشده‌اند و مسیری که می‌پیمایند معنی و هدف

[۱] Heuristic function

خاصی ندارد (browsing)، ایجاد شده است، حذف می‌شوند و از صرف زمان و انرژی که وارد شدن این نشست‌ها به شبکه ایمنی تحمیل می‌کند، جلوگیری می‌شود.

در فاز دوم آنتی‌ژن‌ها یک به یک به آنتی‌بادی‌ها عرضه می‌شوند. تحریک آنتی‌بادی‌ها بر اساس فاصله‌ی آنتی‌بادی و آنتی‌ژن و اندازه‌ی محدوده‌ی تاثیر آنتی‌بادی و همچنین وزن آنتی‌ژن (وزن نشست) عرضه شده، محاسبه می‌شود، آنتی‌بادی‌هایی که سطح تحریک بالاتری دارند، تحت تکثیر و جهش هدایت‌شده قرار می‌گیرند. و در نهایت بعد از ارائه‌ی همه‌ی آنتی‌ژن‌ها به سیستم، آنتی‌بادی‌های موجود هر کدام نماینده‌ی یک مسیر مهم مکرر پیموده شده در وب سایت است. ضمناً با توجه به میزان تحریک آنتی‌بادی، میزان تکرر یا قدرت یا اهمیت مسیرهای استخراج شده، مشخص می‌شود.

آنچه که در پاراگراف بالا به عنوان عملکرد AISWUM گفته شد، اساس AISWUM است و جزئیات به بخش‌های بعدی موکول شده است.

در AISWUM، نوآوری‌های متعددی مشاهده می‌شود که از آن میان می‌توان به موارد زیر اشاره کرد:

- استفاده از تئوری خطر برای فیلتر کردن نشست‌های نامعتبر (نویز)، گرچه سیستم به علت تحمل نویز بالا قادر به دفع تاثیر آن‌ها در الگوهای استخراجی است، ولی عرضه‌ی این نویزها به سیستم باعث بالا رفتن محاسبات و به تبع آن افزایش زمان اجرا می‌شود.
- استفاده از ارتباطات مفهومی بین صفحاب وب، برای تعیین هم‌نواحتی بین URLهای موجود در یک نشست.
- استفاده از فرکانس URLهای درون یک نشست و مدت درنگ کاربر بر روی صفحات مربوط به URLهای درون صفحه، هم برای حذف جلسات نامعتبر و بدون مفهوم و هم برای وزن دادن به مجموعه‌های آیتم و ایجاد تفاوت بین آیتم‌های مختلف بر اساس اهمیت آن‌ها.
- ارائه و استفاده از معادلات افزایشی، برای ایجاد یک الگوریتم افزایشی مناسب برای کاربرد وب کاوی... .
- داخل کردن وزن صفحات و وزن نشست‌ها در معادلات شبکه.

- طراحی جهش هدایت شده برای بهره‌گیری از نتایج جهش در عین حال جلوگیری از تصادفی بودن آن.

سیستمی که برای کاوش داده‌های استفاده از وب توسط سیستم ایمنی مصنوعی در این رساله پیشنهاد شده است مانند فیلتری فعال برای کشف اطلاعات عمل می‌کند. همچنین همان‌طور که توضیح داده شد، دو نوع تصمیم‌گیری در AISWUM انجام می‌شود، یک تصمیم باینری (تئوری خطر) برای تعیین اعتبار آنتی‌ژن‌ها و دیگری تعیین آنتی‌بادی که باید به آنتی‌ژن واکنش نشان دهد یا تعیین تعلق نشست به هر کدام از آنتی‌بادی‌های موجود در شبکه.

۴-۴ الگوریتم پیشنهادی

در این بخش به توصیف الگوریتم با تمام جزئیات آن خواهیم پرداخت. این بخش طوری طراحی شده تا توجه خواننده به نکات مهم AISWUM جلب شود و در عین حال دیدی کلی از نوع عملکرد آن به خواننده می‌دهد. بعد از توضیح الگوریتم، الگوریتم را برای کمک به پیاده‌سازی به صورت شبه کد، ارائه خواهیم کرد.

در این بخش از چهارچوب لایه لایه‌ی AIS که توسط دی‌کاستر و تیمیس [۳] ارائه شده است و در فصل دوم ارائه شد، استفاده خواهیم کرد: تعریف اجزاء سیستم و طرز نمایش آن‌ها، معیار شباهت بین آنتی‌بادی و آنتی‌ژن و الگوریتم‌ها و فرآیندهای بکار رفته در الگوریتم.

۴-۴-۱ نکات کلی

۴-۴-۱-۱ پیش‌پردازش

برای ایجاد کردن نشست‌ها از روی داده‌های دسترسی کاربران به وب (Web log) همان‌طور که در فصل دو توضیح داده شد، بعد از پاک‌سازی داده و حذف ورودی‌هایی که اطلاعات نامربوط دارند، می‌توان از الگوریتم‌های مکاشفه‌ای مختلفی استفاده کرد.

در پاک‌سازی داده، ورودی‌هایی را که مربوط به تصاویر و فایل‌های گرافیکی هستند (که هنگام دسترسی به صفحه‌ی وبی که حاوی این تصاویر و فایل‌های گرافیکی است، در لاگ وب ذخیره می‌شوند و دسترسی صریح به آن‌ها انجام نشده) را حذف می‌کنیم. همچنین ورودی‌هایی را که کد حالت HTTP آن‌ها نشان‌دهنده بروز خطاست یا متد درخواست آن‌ها چیزی به جزء Get و Post است به علاوه‌ی جلساتی که طول آن‌ها کمتر از دو URL است، در فاز پاک‌سازی از لاگ وب حذف می‌کنیم.

در این پروژه برای تشکیل نشست‌ها از داده‌های ذخیره شده در لاگ وب از الگوریتم مکاشفه‌ای استفاده شده است که بر اساس زمان، یعنی زمان کل نشست و با تعیین یک آستانه به عنوان حداکثر زمان مجاز برای یک نشست، طراحی شده است. با توجه به کارهای پیشین [۱۸۴] [۱۴۹] و بر اساس نتایج تجربی به‌دست آمده، در سیستم حاضر، آستانه‌ی ۳۰ دقیقه‌ای برای زمان کل نشست در نظر گرفته شده است.

۴-۴-۱-۲ رویکرد دو سیگناله (تئوری خطر)

در [۲] نشان داده شده است که استفاده از رویکرد دو سیگناله برای الگوریتم‌های یادگیری پیوسته سودمند است. همان‌طوری که قبلاً نیز در فصل سه توضیح داده شده است، این رویکرد می‌تواند از ارائه‌ی نمونه‌های نویزی و اضافی به سیستم جلوگیری کند. در الگوریتم حاضر نیز از رویکرد دو سیگناله جهت نیل به این اهداف استفاده خواهد شد.

زمانی‌که نشستی (آنتی‌ژنی) وارد سیستم می‌شود، باید تصمیمی اتخاذ شود مبنی بر اعتبار یا جذابیت نشست (مضر بودن آنتی‌ژن). بنابراین باید معیار یا خصوصیتی در داده بیابیم که از طریق آن سطح اعتبار یا جذابیت نشست تعیین شود. در مورد این موضوع در بخش‌های بعدی همین فصل به تفضیل بحث خواهیم کرد.

۴-۴-۲ فلوچارت الگوریتم پیشنهادی

بیان دیاگرامی الگوریتم را در فلوچارت شکل ۴-۱ مشاهده می‌کنید. فلوچارت به فهم آسان‌تر مراحل الگوریتم کمک خواهد کرد.

۴-۴-۳ اجزاء سیستم و طرز نمایش آن‌ها

هر سلول ایمنی شامل سه بخش زیر است:

۱. یک مجموعه آیتم مکرر (مجموعه‌ای از صفحات وب که با هم در تعداد زیادی از ورودی‌ها مشاهده شده‌اند).

۲. معیاری که نشان دهنده‌ی میزان قدرت مسیر مربوطه است (سطح تحریک).

۳. معیاری به نام محدوده‌ی تاثیر که نشان‌دهنده‌ی محدوده‌ی عملکرد آنتی‌بادی می‌باشد.

هر آنتی‌ژن یک نشست ایجاد شده از داده‌های ذخیره شده در لاگ وب در وب‌سرور است و دارای وزنی است که از میانگین وزن کل صفحات موجود در نشست به‌دست می‌آید. وزن صفحات نشان دهنده‌ی میزان جذابیت آن صفحه‌ی خاص در نشست مربوطه برای کاربر است، که طرز تعیین این وزن‌ها در بخش‌های بعد توضیح داده خواهد شد.

برای نمایش آنتی‌بادی و آنتی‌ژن از آرایه‌ای به طول تعداد URLهای موجود در وب‌سایت مورد نظر استفاده می‌شود. هر خانه از این آرایه را متناظر با یک URL در نظر گرفته و در صورتی‌که URLی در نشست وجود داشته باشد، خانه‌ی مربوط به آن URL را در آرایه‌ی آن نشست، ۱ و در غیراینصورت ۰ قرار می‌دهیم. هر آنتی‌بادی نیز، آرایه‌ای به طول مشابه (یعنی به طول کل URLهای موجود در وب‌سایت) است. این آرایه، آرایه‌ای باینری است و در صورت وجود یک URL در آنتی‌بادی، خانه‌ی مربوط به آن در آرایه، ۱ و در غیراینصورت ۰ است.

برای هر آنتی‌ژن معیاری به نام اعتبار آن محاسبه می‌شود که یک عدد حقیقی بین ۰ و۱ است و از ترکیب دو معیار دیگر به نام‌های هم‌نواختی[۱] نشست و میانگین جذابیت[۲] صفحات درون نشست ایجاد می‌شود. برای هر آنتی‌بادی نیز هنگام عرضه‌ی آنتی‌ژن، وزنی محاسبه می‌شود که نسبت به شباهت بین آنتی‌بادی و آنتی‌ژن و محدوده‌ی تاثیر آنتی‌بادی متغیر است و بر اساس این وزن میزان تحریک آنتی‌بادی محاسبه می‌شود. آنتی‌بادی‌هایی با سطح تحریک پایین از شبکه حذف می‌شوند.

۴-۴-۴ معیارهای هم‌نواختی و جذابیت نشست

اگر فرض کنیم، P مجموعه صفحات وبی باشند که توسط کاربرها مورد دسترسی قرار گرفته‌اند $P=\{p_1,p_2,...,p_m\}$ که هر کدام از این صفحات با URL یکتای خود مشخص می‌شوند و S مجموعه‌ی جلساتی باشد که از لاگ‌های وب به‌دست آمده است $S=\{s_1,s_2,...,s_n\}$. می‌توان هر نشست را به‌صورت $s_i=\{(p_1,w_{p_1}),(p_2,w_{p_2}),...,(p_m,w_{p_m})\}$ نمایش داد. w_{p_i} وزنی است که به صفحه‌ی وب p_i در نشست s_i داده شده است.

وزن w_{p_i} هر صفحه باید طوری تعیین شود که نشان‌دهنده‌ی علاقه‌ی کاربر به آن صفحه‌ی خاص باشد. به عبارت دیگر، همه‌ی صفحاتی که توسط کاربر ملاقات شده‌اند برای کاربر دارای ارزش یکسانی نبوده‌اند و علاقه یا نیاز کاربر به صفحات درون نشست دارای سطوح متفاوتی است. بنابراین به‌صورت زیر معیار وزنی برای تخمین درجه‌ی علاقه‌ی کاربر به یک صفحه پیشنهاد می‌شود. قبل از ارائه‌ی این معیار، لازم است با سه مفهوم فرکانس[۳] و مدت درنگ[۴] و شباهت[۵] آشنا شویم.

---

[۱] Consistency

[۲] Interest

[۳] Frequency

[۴] Duration

[۵] Similarity

فرکانس، تعداد مراجعات کاربر به یک صفحه‌ی وب در یک نشست است. طبیعی است که فرض کنیم صفحات وبی که بطور مکرر توسط کاربر به آن‌ها مراجعه می‌شود، برای کاربر جذابیت بیشتری دارند. فرمول فرکانس در رابطه‌ی ۴-۱ آمده است. این مقدار با تعداد کل صفحات ملاقات شده، نرمال شده است.

$$Frequency(CurrentPage) = \frac{NumberOfVisits(CurrentPage)}{\sum_{Page \in VisitedPage}(NumberOfVisits(Page))} \qquad (۱-۴)$$

مدت درنگ به زمانی گفته می‌شود که توسط کاربر روی یک صفحه‌ی وب سپری می‌شود. به عبارتی مدت درنگ به زمانی گفته می‌شود که توسط کاربر روی یک صفحه‌ی وب سپری می‌شود. به عبارت دیگر هر چه مدت زمان درنگ کاربر روی یک صفحه‌ی وب بیشتر باشد، آن صفحه برای کاربر جذابیت بیشتری دارد. اگر صفحه‌ای مورد علاقه‌ی کاربر نباشد، کاربر سریعاً به یک صفحه‌ی دیگر می‌رود [۱۵۰]. ولی حرکت سریع کاربر از یک صفحه به صفحه‌ی دیگر ممکن است به علت کوتاهی صفحه نیز باشد، بنابراین بهتر است مدت درنگ کاربر را با طول صفحه‌ی مورد نظر که کل بایت‌های صفحه است، نرمال کنیم. از رابطه‌ی ۴-۲ برای مدت درنگ کاربر استفاده می‌کنیم.

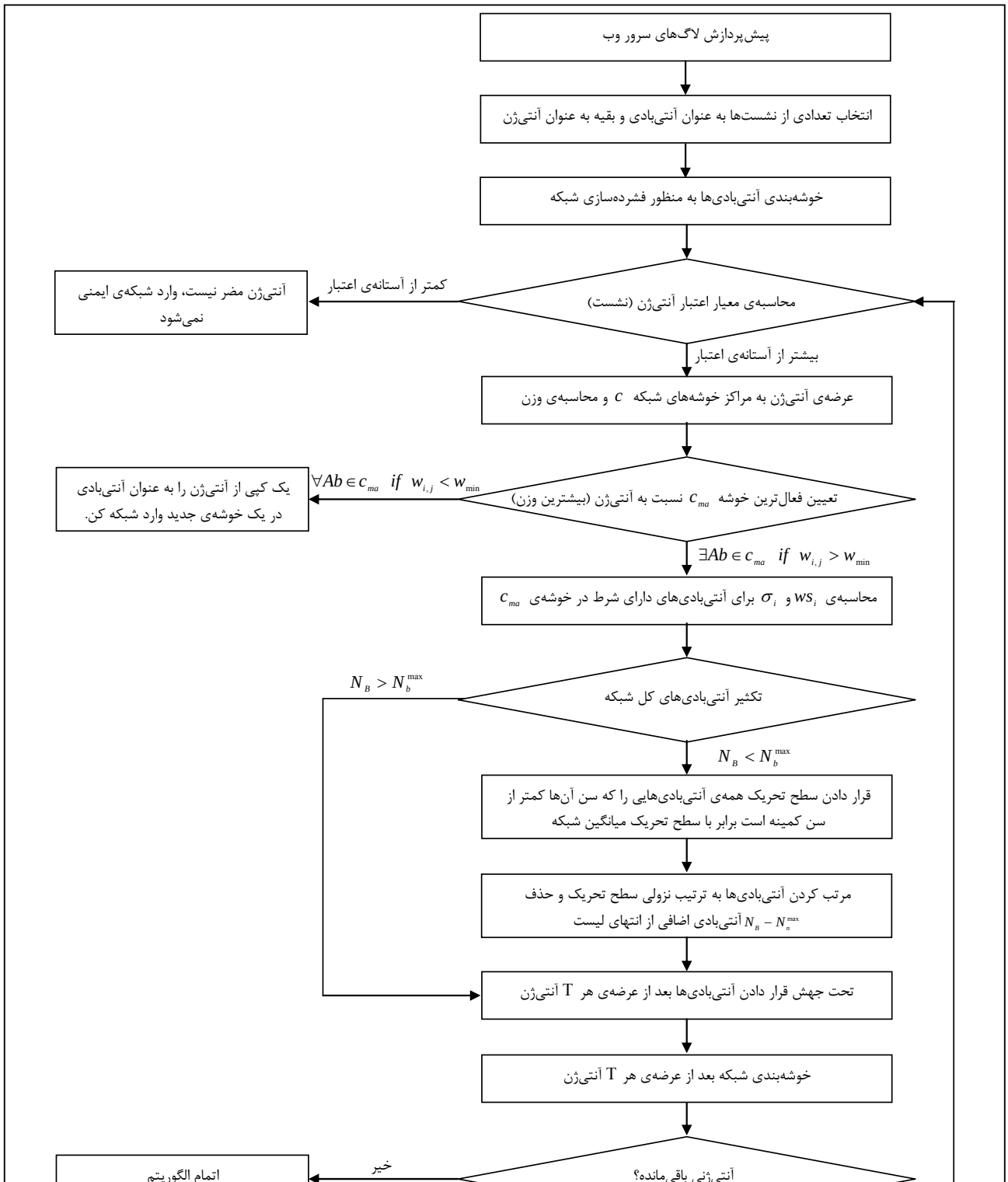
$$Duration(CurrentPage) = \frac{TotalDuration(CurrentPage) / Length(CurrentPage)}{\max_{page \in VisitedPage}(TotalDuration(Page) / Length(Page))} \qquad (۲-۴)$$

درنگ یک صفحه با بیشینه‌ی درنگ صفحات در یک نشست نرمال می‌شود و به این‌صورت مقدار درنگ عددی بین ۰ و ۱ خواهد بود.

مسئله‌ای که به‌وجود می‌آید، اینست که مدت زمان درنگ بر روی آخرین صفحه‌ی نشست موجود نیست، زیرا نمی‌توان این مدت را از تفریق زمان درخواست بعدی (که مربوط به نشست بعدی است) از زمان درخواست صفحه‌ی آخر نشست مربوطه به‌دست آورد. بنابراین از میانگین مدت درنگ در نشست مربوطه به عنوان مدت درنگ روی آخرین صفحه‌ی نشست استفاده می‌کنیم.

معیار هم‌نواختی (Consistency) در یک نشست، به این صورت تعریف می‌شود که رابطه‌ی مفهومی بین هر دو صفحه‌ی وب در یک نشست محاسبه می‌شود. به کمک این معیار، صفحات وبی که هنگام

تعیین نشست‌های موجود در داده‌های ذخیره شده در لاگ وب، اشتباهاً در یک نشست قرار گرفتند، یا صفحاتی که کاربر بدون استفاده از آن‌ها، فقط اشتباهاً به آن‌ها مراجعه کرده است و یا نشست‌هایی که توسط کاربرانی که بدون قصد خاصی (فقط در حال گشت و گذار در وب سایت بوده‌اند) وارد وب‌سایت شده‌اند، به‌وجود آمده است، از جلسات دارای مفهوم که اطلاعات مفیدی از آن‌ها استخراج می‌شود، متمایز می‌شوند.



شکل ۴-۱ فلوچارت الگوریتم پیشنهادی

برای این منظور لازم است، میزان شباهت صفحات موجود در وب‌سایت مورد مطالعه را نسبت به هم به‌دست آوریم. این‌کار را می‌توان به‌صورت دستی در وب سایت‌هایی که تعداد صفحات زیادی ندارند و به‌صورت اتوماتیک توسط رویکردهای موجود از قبیل استفاده از آنتولوژی‌ها یا کلمات کلیدی درون صفحه‌ها و ... انجام داد.

شباهت بین صفحات به این صورت محاسبه می‌شوند که ابتدا صفحات در مقوله‌های[۱] مختلف دسته‌بندی می‌شوند، سپس دسته‌های مختلف، گرافی به نام گراف شباهت ایجاد می‌کنند که در این گراف، هر نود متناظر با یک مقوله است. نودهایی که مربوط به مقوله‌هایی هستند که با هم ارتباط مستقیم دارند با لینک مستقیم به هم متصل می‌شوند و نودهای مربوط به مقوله‌های کم ارتباط تر به طور غیر مستقیم از طریق چندین لینک به هم مرتبط می‌شوند. با توجه به ساختار داده‌ی ارائه شده، محاسبه‌ی شباهت بین دو صفحه‌ی i و j از طریق رابطه‌ی زیر به‌دست می‌آید:

$$similarity(i, j) = 1 - \frac{dist_{i,j}}{D} \qquad (۳-۴)$$

که $dist_{i,j}$ فاصله‌ی بین دو صفحه در گراف شباهت (تعداد لینک‌های موجود در گراف بین دو صفحه) و D ماکزیمم فاصله بین دو نود در گراف است که با داخل کردن آن در رابطه‌ی (۴-۳)، مقدار شباهت[۲] عددی بین ۰ و ۱ خواهد بود.

---

[۱] Categories

[۲] Similarity

با استفاده از معیار شباهت که در رابطه‌ی (۴-۳) تعریف شده است، معیار هم‌نواختی نشست را به‌صورت رابطه‌ی (۴-۴) تعریف می‌کنیم:

$$Consistency(Session) = \frac{\sum_{k=1}^{P-1}\sum_{j=k+1}^{P} similarity(k, j)}{(P-1)(\frac{1}{2}P)} \qquad (۴-۴)$$

که در این رابطه، P تعداد صفحات درون نشست است. مخرج کسر برای نرمال کردن معیار هم‌نواختی در معادله گنجانده شده است.

دو معیار درنگ کاربر و فرکانس، دو معیار قوی برای نشان دادن میزان علاقه‌ی کاربر نسبت به صفحه‌های داخل یک نشست و معیار هم‌نواختی نشان‌دهنده‌ی اعتبار یک نشست است. بنابراین از ترکیب مساوی دو معیار اول می‌توان وزن صفحات در یک نشست را به‌دست آورد. هر چه این وزن برای یک صفحه بیشتر باشد، علاقه‌ی کاربر به آن صفحه بیشتر بوده است. از میانگین هارمونیک فرکانس و درنگ برای ترکیب این دو معیار و ایجاد معیاری برای تعیین درجه‌ی علاقه‌ی کاربر به یک صفحه‌ی وب در یک نشست استفاده می‌کنیم:

$$Interest\,(Page) = \frac{2 \times Frequency\,(Page) \times Duration\,(Page)}{Frequency\,(Page) + Duration\,(Page)} \qquad (۵-۴)$$

با این نوع ترکیب فرکانس و درنگ، زمانی جذابیت صفحه برای کاربر زیاد خواهد بود که هر دو معیار درنگ و فرکانس برای آن صفحه بالا باشد. میزان جذابیت نیز نرمال شده و مقدار آن بین ۰ و ۱ می‌باشد، نرمال کردن معیار جذابیت صفحات، نه تنها برای درک بهتر میزان علاقه‌ی کاربر مفید است، بلکه به این‌صورت استفاده از آن در مراحل بعدی نیز مناسب‌تر است.

همان‌طور که قبلاً هم گفته شد، یک نشست به‌صورت $s_i = \{(p_1, w_{p_1}), (p_2, w_{p_2}), ..., (p_m, w_{p_m})\}$ نشان داده می‌شود، حال با محاسبه‌ی میزان علاقه‌ی کاربر به صفحات می‌توان وزن‌های صفحات را به شکل $w_{p_i} = Interest\,(p_i)$ مقداردهی کرد.

بر اساس وزن هر صفحه در نشست که میزان جذابیت صفحه برای کاربر است می‌توان جذابیت کل نشست را نیز به‌دست آورد. این کار با میانگین ساده از وزن کل صفحات موجود در نشست انجام می‌شود.

$$Interest(Session) = \frac{\sum_{i=1}^{P} w_{p_i}}{P} \qquad (۶-۴)$$

معیار دیگری بر اساس میزان جذابیت نشست و هم‌نواختی یک نشست محاسبه می‌شود که میزان اعتبار یک نشست را مشخص می‌کند. برای ترکیب این دو معیار نیز از میانگین هارمونیک استفاده می‌شود.

$$Validity(Session) = \frac{2 \times Interest(Session) \times Consistency(Session)}{Interest(Session) + Consistency(Session)} \qquad (۷-۴)$$

همان‌طور که در فصل ۳ مفصلاً شرح داده شد، تئوری خطر با تمرکز توجه روی وقایع داخلی یا خارجی که خطرناک هستند و باعث بروز مشکل در سیستم ایمنی می‌شوند، زیرمجموعه‌ی کوچکی(نسبت به مجموعه‌ی همه‌ی سلول‌های بیگانه‌ای که وارد سیستم ایمنی می‌شوند) از سلول‌های بیگانه را انتخاب و برای آن‌ها پاسخ ایمنی ایجاد کند. همچنین در فصل قبل این نکته هم مورد توجه قرار گرفت که سیگنال خطر، بسته به نوع مسئله تعاریف متفاوتی دارد. در الگوریتم پیشنهادی AISWUM، خطر، "معتبر" تعبیر می‌شود، به این معنی که آنتی‌ژن‌هایی (نشست‌هایی) که معتبر هستند و انتظار می‌رود حاوی اطلاعات مفید باشند، آنتی‌ژن معتبر خواهند بود. در AISWUM تئوری خطر با استفاده از معیارهایی که در این بخش تعریف و ارائه شد، مجموعه‌ای از آنتی‌ژن‌ها را که معتبر تشخیص می‌دهد انتخاب و آن‌ها را وارد سیستم می‌کند بنابراین همان‌طور که انتظار داریم با استفاده از این تئوری در سیستم می‌توان هزینه‌ی محاسباتی الگوریتم را تا حد زیادی کاهش داد.

۴-۴-۵ تابع پیوند[۱]

گرچه در بیشتر سیستم‌های ایمنی از فاصله‌ی اقلیدسی استفاده می‌شود، برای بیشتر کاربردهای داده کاوی، مانند خوشه‌بندی اسناد متنی و مجموعه داده‌های دارای ابعاد بالا، فاصله‌ی اقلیدسی معیار مناسبی نیست. برای این نوع کاربردها، به‌دست آوردن فاصله بر اساس معیار شباهت کوسینوسی مناسب‌تر است.

شباهت بین آنتی‌بادی i و آنتی‌ژن j را به شکل ارائه شده در رابطه‌ی (۴-۸) محاسبه می‌کنیم:

$$S_{\cos}(antibody_i, antigen_j) = \frac{\sum_{l=1}^{L} antibody_i[l] \times antigen_j[l]}{\sqrt{\sum_{l=1}^{L} antibody_i[l] \sum_{l=1}^{L} antigen_j[l]}} \qquad (۸-۴)$$

به آسانی نشان داده می‌شود که شباهت کسینوسی مرتبط با دو معیار مشهور در بازیابی اطلاعات به نام‌های دقت (precision) و شمول (coverage) است:

$$S_{\cos}(i, j) = \sqrt{prc_{i,j} \times cvg_{i,j}} \qquad (۹-۴)$$

که $prc_{i,j}$، یا دقت در فاز یادگیری، توصیف کننده‌ی دقت یادگیری آنتی‌بادی i، هنگام عرضه‌ی آنتی‌ژن j یا به عبارت دیگر، نسبت تعداد آیتم‌های هماهنگ[۲] بین آنتی‌بادی و آنتی‌ژن (داده‌ی ورودی) به تعداد کل آیتم‌ها در آنتی‌بادی است.

$$prc_{i,j} = \frac{\sum_{l=1}^{L} antibody_i[l] \times antigen_j[l]}{\sum_{l=1}^{L} antibody_i[l]} \qquad (۱۰-۴)$$

---

[۱] Affinity function

[۲] Matched URLs

و $cvg_{i,j}$ یا شمول در فاز یادگیری، که به آن یادآوری[1] هم می‌گویند، توصیف کننده‌ی درجه‌ی تکمیل[2] آنتی‌بادی هنگام عرضه‌ی آنتی‌ژن، یا نسبت تعداد آیتم‌های هماهنگ بین آنتی‌بادی و آنتی‌ژن به تعداد آیتم‌ها در آنتی‌ژن است.

$$cvg_{i,j} = \frac{\sum_{l=1}^{L} antibody_i[l] \times antigen_j[l]}{\sum_{l=1}^{L} antigen_j[l]} \qquad (۱۱-۴)$$

با توجه به (۴-۹) می‌توان نتیجه گرفت که استفاده از شباهت کسینوسی برای تعیین فاصله برای محاسبه‌ی چگالی حول هر آنتی‌بادی که رابطه‌ی مربوط به آن در بخش‌های بعدی خواهد آمد، باعث بهینه شدن هر دو معیار دقت و شمول، از طریق ترکیب آن دو به‌وسیله‌ی میانگین هندسی شده است. این معیار، از آنتی‌بادی‌های طولانی‌تر که به علت طولشان با تعداد بیشتری داده جور[3] می‌شوند، طرفداری بیشتری می‌کند [۱۵۱] و بنابراین به دقت لطمه می‌زند. برای حل این مشکل می‌توان قسمتی از معیار شباهت کسینوسی را که مربوط به شمول است، حذف کرده و از دقت تنها به عنوان معیار شباهت استفاده نمود. در این حالت نیز به آنتی‌بادی‌های بسیار کوتاه بهای بیشتری داده می‌شود و شمول کاملاً فراموش می‌شود. بنابراین از ترکیب دیگری غیر از میانگین‌گیری هندسی، برای ترکیب دقت و شمول استفاده می‌شود. این ترکیب بدبینانه به صورت زیر است:

$$S_{new}(i,j) = \min\{prc_{i,j}, cvg_{i,j}\} \qquad (۱۲-۴)$$

فاصله‌ی بین آنتی‌ژن و آنتی‌بادی از رابطه‌ی (۴-۱۳) به‌دست می‌آید.

$$d_{i,j}^2 = 1 - S_{new}(i,j) \qquad (۱۳-۴)$$

---

[1] Recall

[2] Completeness

[3] Match

۴-۴-۶ مراحل الگوریتم

در این بخش، بدنه‌ی اصلی الگوریتم و فرآیندهای لازم را توضیح خواهیم داد.

۴-۴-۶-۱ تعریف‌ها

در محیطی پویا، مانند محیط زندگی انسان، آنتی‌ژن‌ها بطور پیوسته وارد شبکه‌ی ایمنی می‌شوند. الگوریتم پیشنهادی در این پایان‌نامه نیز، برای یک محیط پویا طراحی شده است و فرض بر اینست که آنتی‌ژن‌ها (نشست‌ها) بطور پیوسته به سیستم عرضه می‌شوند. با عرضه‌ی هر آنتی‌ژن، معیارهای سطح تحریک و محدوده‌ی تاثیر (IZ)[۱] به‌روز رسانی می‌شوند.

اندازه‌ی IZ بر حسب تابع وزن که خود بر اساس فاصله‌ی داده‌ی ورودی (آنتی‌ژن) تا آنتی‌بادی محاسبه می‌شود، به‌دست می‌آید.

تابع وزن:

برای iامین آنتی‌بادی، $i = 1,2,\ldots,N_B$ ، وزن تاثیر آنتی‌ژن jام، به‌صورت زیر محاسبه می‌شود.

$$w_{ij} = e^{-(\frac{d_{ij}^2}{2\sigma_{ij}^2})} \qquad (۱۴-۴)$$

که در رابطه‌ی (۴-۱۴)، d_{ij}^2 فاصله‌ی آنتی‌ژن jام تا آنتی‌بادی iام است که برابر با $(1 - S_{new}(i,j))$ می‌باشد.

σ_{ij}^2 پارامتر محدوده‌ی تاثیر است که نرخ کاهش وزن در راستای ابعاد مکانی را کنترل می‌کند و بنابراین سایز منطقه‌ی تاثیر، حول یک آنتی‌بادی را بعد از عرضه‌ی آنتی‌ژن jام با σ_{ij}^2 نشان می‌دهند.

---

[۱] Influence Zone

نمونه‌هایی از داده‌های ورودی که خارج از این محدوده قرار می‌گیرند، نمونه‌های دورافتاده[۱] در نظر گرفته می‌شوند.

منطقه‌ی تاثیر:

آنتی‌بادی iام، یک منطقه‌ی تاثیر متغیر به نام IZ_i دارد. شعاع این منطقه‌ی تاثیر به‌صورت پویا و بعد از عرضه‌ی هر آنتی‌ژن دوباره محاسبه می‌شود. بنابراین معیار، نمونه‌های دورافتاده به راحتی شناسایی می‌شوند. نقطه‌های داده‌ای که خارج از این منطقه‌ی تاثیر قرار می‌گیرند و یا وزن کمی دارند ($w_{ij} < w_{\min}$)، دورافتاده محسوب می‌شوند.

سطح تحریک:

بعد از عرضه‌ی J آنتی‌ژن به سیستم، سطح تحریک آنتی‌بادی iام به‌صورت چگالی جمعیت آنتی‌ژن حول آنتی‌بادی مورد نظر تعریف می‌شود.

$$s_{iJ} = \frac{\sum_{j=1}^{J} w_{ij}}{\sigma_{iJ}^2} \qquad (۱۵-۴)$$

از طریق قرار دادن $\frac{\partial s_{iJ}}{\partial \sigma_{iJ}^2} = 0$ ، معادله‌ی مربوط به به‌روزرسانی مقدار شعاع محدوده‌ی تاثیر به‌صورت رابطه‌ی (۴-۱۶) به‌دست می‌آید.

$$\sigma_{iJ}^2 = \frac{\sum_{j=1}^{J} w_{ij} d_{ij}^2}{2\sum_{j=1}^{J} w_{ij}} \qquad (۱۶-۴)$$

---

[۱] Outlier

جهت بهینه‌کردن هزینه‌ی زمانی و فضایی، دو رابطه‌ی (۴-۱۵) و (۴-۱۶) را به‌صورت معادلات افزایشی زیر تعریف می‌کنیم:

$$s_{iJ} = \frac{W_{iJ-1} + w_{iJ}}{\sigma_{iJ}^2} \qquad (۱۷-۴)$$

$$\sigma_{iJ}^2 = \frac{\sigma_{iJ-1}^2 W_{iJ-1} + w_{iJ} d_{iJ}^2}{2(W_{iJ-1} + w_{iJ})} \qquad (۱۸-۴)$$

که $W_{iJ-1} = \sum_{j=1}^{J-1} w_{ij}$، مجموع تاثیر آنتی‌ژن‌های قبلی بر روی آنتی‌بادی iام است.

با وارد شدن آنتی‌ژن جدید، با استفاده از مقادیر قبلی W_{iJ-1} و σ_{iJ-1}^2 و اضافه کردن سهم آنتی‌ژن جدید، مقادیر s_{iJ} و σ_{iJ}^2 محاسبه می‌شوند.

۴-۴-۶-۲ وارد کردن وزن صفحات در معادلات شبکه‌ی ایمنی

در اکثر روش‌های پیشین برای کاوش داده‌های دسترسی کاربران به وب، تفاوتی بین صفحاتی که توسط کاربر ملاقات شده است، قائل نشده‌اند، در حالی‌که احتمال اینکه صفحات ملاقات شده توسط کاربر در یک نشست، دارای درجه‌ی اهمیت و جذابیت مساوی برای کاربر باشد، تقریبا صفر است. گاهی کاربر ممکن است وارد صفحه‌ای شده باشد و بعد از وارد شدن متوجه شود که این صفحه ارزشی برای او ندارد. این مسئله باعث می‌شود صفحات نامرتبط درون نشست ثبت گردد. بنابراین در نظر گرفتن همه‌ی صفحات موجود در نشست با وزن یکسان در فرآیند یادگیری، کار معقولی نخواهد بود.

در بخش ۴-۴-۴، معیارهای جذابیت برای صفحات و جذابیت برای نشست‌ها تعریف شده، و از آن‌ها تنها در فاز تئوری خطر استفاده شد. می‌توان از این وزن‌ها در فرآیند یاگیری پروفایل‌های جذاب و مکرر نیز استفاده کرد، ولی آیا این وزن‌ها به همین صورت برای وارد شدن به فرآیند یادگیری مناسب‌اند و اگر مناسب هستند، چگونه این وزن‌ها را به الگوریتم خود بیافزاییم؟

در [۱۵۲] مباشر[۱] و همکاران با محاسبه‌ی مدت درنگ میانگین و انحراف استاندارد برای همه‌ی درخواست‌های یک صفحه، یک آستانه‌ی مدت درنگ تعیین کرده‌اند. اگر مدت درنگ بر روی یک صفحه بیش از این حد آستانه بود، صفحه جذاب و در غیراینصورت صفحه نامرتبط محسوب می‌شد(یک تصمیم باینری). مشکل این روش اینست که درجه‌ی جذابیت برای صفحه‌ها تعیین نمی‌شد. در معیارهای وزنی که در این رساله تعریف شده است این مشکل وجود ندارد و به‌وسیله‌ی آن‌ها درجه‌ی مناسبی از میزان جذابیت صفحه به‌دست خواهد آمد.

در [۱۵۳] وزن مجموعه آیتم از دو روش جمع وزن آیتم‌ها و میانگین وزن آیتم‌ها به‌دست آمده است و پشتیبانی وزن‌دار یک مجموعه آیتم از طریق حاصل ضرب پشتیبانی مجموعه آیتم در وزن مجموعه آیتم تعریف شده است.

در همه‌ی روش‌های پیشین، وزن یک آیتم در کل داده‌های ورودی ثابت در نظر گرفته می‌شود، در صورتیکه هر آیتم در نشست‌های متفاوت دارای وزن‌های متفاوتی است.

همان‌طور که گفته شد، در بخش ۴-۴-۴، معیارهای جذابیت برای صفحات و جذابیت برای نشست‌ها تعریف شده، و از آن‌ها تنها در فاز تئوری خطر استفاده شد. می‌توان از این وزن‌ها در فرآیند یاگیری پروفایل‌های جذاب و مکرر نیز استفاده کرد. با این‌کار آیتم‌های مهم‌تر، وزن بیشتری خواهند داشت به این‌صورت اضافه کردن وزن آیتم‌ها در الگوریتم دو تاثیر خواهد داشت، اضافه کردن وزن آیتم‌ها به‌صورت وزن کل نشست باعث می‌شود مجموعه آیتم‌هایی (نشست‌هایی) که مهم هستند ولی فرکانس نسبتاً کمتری دارند (یعنی کاربران محدودی آن‌ها را در مسیر بازدید خود از وب سایت ملاقات کرده‌اند ولی آن تعداد محدود کاربرها به منظور هدفی خاص و با توجه بیشتری، صفحات مورد بحث را بررسی کرده‌اند) را بتوان استخراج کرد. اضافه کردن وزن تک تک آیتم‌ها (صفحات درون نشست‌ها) تاثیر دیگری خواهد داشت. دو نشست (آنتی‌ژن) را در نظر بگیرید که دارای طول یکسان هستند، می‌خواهیم میزان شباهت این دو آنتی‌ژن را با یک آنتی‌بادی محاسبه کنیم، تصادفاً تعداد

---

[۱] Mobasher

صفحات مشترک بین آنتی‌ژن یک و آنتی‌بادی با تعداد صفحات مشترک بین آنتی‌ژن دو و آنتی‌بادی مساوی است، بنابراین هر دو آنتی‌ژن بطور مساوی آنتی‌بادی را تحریک می‌کنند، در صورتیکه صفحات مشترک بین دو آنتی‌ژن و آنتی‌بادی متفاوت هسند و وزن صفحات نیز در دو آنتی‌ژن یکسان نیست. در حالت ایده‌آل این دو آنتی‌ژن نباید تاثیر یکسانی بر آنتی‌بادی داشته باشند. بنابراین با اضافه کردن وزن صفحات بطور مجزاء می‌توان به بهینه‌شدن الگوریتم کمک شایانی کرد.

در الگوریتم پیشنهادی از معیار اعتبار نشست که از ترکیب هم‌نواختی و جذابیت نشست به‌دست آمده است به عنوان وزن مجموعه آیتم و از معیار جذابیت که برای صفحات تعریف شد به عنوان وزن هر آیتم در فرآیند یادگیری استفاده خواهد شد.

برای اضافه کردن تاثیر وزن صفحات در معیار شباهت بین آنتی‌بادی و آنتی‌ژن دو رابطه‌ی (۴-۱۰) و (۴-۱۱) به‌صورت زیر تغییر می‌کند:

$$prc_{i,j} = \frac{\sum_{l=1}^{L}(antibody_i[l] \times (w_l \times antigen_j[l]))}{\sum_{l=1}^{L} antibody_i[l]} \qquad (۱۹-۴)$$

$$cvg_{i,j} = \frac{\sum_{l=1}^{L}(antibody_i[l] \times (w_l \times antigen_j[l]))}{\sum_{l=1}^{L} antigen_j[l]} \qquad (۲۰-۴)$$

و برای اضافه کردن وزن آنتی‌ژن(نشست) Jام ($w_{validity}(J)$) رابطه‌ی (۴-۱۷) را به صورت زیر تغییر می‌دهیم:

$$ws_{iJ} = \frac{W_{iJ-1} + w_{iJ}}{\sigma_{i,J}^2} \times w_{validity}(J) \qquad (۲۱-۴)$$

و به تبع آن فرمول (۴-۱۸) به فرمول (۴-۲۲) تبدیل می‌شود:

$$\sigma_{iJ}^2 = \frac{\sigma_{iJ-1}^2 W_{iJ-1} + w_{iJ} d_{iJ}^2}{2(W_{iJ-1} + w_{iJ})} \times w_{validity} \qquad (۲۲-۴)$$

۴-۴-۶-۳ تحریک[۱] و سرکوب[۲] پویا

طبق شبکه‌ی ایمنی مصنوعی، آنتی‌بادی‌ها بر روی یکدیگر تاثیر مثبت و منفی دارند، که این تاثیر بر میزان سطح تحریک آنتی‌بادی درون شبکه تاثیر می‌گذارد.

بنابراین تاثیر مثبت آنتی‌بادی‌ها بر یکدیگر را با ضریب α به رابطه‌ی (۴-۲۱) می‌افزاییم. به طور مشابه تاثیر منفی آنتی‌بادی‌ها بر یکدیگر را نیز با ضریب β به رابطه‌ی (۴-۲۱) خواهیم افزود.

در نظر گرفتن تاثیر منفی یک راه کنترل رشد و جلوگیری از ایجاد آنتی‌بادی‌های تکراری (جلوگیری از افزونگی داده[۳]) در جمعیت آنتی‌بادی‌هاست.

در صورتی‌که فقط تاثیر منفی آنتی‌بادی‌ها بر یکدیگر را در نظر بگیریم، گرچه کنترل جمعیت به خوبی انجام می‌شود ولی در این حالت شبکه، حافظه نخواهد داشت و شبکه آنتی‌ژن‌های قبلی را فراموش خواهد کرد. در حالت دیگر یعنی فقط در نظر گرفتن تاثیر مثبت، سیستم حافظه‌ی خوبی خواهد داشت ولی رقابتی در شبکه وجود ندارد و این باعث به‌وجود آمدن افزونگی بیشینه می‌شود.

بنابراین برای ایجاد تعادل بین دو حالت بالا، رابطه‌ی (۴-۲۱) که به‌وسیله‌ی آن سطح تحریک کلی آنتی‌بادی iام محاسبه می‌شود، به‌صورت زیر تغییر می‌کند:

$$ws_{iJ} = \frac{W_{iJ-1} + w_{iJ}}{\sigma_{iJ}^2} \times w_{validity}(J) + \alpha \frac{\sum_{n=1}^{N_B} w_{in}}{\sigma_{iJ}^2} - \beta \frac{\sum_{n=1}^{N_B} w_{in}}{\sigma_{iJ}^2} \qquad (۲۳-۴)$$

و رابطه‌ی (۴-۲۲)، شعاع IZ_i با در نظر گرفتن تاثیر آنتی‌بادی‌های دیگر خواهد شد:

$$\sigma_{iJ}^2 = \frac{1}{2} \frac{\sigma_{iJ-1}^2 W_{iJ-1} + w_{iJ} d_{iJ}^2 + \alpha \sum_{n=1}^{N_B} w_{in} d_{in}^2 - \beta \sum_{n=1}^{N_B} w_{in} d_{in}^2}{W_{iJ-1} + w_{iJ} + \alpha \sum_{n=1}^{N_B} w_{in} - \beta \sum_{n=1}^{N_B} w_{in}} \times w_{validity}(J) \qquad (۲۴-۴)$$

---

[۱] Stimulation

[۲] Suppression

[۳] Data redundancy

۴-۴-۶-۴ فشرده‌سازی شبکه برای کاهش هزینه‌ی محاسبات

به ارتباطات بین آنتی‌ژن‌ها (عوامل خارجی) و آنتی‌بادی‌ها در شبکه‌ی ایمنی، که باعث به‌وجود آمدن عبارت اول در رابطه‌ی (۴-۲۳) می‌شود، ارتباطات برون شبکه‌ای و به ارتباطات بین آنتی‌بادی‌ها در شبکه‌ی ایمنی که باعث به‌وجود آمدن عبارت دوم و سوم در رابطه‌ی (۴-۲۳) می‌شوند، ارتباطات داخل شبکه‌ای گفته می‌شود.

تعداد ارتباطات داخل شبکه به قدری زیاد است که مشکلاتی جدی برای تقریباً همه‌ی شبکه‌های ایمنی موجود که برای یادگیری طراحی شده‌اند، ایجاد می‌کنند [۱۵۴] [۱۵۵].

فرض کنید شبکه‌ی ایمنی را توسط خوشه‌بندی آنتی‌بادی‌های درون شبکه با یک روش خوشه‌بندی با پیچیدگی خطی، فشرده کنیم. در این حالت، شبکه‌ی ایمنی به k زیر شبکه افراز می‌شود، که مراکز این زیرشبکه‌ها دیدی مجمل بر کل شبکه به‌وجود می‌آورند. در این حالت برای محاسبه‌ی ارتباطات سراسری با رزولوشن پایین‌تر، مانند ارتباطات بین آنتی‌بادی‌های بسیار متفاوت، تنها لازم است ارتباطات بین مراکز زیرشبکه‌ها محاسبه شود. برای ارتباطات جزئی‌تر، مانند ارتباطات بین آنتی‌بادی‌های یک شبکه، که به‌هم شبیه‌ترند، ارتباطات بین آنتی‌بادی‌های درون یک زیرشبکه محاسبه می‌شود.

۴-۴-۶-۴-۱ تاثیر فشرده سازی شبکه بر توان عملکرد در مجموعه داده‌های حجیم[۱]

ادعا می‌کنیم که الگوریتم پیشنهادی AISWUM در نرخ فشرده‌سازی متناهی، در مقیاس بالا نیز به خوبی عمل می‌کند.

جمله‌ی بالا به این‌صورت اثبات می‌شود که: از آنجا که در الگوریتم پیشنهادی، با ورود هر نمونه مقادیر لازم به‌صورت افزایشی محاسبه و به‌روز می‌شوند، لزومی به ذخیره‌ی داده‌ها وجود ندارد و تنها آنتی‌بادی‌های شبکه‌ی ایمنی را باید در حافظه حفظ کرد، و بنابراین بیشترین هزینه‌ای که به سیستم

---

[۱] Scalability

تحمیل می‌شود ناشی از محاسبه‌ی همه ارتباطات داخلی در شبکه است. در حالت بدون فشرده سازی، محاسبه‌ی همه‌ی ارتباطات دارای پیچیدگی زمانی درجه ۲ بر اساس سایز شبکه است. اگر شبکه تقسیم به k زیرشبکه با سایز تقریبا یکسان شود، تعداد ارتباطات داخلی در شبکه‌ی ایمنی با N_B آنتی‌بادی از N_B^2 در شبکه‌ی بدون فشرده سازی به $(\frac{N_B}{k})^2$ ارتباط درون زیرشبکه‌ای در شبکه‌ی فشرده تبدیل خواهد شد.

در این حالت مشخص است که اگر $k \to \sqrt{N_B}$ میل کند، پیچیدگی محاسبه‌ی ارتباطات به سمت پیچیدگی خطی میل می‌کند. بطور مشابه، تعداد ارتباطات برون شبکه‌ای مرتبط با هر آنتی‌ژن از N_B در شبکه‌ی غیرفشرده به k در شبکه‌ی فشرده تقلیل پیدا می‌کند.

بنابراین می‌توان با انتخاب مناسب تعداد کلاسترها ($k \approx \sqrt{N_B}$) نرخ فشرده‌سازی را طوری تنظیم کرد که پیچیدگی خطی $O(N_B)$ حفظ شود.

در حالت فشرده، آمار مختصر و مفیدی از آنتی‌بادی‌ها در هر زیرشبکه محاسبه می‌شود که بعدها به جای محاسبه‌ی مکرر تحریک و سرکوب تک تک آنتی‌بادی‌ها در زیرشبکه از این آمار استفاده می‌شود. این آمار به‌طور خلاصه، به شکل فاصله‌ی میانگین بین اعضاء گروه، درجه[۱] و چگالی گروه محاسبه می‌شود.

۴-۴-۶-۴-۲ تاثیر فشرده‌سازی شبکه بر معادلات سطح تحریک و شعاع محدوده‌ی تاثیر

به جای در نظر گرفتن همه‌ی ارتباطات ممکن (N_B^2) بین همه‌ی N_B سلول در شبکه‌ی ایمنی، تنها ارتباطات درون زیرشبکه با N_{B_K} آنتی‌بادی در زیرشبکه‌ی kام که آنتی‌ژن با مرکز آن زیرشبکه، کمترین فاصله را داشته، محاسبه می‌شود. در این پروژه از رویکرد k-means برای خوشه بندی شبکه استفاده شده است.

بعد از فشرده‌سازی، معادلات تحریک آنتی‌بادی iام و شعاع IZ_i از معادله‌ی زیر به‌دست می‌آید:

---

[۱] Cardinality

$$ws_{iJ} = \frac{W_{iJ-1} + w_{iJ}}{\sigma_{iJ}^2} \times w_{validity}(J) + \alpha \frac{\sum_{n=1}^{N_{B_k}} w_{in}}{\sigma_{iJ}^2} - \beta \frac{\sum_{n=1}^{N_{B_k}} w_{in}}{\sigma_{iJ}^2} \qquad (۲۶-۴)$$

که عبارت اول، تحریک آنتی‌بادی iام توسط آنتی‌ژن Jام و عبارت دوم تاثیر مثبت N_{B_K} آنتی‌بادی در زیرشبکه‌ی kام و عبارت سوم تاثیر بازدارنده‌ی N_{B_K} آنتی‌بادی موجود در زیرشبکه‌ی kام بر آنتی‌بادی iام است.

معادله‌ی به‌روزرسانی شعاع IZ_i نیز به‌صورت زیر تغییر می‌کند.

$$\sigma_{iJ}^2 = \frac{1}{2} \frac{\sigma_{iJ-1}^2 W_{iJ-1} + w_{iJ} d_{iJ}^2 + \alpha \sum_{n=1}^{N_{B_k}} w_{in} d_{in}^2 - \beta \sum_{n=1}^{N_{B_k}} w_{in} d_{in}^2}{W_{iJ-1} + w_{iJ} + \alpha \sum_{n=1}^{N_{B_k}} w_{in} - \beta \sum_{n=1}^{N_{B_k}} w_{in}} \times w_{validity}(J) \qquad (۲۷-۴)$$

۴-۴-۶-۴-۲-۱ انتخاب مقادیر برای ضرایب تحریک و سرکوب

مقادیر α و β باید طوری انتخاب شوند که هنگامی‌که در خوشه‌ی مورد نظر همه‌ی آنتی‌بادی‌ها مشابه هم هستند، یعنی تفاوت بین آنتی‌بادی‌ها در آن خوشه بسیار پایین است و تکرر داده‌ها[۱] بالاست، میزان β یعنی ضریب تاثیر سرکوبی آنتی‌بادی‌های درون خوشه، بالا باشد، یا به عبارت دیگر میزان $(\alpha - \beta)$ کم شود و آنتی‌بادی‌ها تاثیر تحریکی کمتری بر یکدیگر داشته باشند. به این منظور ثابت در نظر گرفتن α و تغییر β متناسب با میزان تفاوت بین آنتی‌بادی‌ها و درجه‌ی خوشه‌ی مورد نظر، هدف برآورده می‌شود.

برای به‌دست آوردن رابطه‌ی β با تفاوت داده‌ها در خوشه و درجه‌ی خوشه، ماکزیمم مورد نظر برای ضریب $(\alpha - \beta)$ تعیین و سپس β از رابطه‌ی زیر به دست می‌آید.

$$\beta = \alpha - \max(\alpha - \beta) \times \frac{\sum_{n=1}^{N_{Bk}} \sum_{m=1}^{N_{Bk}} dissimilarity(n,m)}{n \times m} \qquad (۲۵-۴)$$

---

[۱] Data redundancy

که در رابطه‌ی (۴-۲۵) $dissimilarity(n,m)$ عدم شباهت بین آنتی‌بادی nام و آنتی‌بادی mام در خوشه‌ی مورد نظر است.

۴-۴-۶-۵ تکثیر در شبکه‌ی ایمنی

آنتی‌بادی‌ها معادل با نسبت سطح تحریکشان به میانگین تحریک شبکه، تکثیر می‌شوند. برای جلوگیری از رشد ابتدایی آنتی‌بادی‌ها و برای ایجاد یک گنجینه‌ی متنوع، باید مکانیزمی اندیشید تا آنتی‌بادی‌های جدید قبل از بالغ شدن تکثیر نشوند. به همین علت برای هر آنتی‌بادی متغیری به نام تعداد تحریک در نظر می‌گیریم که مشخص می‌کند، آنتی‌بادی چندبار به واسطه‌ی ورود یک آنتی‌ژن تحریک شده و سطح تحریک و شعاع تاثیر مربوط به آن به‌روز شده است. به هر آنتی‌بادی زمانی اجازه‌ی تکثیر داده می‌شود که بیشتر از تعداد معینی تحریک در متغیر مربوط به آن ثبت شده باشد. ضمناً آنتی‌بادی‌هایی که مدت زمان زیادی از آخرین بار تحریک آن‌ها می‌گذرد نباید تحریک کرد، زیرا تحریک نشدن طولانی به این معناست که آنتی‌ژنی اخیراً وارد نشده‌است که آن آنتی‌بادی بتواند با آن مقابله کند بنابراین به یک آنتی‌ژن منسوخ نباید منابع زیادی تخصیص داده شود. به این ترتیب از تکثیر آنتی‌بادی‌های جدید و نابالغ یا پیر جلوگیری می‌شود.

لازم به ذکر است که برای هر آنتی‌بادی زمان ظهور آن به عنوان زمان تولد حفظ می‌شود.

بنابر رابطه‌ی (۴-۲۸) آنتی‌بادی‌ها تحت تکثیر قرار می‌گیرند.

$$N_{clones} = K_{clones} \frac{ws_i}{\frac{\sum_{n=1}^{N_B} s_n}{N_B}} \qquad (۴-۲۸)$$

$$If \quad NumOfStimulation > StimulationNumber_{\min} \quad and \quad Now - LastStimulation < StagnationTime_{\min}$$

زمانیکه جمعیت آنتی‌بادی‌ها (N_B) از حد بیشینه‌ی از قبل تعیین شده‌ی $N_{B_{\max}}$ تجاوز کند،

آنتی‌بادی‌ها را به ترتیب افزایشی سطح تحریک مرتب کرده و $(top(N_B - N_{B_{\max}}))$ آنتی‌بادی را که ضعیف‌ترین و پیرترین آنتی‌بادی‌ها هستند از جمعیت حذف می‌کنیم.

از آنجا که آنتی‌بادی‌هایی که تازه به جمعیت اضافه می‌شوند، زمان کافی نداشته‌اند تا تحت تاثیر آنتی‌ژن‌ها سطح تحریکشان افزایش پیدا کند، باید مکانیزمی به‌وجود آید که هنگام حذف آنتی‌بادی‌های اضافی از جمعیت که در پاراگراف بالا توضیح داده شد، آنتی‌بادی‌هایی که تازه به جمعیت اضافه شده‌اند از بین نروند بلکه آنتی‌بادی‌هایی از بین روند که علی‌رغم داشتن زمان کافی، موفق به شناسایی آنتی‌ژن‌های کافی نشده‌اند، به این منظور و برای حفظ تنوع جمعیت و زمان دادن به آنتی‌بادی‌های جدید که سنشان از یک آستانه‌ی از پیش تعیین شده کمتر است و برای جلوگیری از حذف زودهنگام آن‌ها، موقتاً مقدار تحریک آن‌ها را برابر با مقدار تحریک میانگین شبکه قرار می‌دهیم. همچنین در هر مرحله آنتی‌بادی‌های بلوغ یافته‌ای که دچار رکود شده‌اند، یعنی آنتی‌بادی‌های با سطح تحریک بالا که در مدت $StagnationTime_{\min}$ تحت تحریک قرار نگرفته‌اند در حافظه‌ی ثانویه نوشته و از جمعیت حذف می‌شوند.

۴-۴-۶-۶ جهش در شبکه‌ی ایمنی

بعد از فرآیند تکثیر که در زیربخش قبل توضیح داده شد و طی آن آنتی‌بادی‌ها به نسبت سطح تحریکشان تکثیر شدند. برای ایجاد تنوع در جمعیت جهش نیز بر روی کلون‌ها اتفاق می‌افتد. جهش در الگوریتم‌های تکاملی و مبتنی بر جمعیت مهم است زیرا در جمعیت ایجاد تنوع می‌کند و بنابراین باعث جستجوی دقیق‌تر برای جواب می‌شود. جهش احتمال اینکه جمعیت به یک نقطه‌ی بهینه‌ی محلی همگرا شود را کم می‌کند. ولی استفاده از جهش کور و تصادفی باعث عدم قطعیت و افزایش هزینه‌ی زمانی الگوریتم می‌شود. بنابراین برای بهره‌مندی از فواید جهش و همچنین گریز از تاثیرات منفی آن نوعی جهش هدایت‌شده[۱] برای اعمال بر الگوریتم ارائه شده، طراحی شده است.

---

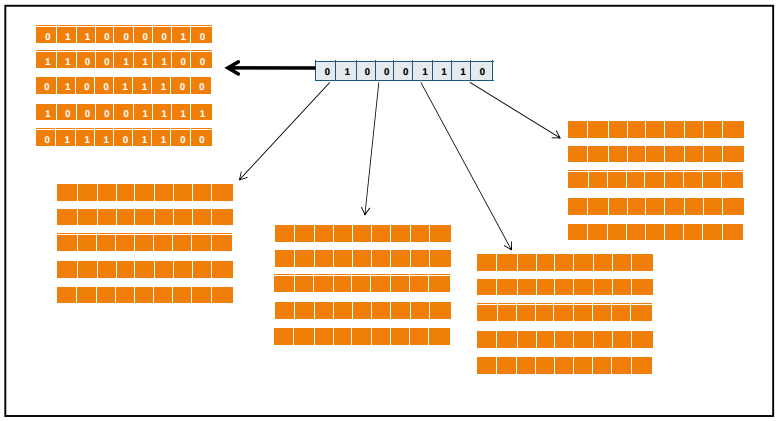
[۱] Directed mutation

این جهش هدایت‌شده، جنبه‌های متفاوتی دارد و با انواع جهش‌های موجود متفاوت است. این جهش به نوعی استفاده از تئوری خطر، با شکل و شمایل دیگری نسبت به شکل مطرح شده در بخش‌های قبلی است.

تئوری خطر در این قسمت، به این‌صورت به حل مسئله کمک می‌کند که هر زمان در جمعیت، سلول‌های ایمنی احساس کردند که نیاز به تولید سلول‌های جدیدی برای مقابله با آنتی‌ژن‌ها است، سیگنال خطر تولید کرده و درخواست کمک می‌کنند. این کمک به وسیله‌ی اضافه شدن آنتی‌بادی‌های جدید توسط سیستم ایمنی پاسخ داده می‌شود.

ولی چه زمانی سلول‌های ایمنی احساس خطر می‌کنند؟، سیگنال خطر چگونه تولید می‌شود و آنتی‌بادی‌های جدیدی که به شبکه اضافه می‌شوند چگونه تولید می‌شوند؟

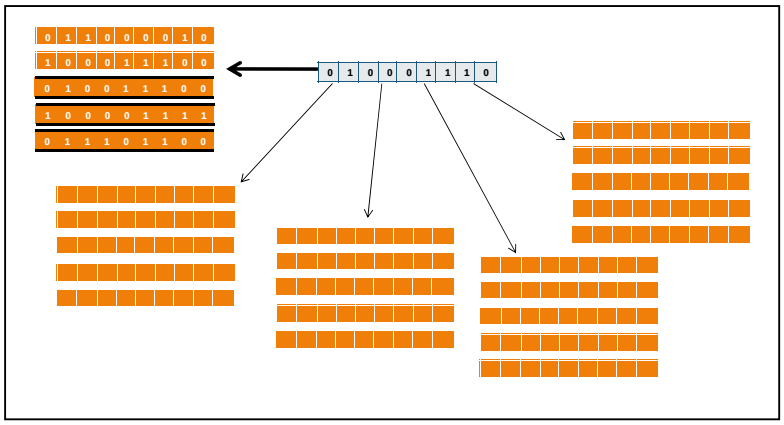
جواب همه‌ی این سوال‌ها در جهش طراحی شده است که برای درک بهتر و توضیح آسان‌تر، جهش هدایت‌شده‌ی طراحی شده با مثالی تشریح می‌شود.



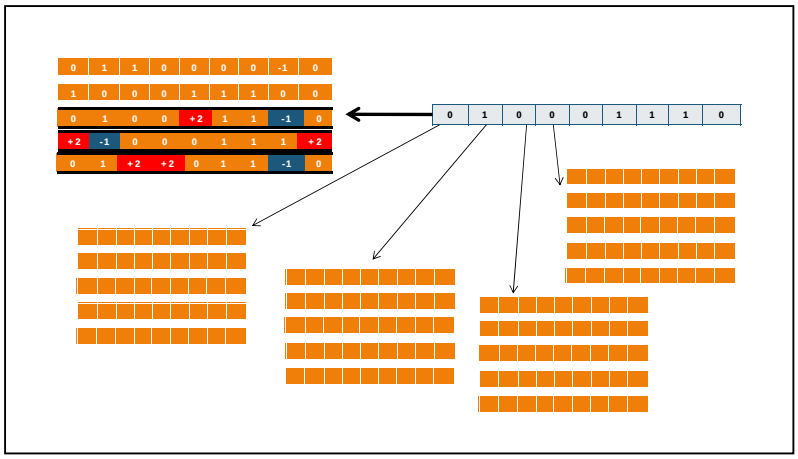
شکل ۴-۲ نمایش اعمال جهش هدایت‌شده در الگوریتم AISWUM (الف).

در شکل ۴-۲ (الف) مشاهده می‌شود که آنتی‌ژنی (آرایه‌ی آبی‌رنگ و منفرد) به شبکه‌ای از آنتی‌بادی‌ها (آرایه‌های نارانجی‌رنگ که در خوشه‌های مختلف دسته‌بندی شده‌اند) عرضه شده است. بعد از عرضه‌ی آنتی‌ژن، خوشه‌ای که مرکز آن با آنتی‌ژن بیشترین وزن را داشته باشد انتخاب می‌شود، در شکل ۴-۲ (ب) مرحله‌ای به نمایش درآمده است که خوشه‌ای از آنتی‌بادی‌ها که بیشترین وزن را با

آنتی‌ژن عرضه شده داشته‌اند فعال شده و در آن خوشه آنتی‌بادی‌هایی که نسبت به آنتی‌ژن تحریک شده‌اند متمایز شده‌اند.



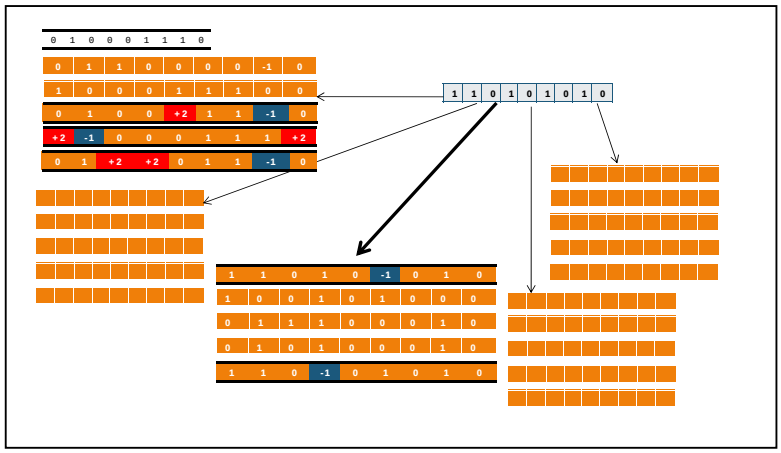
۴-۲ نمایش اعمال جهش هدایت‌شده در الگوریتم AISWUM (ب).



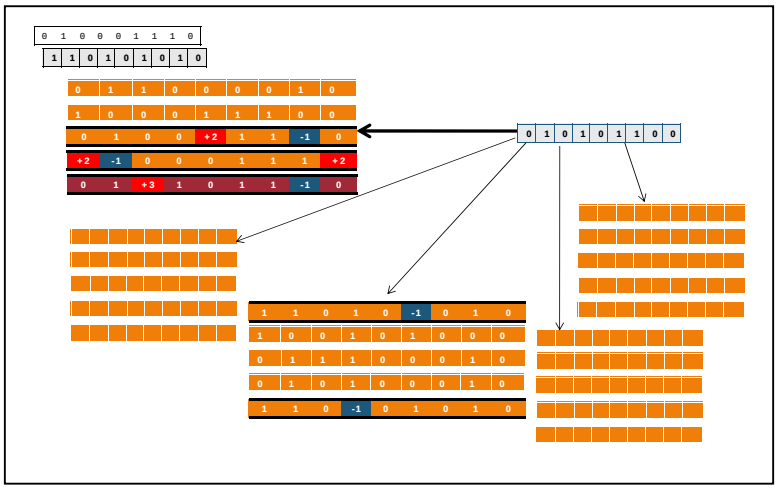
۴-۲ نمایش اعمال جهش هدایت‌شده در الگوریتم AISWUM (ج).

در شکل ۴-۲ (ج)، علاوه بر ثبت تحریک برای آنتی‌بادی‌های درون خوشه‌ی انتخاب شده که نسبت به آنتی‌ژن تحریک شده‌اند که در شکل به‌صورت کادر تیره دور آنتی‌بادی‌های مورد نظر دیده می‌شود، در آرایه‌ی این آنتی‌بادی‌ها تغییراتی نیز قابل مشاهده است. همان‌طور که مشاهده می‌شود، آیتم‌هایی که در آنتی‌ژن و آنتی‌بادی با هم متفاوت بوده‌اند دستخوش تغییر شده‌اند. به خانه‌های متناظر با آیتم‌هایی که در آنتی‌بادی وجود دارد و در آنتی‌ژن وجود ندارد یعنی آنتی‌بادی آن آیتم را کم دارد ۱- واحد اضافه می‌شود (خانه‌های آبی). و به خانه‌های متناظر با آیتم‌هایی که در آنتی‌بادی وجود دارند و در

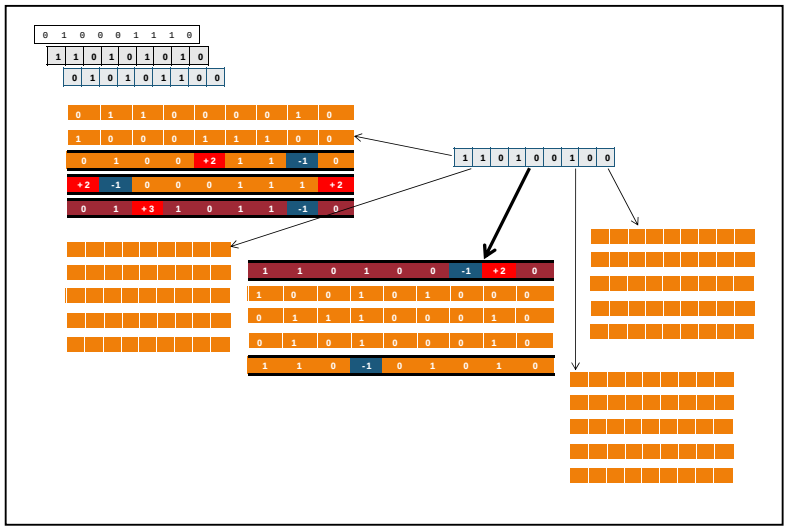
آنتی‌ژن وجود ندارند، یعنی آنتی‌بادی یک واحد در آیتم موردنظر اضافه داشته است، ۱واحد اضافه می‌شود (خانه‌های قرمز).



۴-۲ نمایش اعمال جهش هدایت‌شده در الگوریتم AISWUM (د).



۴-۲ نمایش اعمال جهش هدایت‌شده در الگوریتم AISWUM (ذ).



۴-۲ نمایش اعمال جهش هدایت‌شده در الگوریتم AISWUM (ط).

همان‌طور که می‌بینید این روند با عرضه‌ی آنتی‌ژن‌های دیگر ادامه پیدا می‌کند، با متجاوز شدن اعداد درون آرایه از حد خاصی، می‌توان تصمیم به ایجاد جهش در خانه‌های آرایه گرفت، بنابراین این جهش هدایت شده بطور متناوب بعد از عرضه‌ی هر T آنتی‌ژن برای آنتی‌بادی‌های درون شبکه اجرا می‌شود.

اگر قدر مطلق مقدار یک بیت (ژن) از آنتی‌بادی نسبت به زمان تناوبی که جهش اعمال می‌شود، از حد از پیش تعیین شده‌ای متجاوز شود، به آن بیت، بیت تحت فشار گفته می‌شود.

$$strainedBit(antibody) = \sum_{i=1}^{L} 1 \qquad if\ \left(\frac{|antibody[j]|}{T}\right) > strainBound \qquad (۲۹-۴)$$

نسبت تعداد بیت‌های تحت فشار به کل بیت‌ها نشان دهنده‌ی میزان عدم رضایت آنتی‌بادی است، بنابراین معیاری به‌دست می‌آید به نام عدم رضایت آنتی‌بادی که از رابطه‌ی (۴-۳۰) محاسبه می‌شود:

$$IneptDegree(antibody) = \frac{strainedBit(antibody)}{L} \qquad (۳۰-۴)$$

که در آن صورت کسر تعداد بیت‌های تحت فشار آنتی‌بادی و مخرج تعداد کل بیت‌های آنتی‌بادی است. به این صورت خطر تشخیص داده شده و توسط آنتی‌بادی‌هایی که میزان رضایت آن‌ها کم است، ایجاد می‌شود. بر اثر این سیگنال تعدادی آنتی‌بادی به شبکه اضافه می‌شود. آنتی‌بادی‌هایی که به شبکه اضافه می‌شوند از اعمال جهش نقطه‌ای بر روی بیت (ژن)هایی که تحت فشار هستند، به‌وجود می‌آیند. این جهش بر روی یک کپی از آنتی‌بادی اعمال شده و آنتی‌بادی جدید به شبکه اضافه

می‌شود. حذف آنتی‌بادی تحت جهش قرار گرفته در صورتی‌که راکد شده باشد، منوط می‌شود به زمان حذف آنتی‌بادی‌های اضافه از شبکه.

ضمناً همان‌طور که در بخش‌های پیشین عنوان شد، هر آیتم در آنتی‌ژن دارای وزنی است که نشان‌دهنده‌ی میزان اهمیت یا جذابیت صفحات ملاقات شده در نشست است. با اضافه کردن این وزن در خانه‌های مربوط به آن صفحه یا URL در آرایه‌ی آنتی‌بادی یعنی به جای اضافه کردن ۱- اضافه کردن $Interest\ (Page) * (-1)$ به خانه‌ی مربوطه می‌توان تخمین مناسب‌تری از لزوم وجود آن URL در آنتی‌بادی به‌وجود آورد و بر اساس متجاوز شدن این مقدار از حد آستانه‌ی از پیش تعیین شده، تصمیم به رخداد یا عدم رخداد جهش در ژن‌های آنتی‌بادی گرفت، که اعمال این وزن و ارزیابی نتیجه به کارهای آینده موکول می‌شود.

۴-۴-۶-۷ یادگیری آنتی‌ژن‌های جدید و رابطه با تشخیص نمونه‌های دورافتاده

فوق جهش[۱]، یک مکانیزم اکتشاف قوی در سیستم ایمنی است که به این سیستم طبیعی اجازه می‌دهد تا چگونگی پاسخ به آنتی‌ژن‌های جدیدی که تا به حال هرگز دیده نشده‌اند را آموزش ببیند. از دید محاسباتی، این عملیات بسیار هزینه و زمان‌بر است و پیچیدگی آن نسبت نمایی با تعداد آیتم‌های درون یک مجموعه آیتم (URLهای موجود در آنتی‌بادی) دارد. بنابراین این عملیات در AISWUM پیشنهادی با ایجاد یک نسخه‌ی کپی از آنتی‌ژنی که وارد شده و قادر به فعال کردن شبکه‌ی ایمنی نیست، به عنوان یک آنتی‌بادی در شبکه، مدل می‌شود. آنتی‌ژن j زمانی می‌تواند آنتی‌بادی iام را فعال کند که شبیه به آن باشد و در IZ سلول iام قرار گیرد و بنابراین w_{ij} از یک حد آستانه‌ی کمینه بیشتر باشد.

[۱] Hyper mutation

۴-۴-۶-۸ نمونه‌ی دور افتاده‌ی

یک نمونه‌ی دور افتاده‌ی بالقوه، آنتی‌ژنی است که قادر به فعال کردن شبکه‌ی ایمنی نیست. چنین آنتی‌ژنی شرط مقابل را خواهد داشت: $\forall i = 1,2,..., N_B \quad w_{ij} < w_{\min}$.

برای بهره‌گیری بیشتر از شبکه‌ی ایمنی فشرده، تنها آنتی‌بادی‌هایی که در نزدیک‌ترین زیرشبکه ($\text{subnet } k^{th}$) نسبت به آنتی‌ژن مورد مطالعه قرار دارند، در نظر گرفته می‌شوند. بنابراین شرط به این صورت تغییر می‌کند: $\forall i = 1,2,..., N_{B_k} \quad w_{ij} < w_{\min}$.

به یک نمونه‌ی دیده نشده، نمونه‌ی دورافتاده‌ی بالقوه گفته می‌شود زیرا این نمونه ممکن است واقعاً یک نمونه‌ی دورافتاده باشد و یا ممکن است یک الگوی نوظهور باشد. تنها در طی فرآیند یادگیری پیوسته است که مشخص می‌شود این نمونه، یک نمونه‌ی دورافتاده هست یا خیر. اگر این نمونه دورافتاده باشد، هیچ سلول بالغی در شبکه‌ی ایمنی ایجاد نمی‌کند و از شبکه حذف می‌شود.

۴-۴-۷ شبه کد

شبه کد الگوریتم در شکل ۴-۳ آمده است.

۴-۵ جمع‌بندی

در این بخش الگوریتمی برای استخراج مجموعه آیتم‌های مکرر از داده‌های دسترسی به وب ارائه شد. نام الگوریتم ارائه شده AISWUM (Artificial Immune System for Web Usage Mining) است و نوآوری‌های متعددی در آن مشاهده می‌شود که از آن میان می‌توان به موارد زیر اشاره کرد:

- استفاده از تئوری خطر برای فیلتر کردن نشست‌های نامعتبر (نویز)، گرچه سیستم به علت تحمل نویز بالا قادر به دفع تاثیر آن‌ها در الگوهای استخراجی است، ولی عرضه‌ی این نویزها به سیستم باعث بالا رفتن محاسبات و به تبع آن افزایش زمان اجرا می‌شود.
- استفاده از ارتباطات مفهومی بین صفحات وب، برای تعیین هم‌نواحتی بین URLهای موجود در یک نشست.

1- Fix the maximal population size, $N_{B_{\max}}$.

2- Initialize antibodies using a cross section of input data, $\sigma_i^2 = \sigma_{init}$.

3- Compress immune network into K subnets using five iteration of K-means.

4- Repeat for each antigen $antigen_j$ {

4-1 Compute $validity(antigen_j)$;

4-2 If $validity(antigen_j) <$ validity threshold

4-2-1 discard $antigen_j$ and continue with a new antigen;

4-1 Present $antigen_j$ to each subnet centroid ($C_k, k = 1,...,K$) in network, compute distance and weight.

4-2 Determine the most activated subnet (*ma* subnet) which has maximum w_{kj} .

4-3 If all antibodies in *ma* subnet have $w_{ij} < w_{\min}$ (antigen weak to activate subnet){

4-3-1 create by duplication a new antibody (antibody= $antigen_j$, $\sigma_i^2 = \sigma_{init}$)

}else{

4-3-1 Increment number of stimulation of antibody i;

4-3-2 Compute $antibody_i$ stimulation level (ws_{ij})

4-3-3 Update $antibody_i$ scale value (σ_{ij}^2)

}

4-4 clone antibodies;

4-5 If population size > $N_{B_{\max}}$ {

4-5-1 For each antibody i in network

4-5-1-1 If $antibody_i.age < age_{\min}$ $\quad antibody_i.ws_{iJ} = \sum_{n=1}^{N_B} s_n$;

4-5-2 Sort antibodies in ascending order of their stimulation level;

4-5-3 Kill worst excess $(top(N_B - N_{B_{\max}}))$ antibodies.

}

4-6 mutate antibodies after every T antigen.

4-7 After every T antigen, use five iteration K-means with previous centroid as initial centroid.

}

شکل ۴-۳ شبه کد الگوریتم AISWUM

- استفاده از فرکانس URLهای درون یک نشست و مدت درنگ کاربر بر روی صفحات مربوط به URLهای درون صفحه، هم برای حذف جلسات نامعتبر و بدون مفهوم و هم برای وزن دادن به مجموعه‌های آیتم و ایجاد تفاوت بین آیتم‌های مختلف بر اساس اهمیت آن‌ها.
- ارائه و استفاده از معادلات افزایشی، برای ایجاد یک الگوریتم افزایشی مناسب برای کاربرد وب کاوی...
- داخل کردن وزن صفحات و وزن نشست‌ها در معادلات شبکه.
- طراحی جهش هدایت شده برای بهره‌گیری از نتایج جهش در عین حال جلوگیری از تصادفی بودن آن.

ادعا می‌کنیم که الگوریتم ارائه شده برای کاربردهای بلادرنگ مناسب است و در مقیاس‌های بالا نیز به خوبی عمل می‌کند.

۵

تحلیل نتایج

مقدمه

معیارهای مناسب برای ارزیابی الگوریتم استخراج اطلاعات از داده‌های دسترسی به وب توسط تکنیک سیستم ایمنی مصنوعی

نتایج شبیه‌سازی برای استخراج مجموعه آیتم‌های مکرر از داده‌های دسترسی به وب با یک‌بار عبور از داده‌ها

ارزیابی کارایی استفاده از وزن آیتم، وزن نشست و تئوری خطر

مقایسه‌ی ویژگی‌های الگوریتم با الگوریتم‌های دیگر

جمع‌بندی

"The person who behave sensibly is my tailor, he takes my measures anew everytime he sees me. All the rest go on with their old measurements."

--George Bernard Shaw

فصل ۵: تحلیل نتایج

۵-۱ مقدمه

در بخش قبل، الگوریتمی برای یافتن مجموعه آیتم‌های مکرر از داده‌های دسترسی به وب، بر اساس سیستم ایمنی مصنوعی، ارائه شد.

کمبود الگوریتمی بلادرنگ برای یافتن مجموعه آیتم‌های مکرر در ادبیات این زمینه مشهود است. در این پروژه برای اولین بار سعی در حل مشکل یافتن مجموعه آیتم‌های مکرر که مرحله‌ی پیش‌نیاز برای یافتن قوانین انجمنی است، با استفاده از سیستم ایمنی مصنوعی شده است. تکیه در این پروژه بر کاربرد استخراج اطلاعات از داده‌های دسترسی به وب است و انگیزه‌ی استفاده از سیستم ایمنی مصنوعی در این کاربرد، خصوصیات نهفته در سیستم ایمنی طبیعی است که بسیار مناسب با طبیعت وب به نظر می‌رسد.

ولی تست الگوریتم پیشنهادی برای تخمین نتایج به‌دست آمده، بسیار مشکل است. زیرا به دلیل ذات سیستم استفاده از روش‌های معمول تعیین دقت برای این الگوریتم در دید اولیه غیر قابل اعمال به نظر می‌رسد. از طرف دیگر نمی‌توان نتایج را با الگوریتم‌های دیگر مقایسه کرد، زیرا همان‌طور که در فصل ۲ و در بخش قوانین انجمنی آمد، الگوریتمی وجود ندارد که در کلاس این الگوریتم قرار گیرد و برای داده‌های بسیار حجیم و خلوت مانند داده‌های دسترسی به وب قابل استفاده باشد. به‌علاوه هیچ مجموعه داده‌ی محکی در ادبیات این موضوع گزارش نشده است، زیرا معمولاً اطلاعات دسترسی به وب، خصوصی است و در اختیار دیگران قرار نمی‌گیرد و اگر هم وجود داشت، ارزیابی و مقایسه‌ی متدهای مختلف در چهارچوب داده‌های پویا و حجیم که امکان بیش از یک‌بار پیمایش داده‌ها وجود ندارد بسیار مشکل و شاید غیرممکن است.

یک راه تخمین چنین الگوریتم‌هایی از طریق استفاده از نتایج آن‌ها در کاربردهای متفاوتی است که برای WUM می‌توان متصور شد، به نظر می‌رسد این راه ارزیابی، یعنی استفاده از نتایج الگوریتم در

کاربردی مانند پیشنهاددهی، و ارزیابی نتایج با توجه به بازخورد کاربر، آسان‌ترین و معمول‌ترین راه ارزیابی چنین الگوریتم‌هایی است، بنابراین ما هم برای ارزیابی بخشی از الگوریتم که به روش دیگری قابل ارزیابی نیست از این روش استفاده کرده و با استفاده از شبیه‌سازی یک سیستم پیشنهاددهی بسیار ساده، قسمت‌هایی از الگوریتم را ارزیابی می‌کنیم.

یک راه دیگر نمایش مناسب بودن الگوریتم پیشنهادی، مقایسه‌ی خصوصیات الگوریتم ارائه شده با الگوریتم‌های خوشه‌بندی است. همان‌طور که در فصل ۲، در بخش خوشه‌بندی آمد، الگوریتم پیشنهادی را در دسته الگوریتم‌های خوشه‌بندی نیز می‌توان قرار داد، و از آن‌جا که در ادبیات الگوریتم‌های خوشه‌بندی مانند Scalable KMeans ، DBSCAN و BIRCH وجود دارند که قابل اعمال بر داده‌های حجیم هستند، می‌توان خصوصیات این الگوریتم‌ها را با یکدیگر مقایسه کرد.

برای ارزیابی الگوریتم و برای مونیتور کردن کارکرد الگوریتم لازم است معیارهایی تعریف شود. که در بخش‌های بعد به معرفی و طرز استفاده از این معیارها برای بررسی الگوریتم پرداخته خواهد شد.

۵-۲ معیارهای مناسب برای ارزیابی الگوریتم استخراج اطلاعات از داده‌های دسترسی به وب توسط تکنیک سیستم ایمنی مصنوعی

در مدل ارائه شده برای استخراج مجموعه آیتم‌های مکرر از داده‌های دسترسی به وب، AISWUM، نتیجه بعد از ارائه‌ی آخرین داده‌ی ورودی، آنتی‌بادی‌های شبکه خواهند بود که هر کدام از آن‌ها یک مجموعه آیتم مکرر است که یک مسیر راهبردی وب را که به آن پروفایل هم می‌گویند در خود خلاصه کرده است. هر کدام از آنتی‌بادی‌های نهایی، نماینده‌ی یک مجموعه آیتم است و هر کدام از آن‌ها دارای یک شعاع تاثیر حول خود است که این شعاع همان میزان واریانس در فاصله‌ی داده‌های ورودی مطابق با آنتی‌بادی می‌باشد. این مقدار همچنین معیاری برای خطا یا واریانس است که دقت آنتی‌بادی را به عنوان نماینده‌ی داده‌های ورودی منعکس می‌کند. هر آنتی‌بادی دارای صفتی به نام زمان تولد است که زمان تولد آنتی‌بادی (پروفایل) را منعکس می‌کند، بنابراین از روی سن آنتی‌بادی‌ها

می‌توان به زمان ظهور الگوی استفاده‌ای که آنتی‌بادی نماینده‌ی آن است نیز پی‌برد که از این نکته می‌توان در کاربردهای مونیتور کردن اطلاعات استفاده کرد.

برای تخمین مدل، از معیارهایی که در بازیابی اطلاعات استفاده می‌شوند، سود جسته و کیفیت آنتی‌بادی‌ها را به عنوان نماینده‌ی داده‌های ورودی که اطلاعات در داده‌های ورودی را در خود خلاصه کرده است، ارزیابی خواهیم کرد. این معیارها، دقت و شمول هستند.

که دقت در فاز ارزیابی، توصیف کننده‌ی دقت آنتی‌بادی یادگرفته شده نسبت به پروفایل‌های پایه یا به عبارت دیگر، نسبت تعداد آیتم‌های هماهنگ[۱] بین آنتی‌بادی و پروفایل پایه به تعداد کل آیتم‌ها در آنتی‌بادی است.

شمول در فاز ارزیابی توصیف کننده‌ی درجه‌ی تکمیل[۲] آنتی‌بادی نسبت به پروفایل‌های پایه، یا نسبت تعداد آیتم‌های هماهنگ بین آنتی‌بادی و پروفایل‌های پایه به تعداد آیتم‌ها در پروفایل پایه است. شمول بالا یعنی آن آنتی‌بادی بیشتر آیتم‌های پروفایل پایه را در خود دارد. از طرفی دقت بالا به این معنی است که آنتی‌بادی تنها آیتم‌های موجود در پروفایل پایه را پوشش می‌دهد و دارای آیتم‌های اضافه که بر اثر نویز یا مسائل دیگر به‌وجود آمده‌اند، نیست.

شمول و دقت[۳] در دو جهت متضاد عمل می‌کنند. برای مثال زمانی که شمول بالاترین حد خود را دارد، یعنی یک نود همه‌ی آیتم‌های ممکن را در خود دارد، دقت بسیار پایین خواهد بود. بر عکس این قضیه هم صادق است، نودی که شامل یک آیتم درست باشد، دقت ۱۰۰ درصد خواهد داشت ولی شمول کمی دارد.

استراتژی ارزیابی که استفاده خواهد شد، هم محتوا و هم طبیعت متغیر داده‌ها را در نظر می‌گیرد و دقت و شمولی که محاسبه می‌شود برای کل مدت و با توجه به تغییر مدل بر اثر عرضه‌ی متوالی

---

[۱] Matched URLs

[۲] Completeness

[۳] Precision

داده‌های ورودی محاسبه می‌شود. در این استراتژی آنتی‌بادی‌ها بر اساس گروه‌های متفاوت از نظر محتوا که هر یک از داده‌های ورودی در یکی از این گروه‌ها جای می‌گیرد و از این به بعد به این گروه‌ها گروه‌های پایه می‌گوییم. بررسی می‌شوند.

در کاربرد استخراج اطلاعات از داده‌های دسترسی به وب، گروه‌ها بر اساس پروفایل‌ها و مسیرهای دسترسی متفاوت دسته‌بندی می‌شوند. همان‌طور که در فصل‌های پیشین ذکر شده است الگوریتم پیشنهادی یک الگوریتم بدون ناظر است و گروه‌های پایه در فاز یادگیری مورد استفاده قرار نمی‌گیرند بلکه از این گروه‌های پایه فقط برای ارزیابی مدل استفاده می‌شود.

۵-۲-۱ تعریف معیارهای مورد استفاده برای ارزیابی الگوریتم

می‌توان از معیارهایی که در کاربرد بازیابی اطلاعات استفاده می‌شوند برای ارزیابی مجموعه آیتم‌های به‌دست آمده توسط AISWUM استفاده کرد. معیارهایی که برای ارزیابی نتایج AISWUM استفاده می‌شوند عبارتند از: دقت، شمول.

برای تخمین مناسب بودن پروفایل‌های یادگرفته شده، به خاطر بیاورید که آنتی‌بادی (پروفایل)های ایده‌آل باید با حداکثر دقت و حداکثر شمول با توجه به زیرگروهی که در آن قرار دارند یا همان پروفایل‌های پایه، نماینده‌ی جریان داده‌ی ورودی باشند. به عبارت دیگر توزیع آنتی‌بادی‌های یادگرفته شده توسط مدل، باید منعکس کننده‌ی جریان داده‌های ورودی باشد.

صحت[۱] مدل را می‌توان بر اساس دقت آنتی‌بادی‌های (پروفایل) یادگرفته شده Ab_i، نسبت به پروفایل‌های پایه برای گروه c و کامل‌بودن[۲] را بر اساس شمول آنتی‌بادی‌های (پروفایل) یادگرفته شده Ab_i نسبت به پروفایل‌های پایه برای گروه c به‌دست آورد.

---

[۱] Accuracy

[۲] Completeness

دقت در فاز ارزیابی توصیف کننده‌ی صحت آنتی‌بادی‌های (پروفایل‌های) به‌دست آمده به نمایندگی از پروفایل‌های پایه از نظر تعداد آیتم‌های هماهنگ (URLهای مشترک) بین آنتی‌بادی‌های (پروفایل‌های) یادگرفته شده و پروفایل‌های پایه است.

اگر k، اندیس آیتم‌های (URLهای) هر آنتی‌بادی (پروفایل) و L نشان‌دهنده‌ی کل آیتم‌های موجود (تعداد کل URLها در داده‌های دسترسی به وب) باشد و t اندیس داده‌ای (نشستی) که اخیراً وارد سیستم شده است (s_t)، آنگاه عبارت $s_{t,k}=1$ یعنی tامین داده (s_t)، شامل kامین آیتم (URL) است و اگر c اندیس گروه برای cامین پروفایل پایه g_c باشد، $g_{c,k}=1$ یعنی cامین پروفایل مرجع نیز شامل آیتم (URL) kام است. و اگر $s_{t,k}=0$ و $g_{c,k}=0$ یعنی kامین آیتم (URL) در داده‌ی tام و در پروفایل cام وجود ندارد.

حال اگر i اندیس آنتی‌بادی‌های شبکه، $N_{Ab}(t)$ تعداد کل آنتی‌بادی‌ها بعد از ورود t داده‌ی ورودی و $Ab_i(t)$، iامین آنتی‌بادی (پروفایل) بعد از t ورودی باشد، آنگاه $Ab_{i,k}(t)=1$ یعنی $Ab_i(t)$ شامل kامین آیتم (URL) است.

همه‌ی $N_{Ab}(t)$ آنتی‌بادی موجود در شبکه بعد از عرضه‌ی t داده‌ی ورودی با $AB(t)$ نشان داده می‌شود. در این‌صورت دقت iامین آنتی‌بادی (پروفایل) $Ab_i(t)$ نسبت به cامین پروفایل پایه g_c به‌صورت رابطه‌ی (۵-۱) محاسبه می‌شود.

$$prc(Ab_i(t),g_c)=\frac{\sum_{k=1}^{L}(Ab_{i,k}(t)\times g_{c,k})}{\sum_{k=1}^{L}Ab_{i,k}(t)} \qquad (۵-۱)$$

و شمول iامین آنتی‌بادی (پروفایل) $Ab_i(t)$ نسبت به cامین پروفایل پایه g_c به‌صورت رابطه‌ی (۵-۲) محاسبه می‌شود.

$$cvg(Ab_i(t),g_c)=\frac{\sum_{k=1}^{L}(Ab_{i,k}(t)\times g_{c,k})}{\sum_{k=1}^{L}g_{c,k}} \qquad (۵-۲)$$

معیارهای بالا، کیفیت تک تک آنتی‌بادی‌های به‌دست آمده بعد از عرضه‌ی t داده‌ی ورودی (نشست)، $Ab_i(t)$، را ارزیابی می‌کنند. برای به‌دست آوردن کیفیت کل آنتی‌بادی‌های تولید شده بعد از عرضه‌ی t داده‌ی ورودی (نشست) از نظر دقت و شمول، باید به نحوی کیفیت تک تک آنتی‌بادی‌ها را با هم ترکیب کرد.

برای به‌دست آوردن دقت و شمول کلی سیستم بعد از عرضه‌ی t داده‌ی ورودی (نشست)، به‌ترتیب از روابط (۵-۳) و (۵-۴) استفاده می‌شود:

$$PRC(AB(t), g_c) = \begin{cases} 1 & if \ \max_{i=1}^{N_{Ab}(t)} \{prc(Ab_i(t), g_c)\} > \min prc \\ 0 & otherwise \end{cases} \quad (۵-۳)$$

$$CVG(AB(t), g_c) = \begin{cases} 1 & if \ \max_{i=1}^{N_{Ab}(t)} \{cvg(Ab_i(t), g_c)\} > \min cvg \\ 0 & otherwise \end{cases} \quad (۵-۴)$$

$PRC(AB(t), g_c)$ تشکیل یک ماتریس باینری رابطه‌ای می‌دهد که توزیع آنتی‌بادی‌های دقیق (با کیفیت دقت کمینه‌ی $\min prc$) را برای هر گروه پایه c بعد از عرضه‌ی t ورودی به شبکه نشان‌می‌دهد. $CVG(AB(t), g_c)$ نیز تشکیل یک ماتریس باینری رابطه‌ای می دهد که توزیع آنتی‌بادی‌های کامل (با کیفیت شمول کمینه‌ی $\min prc$) را برای هر گروه پایه c بعد از عرضه‌ی t ورودی به شبکه نشان می‌دهد.

می‌توان دو معیار دقت و شمول را ترکیب کرد تا یک معیار کلی برای کیفیت مدل که نشان‌دهنده‌ی توزیع آنتی‌بادی‌های یادگرفته شده است که به طور همزمان به سطح دقت کمینه‌ی $\min prc$ و سطح شمول کمینه‌ی $\min cvg$ رسیده‌اند، به‌دست آورد.

$$S_{PRC,CVG}(t,c) = PRC(AB(t), g_c) \times CVG(AB(t), g_c) \quad (۵-۵)$$

به منظور ایجاد ارزیابی عینی آنتی‌بادی‌های یادگرفته شده باید معیارهای بالا را برای داده‌های ورودی نیز محاسبه کرد. این‌کار را می‌توان به آسانی با قرار دادن داده‌های ورودی به جای آنتی‌بادی‌ها در

روابط (۵-۳) و (۵-۴) انجام داد، ضمناً برای محاسبه‌ی این معیارها Δt داده‌ی ورودی گذشته در نظر گرفته می‌شوند. بنابراین روابط (۵-۶) و (۵-۷) به صورت زیر به‌دست می‌آیند.

$$PRC'(s_t, g_c, \Delta t) = \begin{cases} 1 & if \exists t' \in [t-\Delta t, t]\{prc(s_{t'}, g_c)\} > \min prc \\ 0 & otherwise \end{cases} \quad (۶-۵)$$

$$CVG'(s_t, g_c, \Delta t) = \begin{cases} 1 & if \exists t' \in [t-\Delta t, t]\{cvg(s_{t'}, g_c)\} > \min cvg \\ 0 & otherwise \end{cases} \quad (۷-۵)$$

با اضافه کردن Δt در معیارهای بالا، حالت داده‌ی ورودی نه تنها در یک لحظه‌ی t بلکه در فاصله‌ی Δt ورودی گذشته به‌دست می‌آید. زمانی‌که $\Delta t = 0$ در نظر گرفته شود تصویر لحظه‌ای از جریان ورودی یا همان توزیع داده‌های ورودی اخذ می‌شود و زمانی‌که Δt مقداری مخالف صفر داشته باشد، "توزیع با حافظه"ی جریان داده‌ی ورودی به‌دست می‌آید..

مشابه به حالت قبل دو معیار تعریف شده در رابطه‌های (۵-۶) و (۵-۷) نیز به‌صورت رابطه‌ی (۵-۸) با هم ترکیب می‌شوند تا یک معیار مرجع کلی که نماینده‌ی "توزیع با حافظه" جریان داده‌ی ورودی در زمان t است، به‌دست آید.

$$S'_{PRC,CVG}(t, c, \Delta t) = PRC'(s_t, g_c, \Delta t) \times CVG'(s_t, g_c, \Delta t) \quad (۸-۵)$$

در نهایت دو معیار کلی تعریف می‌شود که به ترتیب سطح کلی دقت و سطح کلی شمول همه‌ی آنتی‌بادی‌ها را در کل زمانی که داده‌های ورودی جریان دارند و نسبت به همه‌ی پروفایل‌های پایه، تخمین می‌زند. دقت کلی آنتی‌بادی‌های یادگرفته شده نسبت به Δt داده‌ی ورودی به‌صورت رابطه‌ی (۵-۹) به‌دست می‌آید.

$$P(\Delta t) = \frac{\sum_{t=1}^{N_x}\sum_{c=1}^{N_c} S_{PRC,CVG}(t,c) \times S'_{PRC,CVG}(t,c,\Delta t)}{\sum_{t=1}^{N_x}\sum_{c=1}^{N_c} S_{PRC,CVG}(t,c)} \quad (۹-۵)$$

مشابه به رابطه‌ی (۵-۹)، شمول کلی آنتی‌بادی‌ها نسبت به Δt داده‌ی ورودی به‌صورت رابطه‌ی (۵-۱۰) به‌دست می‌آید.

$$C(\Delta t)=\frac{\sum_{t=1}^{N_x}\sum_{c=1}^{N_c}S_{PRC,CVG}(t,c)\times S'_{PRC,CVG}(t,c,\Delta t)}{\sum_{t=1}^{N_x}\sum_{c=1}^{N_c}S'_{PRC,CVG}(t,c,\Delta t)} \qquad (۱۰-۵)$$

که در هر دو رابطه N_c تعداد گروه‌های پروفایل‌های پایه و N_x تعداد کل داده‌های ورودی است.

$P(\Delta t)$ نسبت آنتی‌بادی‌های یادگرفته شده را که نماینده‌ی دقیق Δt داده‌ی ورودی گذشته هستند نسبت به همه‌ی آنتی‌بادی‌های یادگرفته شده و $C(\Delta t)$ نسبت Δt داده‌ی ورودی گذشته را که توسط آنتی‌بادی‌ها به‌طور دقیقی خلاصه شده‌اند، اندازه‌گیری می‌کند.

دو معیار اخیر طوری تعریف شده‌اند که هنگام مقایسه‌ی داده‌های ورودی و آنتی‌بادی‌های یادگرفته شده، گذشته‌ی اخیر نیز مدنظر قرار گرفته شده است. اضافه کردن این حالت یعنی ایجاد نیم‌نگاهی به گذشته‌ی اخیر داده‌های ورودی به این جهت است که علاوه بر این‌که آنتی‌بادی‌های یادگرفته شده باید نسبت به داده‌ی ورودی در لحظه‌ی t وفق پیدا کنند، آنتی‌بادی‌ها لازم است حافظه‌ای برای نگهداری گذشته‌ی اخیر داشته باشند.

ایده‌ی اصلی در فرآیند ارزیابی، مقایسه‌ی توزیع آنتی‌بادی‌های یادگرفته شده با توزیع داده‌های ورودی است. این مقایسه را با در کنار هم قرار دادن توزیع آنتی‌بادی‌های یادگرفته شده از منظر دقت ($PRC(Ab(t),g_c)$) و شمول ($CVG(Ab(t),g_c)$) با توزیع داده‌های ورودی ($S'_{PRC,CVG}(t,c,\Delta t)$) انجام خواهیم داد.

۵-۳ نتایج شبیه‌سازی برای استخراج مجموعه آیتم‌های مکرر از داده‌های دسترسی به وب با یک عبور از داده‌ها

آزمایشات بر روی دو مجموعه داده انجام شده‌اند. مجموعه داده‌ی اول که داده‌های دسترسی به وب سایت ابزارآلات موسیقی (www.hyperreal.org) که وب‌سایتی است که در آن ابزار آلات موسیقی دیجیتال که توسط کارخانجات مختلف تولید شده‌اند و توضیحاتی برای هر دسته از ابزار آمده است.

این مجموعه داده توسط مایک پرکویتز[۱] و اورن اتزیونی[۲] در دانشگاه واشنگتن جمع‌آوری و استفاده شده است. کش کردن صفحات در سایت از کار افتاده است، بنابراین برای هر صفحه حتی اگر قبلاً به آن دسترسی ایجاد شده باشد، درخواست مجزایی صادر می‌شود.

در این مجموعه داده، درخواست‌های مربوط به هر روز در یک فایل جدا ذخیره شده‌اند. هر دسترسی با فرمت خاصی درون لاگ وب در وب سرور ذخیره شده است. چندین خط نمونه از داده‌های ذخیره شده در لاگ وب در شکل ۵-۱ آورده شده است. هر خط شامل شماره‌ایست که به جای IP درخواست کننده قرار گرفته است (برای حفظ حقوق شخصی) (O)، زمان درخواست (T)، URL درخواست شده (U) و URLی که کاربر از آن صفحه به صفحه‌ی درخواست شده منتقل می‌شود (R). هر کدام از این فیلدها با || از هم جدا می‌شوند.

```
O:<origin> || T:<time> || U:<url> || R:<referrer>
O:0000002560 || T:1997/09/12-22:43:00 || U:/ ||
R:http://www.hyperreal.org/
O:0000002560 || T:1997/09/12-22:50:27 || U:/categories/software/ ||
R:http://www.hyperreal.org/music/machines/
O:0000002560 || T:1997/09/12-22:50:38 || U:/categories/software/Windows/
|| R:http://www.hyperreal.org/music/machines/categories/software/
O:0000002560 || T:1997/09/12-22:50:47 ||
U:/categories/software/Windows/V909V03.TXT ||
R:http://www.hyperreal.org/music/machines/categories/software/Windows/
O:0000002560 || T:1997/09/12-22:51:06 || U:/categories/software/Windows/
|| R:http://www.hyperreal.org/music/machines/categories/software/
O:0000002560 || T:1997/09/12-22:51:18 ||
U:/categories/software/Windows/ravemusc.txt ||
R:http://www.hyperreal.org/music/machines/categories/software/Windows/
```

شکل ۵-۱ نمونه‌ای از مسیرهای ذخیره شده در سرور وب ابزارآلات موسیقی

در این پروژه از داده‌های یک هفته از این مجموعه داده استفاده شده است که شامل ۲۲۰۱۴۶ درخواست برای وب‌سایت ابزارآلات موسیقی است. بعد از فاز پیش‌پردازش که در فصل ۲ کاملاً تشریح شده است، ۱۹۵۴۲ نشست از داده‌ها استخراج شده است که از طریق این نشست‌ها به ۴۷۵۶ URL دسترسی ایجاد شده است.

---

[۱] Mike Perkowitz

[۲] Oren Etzioni

مجموعه داده‌ی دوم، داده‌های دسترسی به سرور وب دانشگاه ساسکاچوان[۱] در کانادا است. این داده‌ها توسط ارل فوگل[۲] در دانشگاه ساسکاچوان جمع‌آوری شده است. از داده‌ی ۷ روز دسترسی به این وب سرور برای انجام آزمایشات استفاده شده است. تعداد دسترسی‌ها در این ۷ روز ۴۴۲۹۸ که در طی این مدت به ۱۵۱۹ URL مراجعه شده است. تعداد نشست‌های به‌دست آمده از این داده‌ها ۹۱۸۸ است.

```
202.32.92.47 [01/Jun/1995:00:00:59 -0600] "GET/~scottp/publish.html" 200 271
ix-or7-27.ix.netcom.com[01/Jun/1995:00:02:51-0600]  "GET/~ladd/ostriches.html"
200 205908
ram0.huji.ac.il[01/Jun/1995:00:05:44-0600] "GET/~scottp/publish.html" 200 271
eagle40.sasknet.sk.ca[01/Jun/1995:00:08:06 -0600] "GET/~lowey/" 200 1116
eagle40.sasknet.sk.ca[01/Jun/1995:00:08:19  -0600]  "GET/~lowey/kevin.gif"  200
49649
cdc8g5.cdc.polimi.it[01/Jun/1995:00:11:03-0600]
"GET/~friesend/tolkien/rootpage.html" 200 461
freenet2.carleton.ca[01/Jun/1995:00:16:54-0600]   "GET/~scottp/free.html"   200
5759
```

شکل ۵-۲ نمونه‌ای از مسیرهای ذخیره شده در سرور وب دانشگاه ساسکاچوآن

پارامترهای کنترلی برای انجام آزمایشات در این بخش به‌صورت زیر تنظیم شده‌اند: تعداد کلاسترها برای انجام فشرده‌سازی $k_{cl} = \sqrt{N_B}$ که حالت بهینه برای رسیدن به پیچیدگی زمانی O(N) است، انتخاب و بعد از عرضه‌ی هر $t = 10$ نشست خوشه‌بندی شبکه‌ی آنتی‌بادی‌ها با استفاده از مراکز خوشه‌ی فعلی به عنوان نقاط اولیه انجام می‌شود. وزن کمینه‌ی بین آنتی‌بادی و آنتی‌ژن که باعث تحریک آنتی‌بادی می‌شود $w_{\min} = 0.3$ و ماکزیمم سایز جمعیت $N_{Ab_{\max}}$ که می‌توان آن را تعداد منابع موجود در سیستم در نظر گرفت برای مجموعه داده‌ی اول ۱۵۰ و برای مجموعه داده‌ی دوم ۱۰۰ در نظر گرفته شده‌است.

توجه کنید که اگر سایز جمعیت کم درنظر گرفته شود، الگوریتم برای اجرا به حافظه‌ی کمتری نیاز خواهد داشت ولی تعداد پروفایل‌های کمتری را یاد خواهد گرفت. در صورت زیاد بودن سایز جمعیت، حافظه‌ی بیشتری مورد نیاز خواهد بود و پروفایل‌های بیشتری فراگرفته خواهد شد.

---

[۱] Saskatchewan

[۲] Earl Fogel

همان‌طور که در بخش قبل آمد برای نمایش توانایی یادگیری پیوسته‌ی سیستم پیشنهادی نیاز به مجموعه آیتم‌های مکرری است که برای مجموعه داده‌ی مورد استفاده توسط الگوریتم دیگری به‌دست آمده‌اند تا بتوان از این مجموعه آیتم‌های مکرر به عنوان مجموعه آیتم‌های مکرر پایه استفاده کرد. روش مورد استفاده برای استخراج مجموعه آیتم‌های مکرر و دقت این مجموعه آیتم‌های مکرر حائز اهمیت نیست زیرا بر اساس روش ارزیابی که در فصل پیش مفصلاً تشریح شد، از این مجموعه آیتم‌های مکرر پایه تنها برای مقایسه و نشان دادن وضعیت آنتی‌بادی‌های یادگرفته شده و داده‌های ورودی نسبت به هم و نه نسبت به مجموعه آیتم‌های پایه استفاده می‌شود.

در آزمایشات از مجموعه آیتم‌های پایه‌ای استفاده شده است که توسط الگوریتم Scalable K-Means به‌دست آمده است و در جدول ۵-۱ برخی از این مجموعه آیتم‌های مکرر نشان داده شده است.

جدول۵-۱ برخی از پروفایل‌های به‌دست آمده توسط الگوریتم Scalable K-Means برای مجموعه داده‌ی ابزار آلات موسیقی

۱	/samples.html/ /manufacturers/Moog/Rogue/samples/ /manufacturers/Roland/Juno/samples/ /manufacturers/ARP/Odyssey/samples/
۲	/manufacturers/Doepfer/ /manufacturers/Doepfer/Overview/ /Analogue-Heaven/
۳	/samples.html/ /manufacturers/Yamaha/RX/samples/ /manufacturers/Univox/Micro-Rhythmer-12/samples/ /manufacturers/Casio/RZ-1/samples/
٤	/manufacturers/Roland/TR-909/samples/ /samples.html/ /categories/drum-machines/samples/ /categories/drum-machines/samples/deepsky_kicks/
٥	/manufacturers/ /manufacturers/Yamaha/ /manufacturers/Sequential/ /manufacturers/Yamaha/CS-50/
٦	/manufacturers/Yamaha/RX/samples/ /MMAgent/

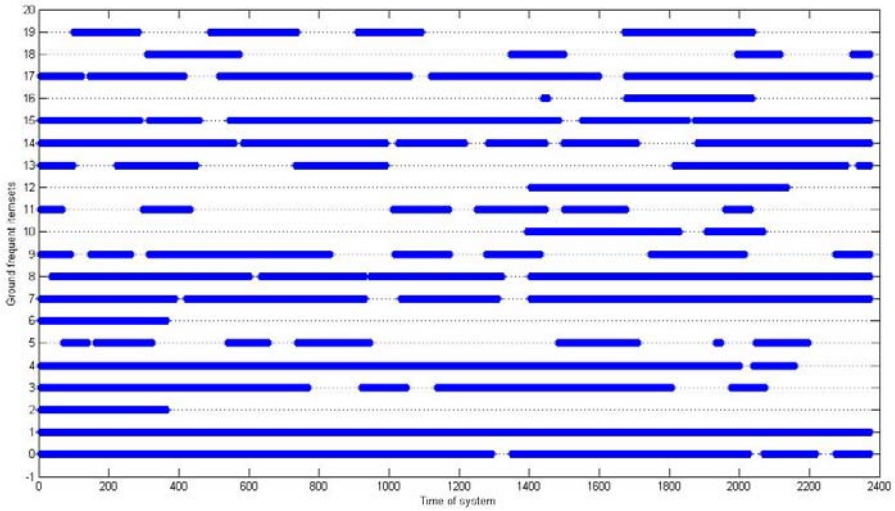
	/manufacturers/Yamaha/MR-10/samples/ /manufacturers/Sequential/Tom/samples/
۷	/manufacturers/Novation/BassStation/i/ /manufacturers/Novation/BassStation/ /manufacturers/Novation/BassStation/novation.BassStation-rack.specs/ /manufacturers/Novation/BassStation/info/
۸	/manufactures/Roland/TR-808/samples/ /manufactures/Roland/TR-606/samples/ /manufactures/Roland/TR-303/samples/ /manufactures/Roland/TR-303/
۹	/categories/do-it-yourself/schematics/digisound.Ring-Mod.schematic/ /manufacturers/Oberheim/Xpander.Matrix-12/ /manufacturers/Oberheim/Xpander.Matrix-12/oberheim.Xpander.specs/ /manufacturers/Oberheim/Xpander.Matrix-12/info/
۱۰	/manufacturers/ /links/ /software.html/ /manufacturers/Novation/DrumStation/ /manufacturers/Novation/DrumStation/info/
۱۱	/manufacturers/Roland/TR-808/samples/ /manufacturers/Emu/Emax/info/ /manufacturers/Electro-Harmonix/info/ /manufacturers/EML/Poly-Box/
۱۲	/samples.html/ /categories/drum-machines/samples/ /categories/drum-machines/samples/deepsky_kicks/ /categories/drum-machines/samples/dpm48.txt/ /categories/drum-machines/samples/Rhythm-Ace.txt/ /categories/drum-machines/samples/mfb512.txt/
۱۳	/manufacturers/ /Analogue-Heaven/ /manufacturers/Moog/ /manufacturers/Moog/images/
۱٤	/Analogue-Heaven/ /manufacturers/Roland/ /manufacturers/Roland/JX/ /manufacturers/Roland/JX/info/

۱۵	/schematics.html/ /categories/do-it-yourself/schematics/ /manufacturers/Roland/TB-303/schematics/ /categories/do-it-yourself/schematics/noisemaking.circuit/
۱۶	/manufacturers/ /manufacturers/Yamaha/ /links/ /links/manufacturers.html/

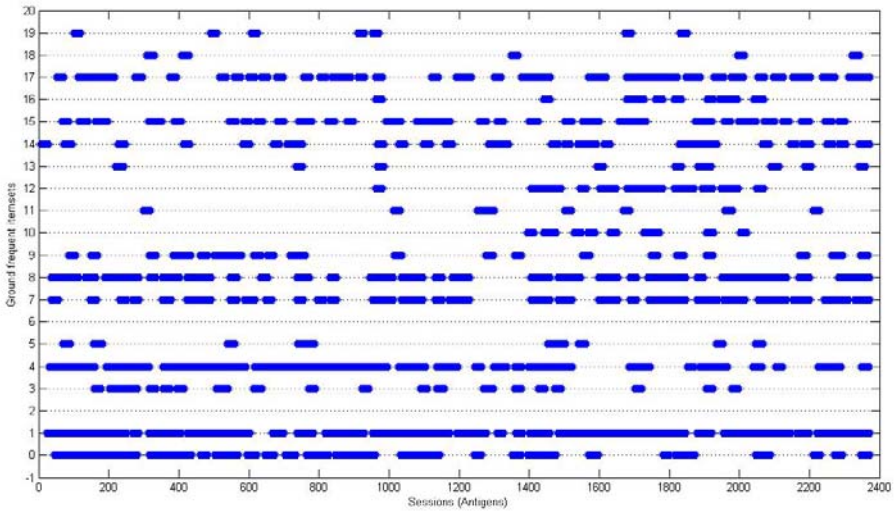
هر چند همان‌طور که گفته شد روش به‌دست آوردن این مجموعه آیتم‌های مکرر پایه اهمیتی در فرآیند ارزیابی ندارد، لازم است دلیل استفاده از SKM برای به‌دست آوردن مجموعه آیتم‌های مکرر پایه عنوان شود که این دلایل عبارتند از سهولت پیاده‌سازی و قابل اعمال بودن آن برای مجموعه داده‌های بزرگ با پیچیدگی زمانی و حافظه‌ای قابل قبول. از طرف دیگر الگوریتم اپریوری به جهت پیچیدگی حافظه‌ای و زمانی بسیار بالا، برای اعمال بر مجموعه داده‌های مورد استفاده غیرقابل اعمال بوده و استفاده از نسخه‌های تغییریافته‌ی اپریوری هم به دلیل پیچیدگی بالای این الگوریتم‌ها و هم به دلیل برطرف شدن نیاز ما با استفاده از SKM مردود شد.

در زیر نتایج به‌دست آمده طبق معیارهای تعریف شده در بخش‌های پیشین برای دو مجموعه داده معرفی شده در بخش‌های پیش را مشاهده می‌کنید.

برای نمایش توانایی یادگیری سیستم، آنتی‌بادی‌هایی را که بعد از عرضه‌ی هر آنتی‌ژن به سیستم، تولید شدند و دقت و شمول آن‌ها نسبت به مجموعه آیتم‌های مکرر پایه بیش از ۰.۴ بوده است ردگیری کردیم. یا به عبارت دیگر بعد از عرضه‌ی هر داده‌ی ورودی که یک نشست معتبر است، آنتی‌بادی‌هایی که موفق شدند هر کدام از ۲۰ پروفایل پایه را یاد بگیرند، دنبال کردیم. و در اشکالی که در زیر آورده شده‌اند که همان ماتریس‌های توزیعی هستند که توسط رابطه‌های (۵-۳)، (۵-۴) و (۵-۵) تعریف شده‌اند، به نمایش درآمده‌اند. در این ماتریس‌ها محور y، به ۲۰ قسمت تقسیم شده است که هر کدام از این قسمت‌ها مربوط به یکی از مجموعه آیتم‌های مکرر پایه است. محور x، محور زمان است، که هر واحد زمانی با ورود یک داده‌ی ورودی مشخص می‌شود.

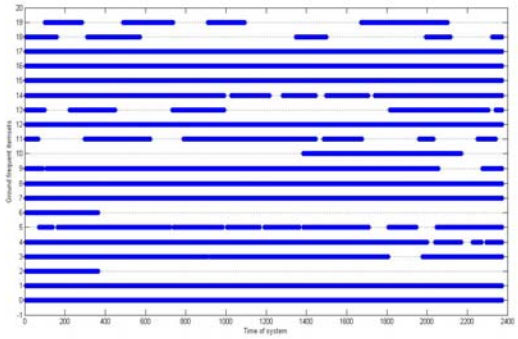


شکل ۵-۳ توزیع آنتی‌بادی‌های دقیق و کامل که دقت و شمول آن‌ها در مقایسه با مجموعه آیتم‌های مکرر پایه از ۰.۴ بیشتر است $S_{PRC,CVG}(t,c)$.

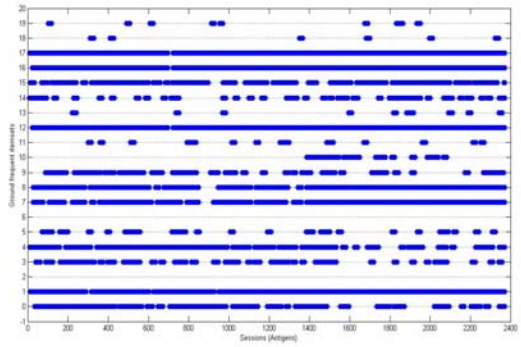


شکل ۵-۴ توزیع داده‌های ورودی که دقت و شمول آن‌ها در مقایسه با مجموعه آیتم‌های مکرر پایه از ۰.۴ بیشتر است $S'_{PRC,CVG}(t,c,0)$.

بنابراین اگر در زمان t، یعنی بعد از عرضه‌ی tامین داده‌ی ورودی، در آنتی‌بادی‌هایی که در سیستم وجود دارند، آنتی‌بادی‌ای باشد که دقت و شمول آن با مجموعه آیتم مکرر پایه‌ی i بیش از ۰.۴ باشد در نقطه‌ی (t,i) در شکل یک علامت * قرار می‌گیرد.

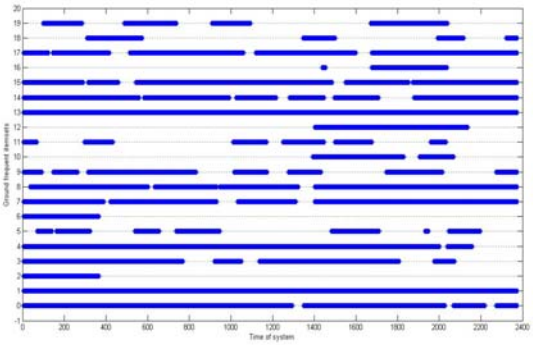


(ب)

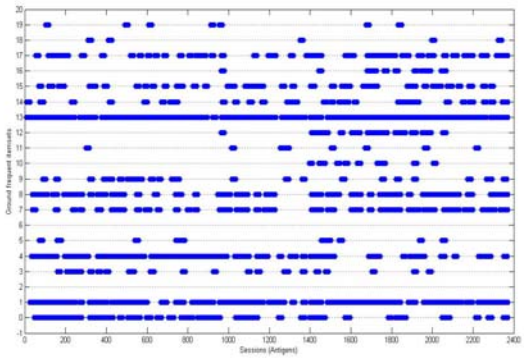


(الف)

شکل ۵-۵ الف)توزیع داده‌های ورودی با دقت بالاتر از ۰.۴ نسبت به مجموعه آیتم‌های پایه ب)توزیع آنتی‌بادی‌های دقیق (با دقت بالاتر از ۰.۴ نسبت به مجموعه آیتم‌های پایه)



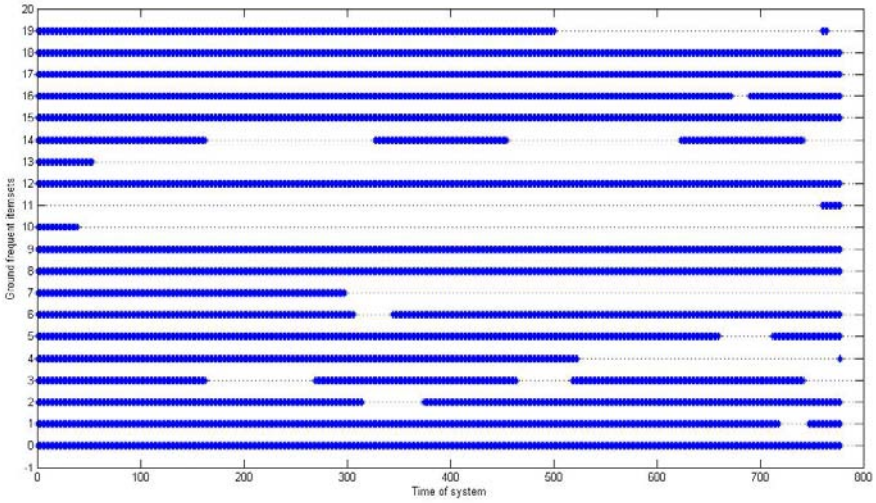
(ب)



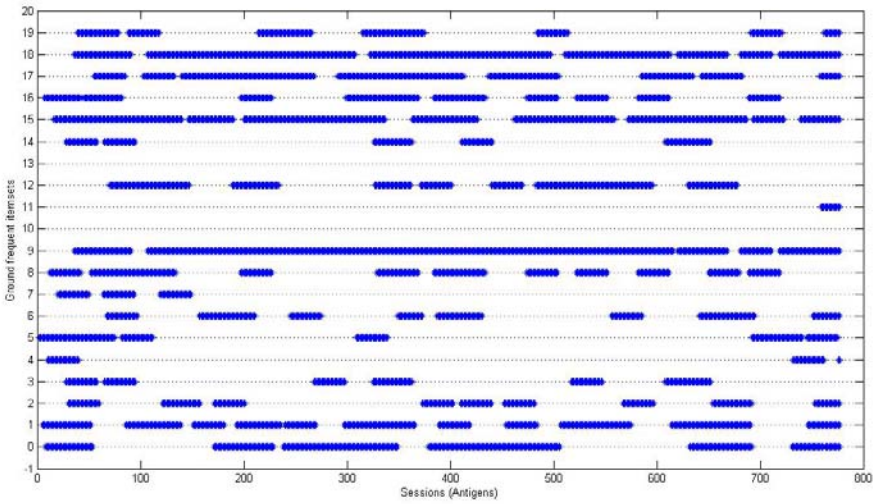
(الف)

شکل ۵-۶ الف)توزیع داده‌های ورودی با شمول بالاتر از ۰.۴ نسبت به مجموعه آیتم‌های پایه ب)توزیع آنتی‌بادی‌های کامل (با شمول بالاتر از ۰.۴ نسبت به مجموعه آیتم‌های پایه)

متناسب با هر شکل که نشان دهنده‌ی توزیع آنتی‌بادی‌های با دقت و شمول بالاست، توزیع داده‌های ورودی نیز نسبت به مجموعه آیتم‌های مکرر پایه نشان داده شده است $S'_{PRC,CVG}(t,c,0)$. شکل ۵-۴ نشان می‌دهد که داده‌های نشست‌ها دارای نویز زیادی هستند، ترتیب درخواست‌ها از نظم خاصی پیروی نمی‌کنند و الگوی نشست‌هایی که مربوط به یک گروه دسترسی هستند چنان متغیرند که ردگیری و کشف مجموعه آیتم‌های مکرر به‌صورت افزایشی بسیار مشکل است.



شکل ۵-۷ توزیع آنتی‌بادی‌های که دقت و شمول آن‌ها در مقایسه با مجموعه آیتم‌های مکرر پایه از ۰.۴ بیشتر است $S_{PRC,CVG}(t,c)$.

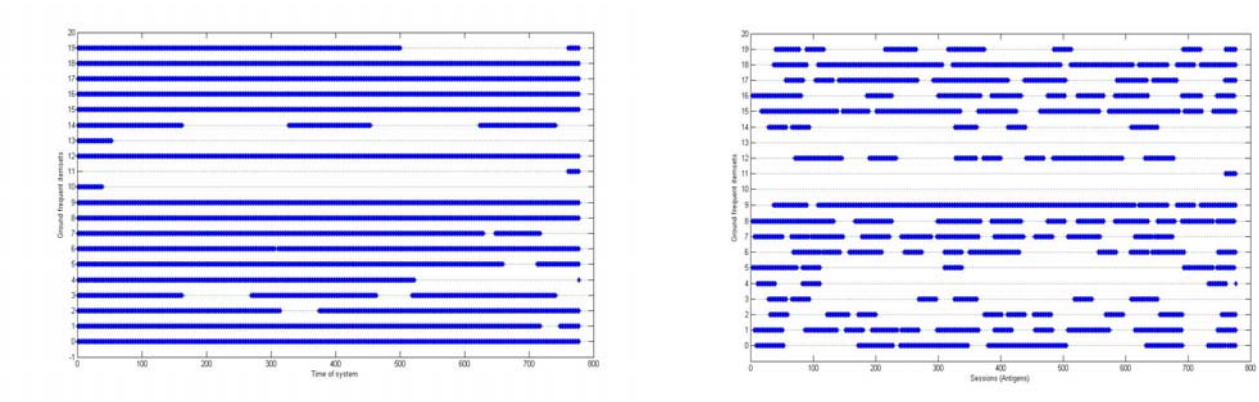


شکل ۵-۸ توزیع داده‌های ورودی که دقت و شمول آن‌ها در مقایسه با مجموعه آیتم‌های مکرر پایه از ۰.۴ بیشتر است $S'_{PRC,CVG}(t,c,0)$.

همین‌طور که مشاهده می‌کنید، شکل کلی توزیع آنتی‌بادی‌ها و آنتی‌ژن‌ها بسیار مشابه هستند و این شباهت گواه این حقیقت است که با یک‌بار عبور از داده‌های ورودی، آنتی‌بادی‌های تولید شده به خوبی توانسته‌اند داده‌های ورودی را در خود خلاصه کنند.

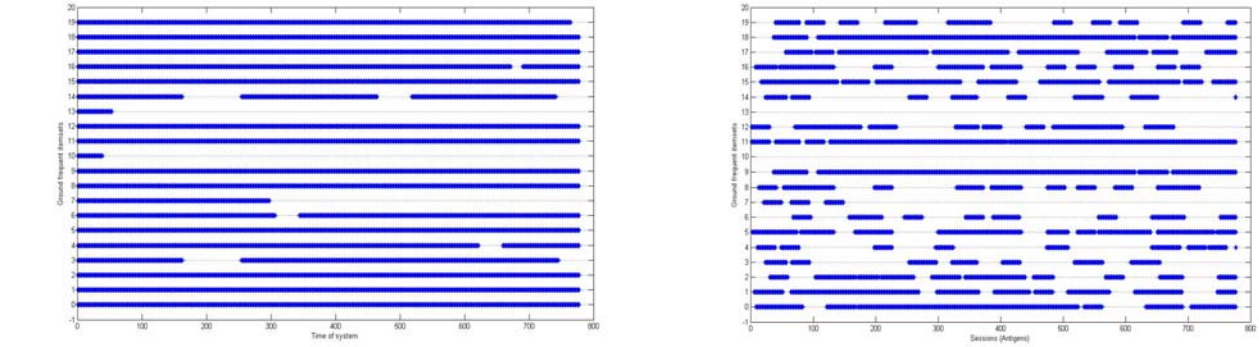
یک عبور از روی همه‌ی ۷ روز داده‌ی لاگ سرور ابزارآلات موسیقی با کد C++ غیربهینه حدود کمتر از ۶ دقیقه به طول انجامیده است.

در شکل‌های ۵-۷ تا ۵-۱۰ نتایج الگوریتم برای مجموعه داده‌ی دانشگاه ساسکاچوان را مشاهده می‌کنید.



(ب) (الف)

شکل ۵-۹ الف)توزیع داده‌های ورودی با دقت بالاتر از ۰.۴ نسبت به مجموعه آیتم‌های پایه ب)توزیع آنتی‌بادی‌های با دقت بالاتر از ۰.۴ نسبت به مجموعه آیتم‌های پایه

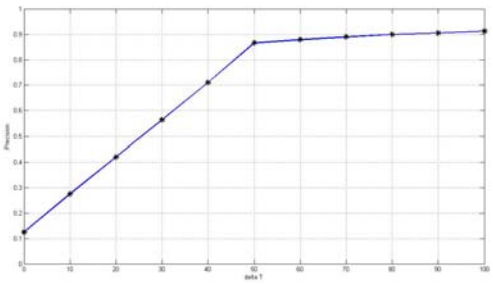


(ب) (الف)

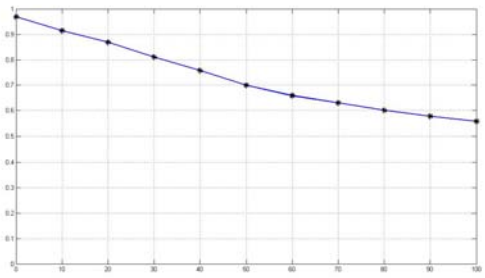
شکل ۵-۱۰ الف)توزیع داده‌های ورودی با شمول بالاتر از ۰.۴ نسبت به مجموعه آیتم‌های پایه ب)توزیع آنتی‌بادی‌های با شمول بالاتر از ۰.۴ نسبت به مجموعه آیتم‌های پایه

در نهایت شکل‌های ۵-۱۱ و ۵-۱۲ معیارهای تعریف شده در روابط (۵-۹) و (۵-۱۰)، $P(\Delta t)$ نسبت آنتی‌بادی‌های یادگرفته شده را که نماینده‌ی دقیق Δt داده‌ی ورودی گذشته هستند نسبت به همه‌ی آنتی‌بادی‌های یادگرفته شده و $C(\Delta t)$ نسبت Δt داده‌ی ورودی گذشته را که توسط آنتی‌بادی‌ها

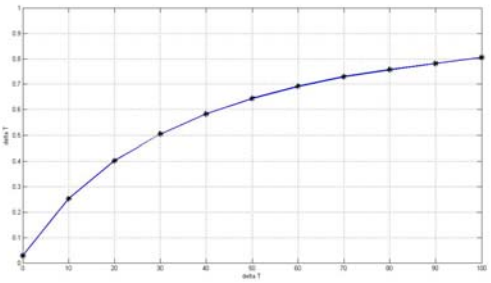
به‌طور دقیقی خلاصه شده‌اند، برای Δtهای متفاوت نشان می‌دهند. انتظار می‌رود با افزایش Δt یعنی با توجه به گذشته‌ی دورتر و افزایش برد نگاه به گذشته، شمول آنتی‌بادی‌ها کم شود، زیرا در این حالت آنتی‌بادی‌ها را با بازه‌ی بزرگ‌تر و با داده‌های ورودی قدیمی‌تری مقایسه می‌کنیم. و انتظار می‌رود دقت افزایش یابد زیرا آنتی‌بادی‌ها نه تنها داده‌ی ورودی در زمان حال را یاد میگیرند بلکه نتیجه‌ی یادگیری پیوسته هستند و داده‌های ورودی گذشته را نیز در حافظه‌ی خود دارند، بنابراین با در نظر گرفتن داده‌های ورودی گذشته، دقت افزایش می‌یابد.



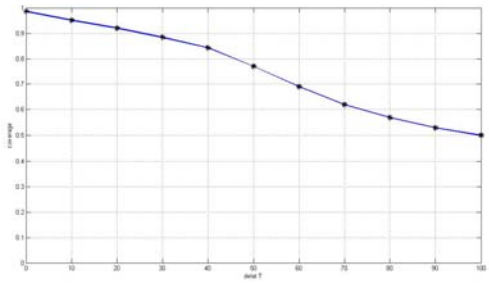
شکل ۵-۱۱ $P(\Delta t)$ برای مجموعه داده‌ی اول



شکل ۵-۱۲ $C(\Delta t)$ برای مجموعه داده‌ی اول



شکل ۵-۱۳ $P(\Delta t)$ برای مجموعه داده‌ی دوم



شکل ۵-۱۴ $C(\Delta t)$ برای مجموعه داده‌ی دوم

شکل‌های ۵-۱۱ تا ۵-۱۴ برطبق انتظار و توجیهات انجام شده نشان‌دهنده‌ی افزایش دقت و کاهش شمول با افزایش Δt است. در شکل ۵-۱۱ و ۵-۱۳، مشاهده می‌شود که زمانی‌که $\Delta t = 0$ است یعنی فقط داده‌ی ورودی در زمان t بدون در نظر گرفتن داده‌های ورودی گذشته برای به‌دست آوردن $P(\Delta t)$ استفاده می‌شود، دقت بسیار پایین است و البته این مسئله طبیعی است، زیرا جستجو برای یافتن آنتی‌بادی‌های مناسب در الگوریتم پیشنهادی که مبتنی بر AIS است طبیعت تکاملی دارد و بر

اساس جمعیتی از نمونه‌ها و تکثیر نمونه‌های مناسب عمل می‌کند بنابراین تکرر نمونه‌ها یکی از محصولات این نوع جستجو است که به جای یک نمونه‌ی کاندیدای جواب، با ایجاد گروهی از کاندیداها برای جواب که با یکدیگر همکاری می‌کنند باعث بهبود جستجو نیز می‌شود. به علاوه‌ی این تکرر داده، با وجود اینکه آنتی‌بادی‌ها سعی می‌کنند با داده‌های جدید وفق پیدا کنند، آنتی‌بادی‌هایی که برای مقابله با داده‌های گذشته به‌وجود آمده‌اند نیز در سیستم وجود دارند. بنابراین کم بودن مقدار دقت آنتی‌بادی‌ها زمانی که $\Delta t = 0$ است، توجیه می‌شود.

۵-۴ ارزیابی سود استفاده از وزن آیتم، وزن نشست و تئوری خطر

همان‌طور که در مقدمه‌ی این فصل آمد، برخی از خصوصیات الگوریتم را نمی‌توان با استفاده از روش بالا ارزیابی کرد، از جمله‌ی این خصوصیات، اضافه کردن تئوری خطر برای فیلتر کردن داده‌های نویزی، اضافه کردن وزن آیتم‌ها برای محاسبه‌ی شباهت بین دسترسی‌ها و اضافه کردن وزن اعتبار برای هر نشست هستند. برای ارزیابی این موارد، نوعی سیستم پیشنهاددهی ساده شبیه‌سازی کردیم، وب سایت ابزار آلات موسیقی را با URLهای ملاقات شده در داده‌های مورد استفاده برای استخراج مجموعه آیتم‌های مکرر در اختیار ۵۰ کاربر قرار داده و بعد از وارد شدن کاربر به سیستم و دسترسی به دو الی سه صفحه، بر اساس مجموعه آیتم‌های مکرر استخراج شده و صفحات ملاقات شده توسط کاربر، صفحات دیگری به کاربر پیشنهاد داده می‌شود، در انتهای هر صفحه از کاربر درخواست می‌شود تا میزان رضایت خود را از صفحه‌ی پیشنهادی بر اساس یک مقدار بین ۰ تا ۱۰۰ مشخص کند. صفحات پیشنهادی به کاربر با ساده‌ترین روش، یعنی با محاسبه‌ی نزدیک‌ترین مجموعه آیتم‌های مکرر استخراج شده به مجموعه صفحات ملاقات شده با کاربر انجام می‌شود. این سیستم با کمک ۵۰ کاربر و در سه حالت (۱) مجموعه آیتم‌های مکرر به‌دست آمده بدون اضافه کردن وزن آیتم و نشست و بدون اعمال تئوری خطر، (۲) مجموعه آیتم‌های مکرر به‌دست آمده با اعمال تئوری خطر و (۳) مجموعه آیتم‌های مکرر به‌دست آمده با اضافه کردن وزن آیتم و نشست و اعمال تئوری خطر

شبیه‌سازی شده است، نتایج میزان رضایت‌مندی کاربران برای هر یک از سه حالت در جدول ۵-۲ آمده است.

جدول ۵-۲ میزان رضایت‌مندی کاربران از صفحات پیشنهادشده به آن‌ها در حالات مختلف

	میزان رضایت‌مندی بیشینه	میزان رضایت‌مندی کمینه	میانگین نظر ۵۰ کاربر
حالت اول	٪۴۱	٪۱۵	٪۲۸
حالت دوم	٪۶۰	٪۴۰	٪۵۱
حالت سوم	٪۶۷	٪۴۵	٪۵۶

همان‌طور که از نتایج مشخص است اضافه شدن تئوری خطر، سطح رضایت‌مندی کاربران را از صفحات پیشنهاد داده شده به آن‌ها تا حد زیادی بهبود می‌دهد، که این مسئله به علت اضافه نشدن نشست‌هایی است که دارای اطلاعات صحیحی برای استخراج نیستند و تنها باعث پیچیدگی و منحرف شدن مجموعه آیتم‌های مکرر استخراج شده و همچنین بالا رفتن زمان اجرای الگوریتم می‌شوند. بر اساس نتایج مشخص است که اضافه شدن وزن آیتم و نشست در فرآیند محاسبه‌ی فاصله بین آنتی‌بادی و آنتی‌ژن نیز سطح رضایت‌مندی سیستم را کمی بهبود می‌دهد.

جدول ۵-۳ برخی از پروفایل‌های به‌دست آمده توسط الگوریتم پیشنهادی برای مجموعه داده‌ی ابزار آلات موسیقی

۱	/categories/do-it-yourself/schematics/digisound.Ring-Mod.schematic/ /samples.html/ /manufacturers/Moog/MiniMoog/info/ /search.cgi/ /links/machines.html/ /manufacturers/Moog/MiniMoog/moog.MiniMoog.specs/ /manufacturers/Moog/Modular/images/
۲	/categories/do-it-yourself/schematics/digisound.Ring-Mod.schematic/ /manufacturers/Casio/ /manufacturers/Casio/Overview/ /manufacturers/Casio/Overview/Casio/

۳	/manufacturers/ /Analogue-Heaven/ /manufacturers/Univox/ /manufacturers/Oxford/
٤	/categories/do-it-yourself/schematics/digisound.Ring-Mod.schematic/ /search.cgi/ /MMAgent/notfound.html/ /manufacturers/Cheetah/
٥	/categories/do-it-yourself/schematics/digisound.Ring-Mod.schematic/ /manufacturers/ /schematics.html/ /categories/do-it-yourself/schematics/ /manufacturers/Buchla/Modular/schematics/
٦	/manufacturers/Casio/ /manufacturers/Casio/Overview/ /manufacturers/Casio/Overview/Casio/ /manufacturers/Casio/CZ/
۷	/manufacturers/ /manufacturers/Yamaha/ /links/ /links/manufacturers.html/
۸	/manufacturers/ARP/ /manufacturers/ /manufacturers/Moog/ /manufacturers/Roland/ /manufacturers/Moog/Modular/info/ /manufacturers/Moog/Liberation/ /manufacturers/Moog/MicroMoog/ /manufacturers/Roland/TB-303/ /manufacturers/Moog/Prodigy/ /manufacturers/Moog/Rogue/ /manufacturers/Moog/Opus-3/ /manufacturers/ARP/Odyssey/
۹	/manufacturers/Roland/TR-909/samples/ /samples.html/ /categories/drum-machines/samples/ /categories/drum-machines/samples/deepsky_kicks/
۱۰	/categories/do-it-yourself/schematics/digisound.Ring-Mod.schematic/ /manufacturers/ARP/Odyssey/samples/ /search.cgi/

۱۱	/manufacturers/ /manufacturers/Roland/ /manufacturers/Roland/TR-909/ /manufacturers/Roland/TR-909/info/
۱۲	/samples.html/ /links/ /links/links.html/ /links/misc.html/
۱۳	/manufacturers/Mu-Tron/Bi-Phase/ /manufacturers/Mu-Tron/ /manufacturers/Mu-Tron/Bi-Phase/mods/
۱٤	/samples.html/ /manufacturers/Moog/MiniMoog/samples/ /manufacturers/ARP/Odyssey/samples/ /manufacturers/Akai/MPC/samples/
۱٥	/manufacturers/ /manufacturers/Roland/ /adaptive/2.html/ /categories/monosynths/System-100/ /manufacturers/RSF/ /manufacturers/RSF/info/ /manufacturers/RSF/info/RSF.Kobol/ /manufacturers/RSF/info/RSF.expander/
۱۶	/samples.html/ /categories/drum-machines/samples/ /categories/drum-machines/samples/deepsky_kicks/ /categories/drum-machines/samples/Rhythm-Ace.txt/
۱۷	/links/ /links/links.html/ /manufacturers/Peavey/ /manufacturers/Peavey/info/
۱۸	/manufacturers/ /search.cgi/ /guide/ /guide/finding.html/

۵-۵ مقایسه‌ی ویژگی‌های الگوریتم با الگوریتم‌های دیگر

در جدول ۵-۳ برخی از خصوصیات الگوریتم پیشنهادی به همراه الگوریتم‌های متفاوتی را که به نوعی در کلاس الگوریتم پیشنهادی قرار می‌گیرند، خلاصه شده است. در ستون اول ویژگی‌های الگوریتم پیشنهادی، سه ستون بعد الگوریتم‌های خوشه‌بندی SKM، DBSCAN و BIRCH که در فصل دو به آن‌ها اشاره شده و دو ستون آخر الگوریتم‌های SOSDM و aiNet که الگوریتم‌هایی هستند که بر اساس سیستم ایمنی مصنوعی طراحی شده‌اند و در فصل سه به آن‌ها اشاره شده است، آمده‌اند.

توجه کنید که همه‌ی تکنیک‌های مبتنی بر ایمنی مصنوعی (مانند اکثر تکنیک‌های خوشه‌بندی تکاملی)، نسبت به شرایط اولیه بی‌تفاوت هستند (قابل‌اطمینان) و این مسئله به این علت است که این تکنیک‌ها بر اساس جمعیتی از سلول‌ها کار می‌کنند. همچنین برخی از تکنیک‌ها برای ذخیره‌ی مجموعه داده در حافظه‌ی اصلی نیاز به بافر دارند که این مسئله در مورد مجموعه داده‌های بزرگ باعث ایجاد مشکل می‌شوند. به علاوه برخی از تکنیک‌ها قابلیت کارکرد در مقیاس‌های بالا را مرهون استفاده از ساختار اندیس‌گذاری خاصی هستند که به پیمایش اضافی داده دارد که این تکنیک‌ها هم در مورد مجموعه داده‌های بزرگ غیرقابل استفاده هستند.

در نهایت، برخلاف بیشتر الگوریتم‌های خوشه‌بندی که مبتنی بر محاسبه‌ی فاصله و افراز هستند (مانند k-means و الگوریتم‌های شبیه به آن) متدهایی که بر اساس چگالی طراحی شده‌اند، مانند الگوریتمی که در این پروژه ارائه شده است، مستقیماً در جستجوی نواحی چگال در فضای داده‌ها هستند و بنابراین می‌توانند در حین مقاوم بودن در برابر نویز، خوشه‌های مناسب‌تری را تشخیص دهند. این مسئله در مورد داده‌های با ابعاد بالا و خلوت مانند داده‌های دسترسی به وب، خود را بیشتر نشان می‌دهد و اهمیت بیشتری دارد.

جدول ۵-۴ مقایسه‌ی ویژگی‌های الگوریتم‌های متفاوت

روش	AIS-WUM	SKM	DBSCAN	BIRCH	aiNet	Fuzzy AIS	SOSDM
قابلیت اطمینان/عدم حساسیت به شرایط اولیه	بله	خیر	بله	خیر	بله	بله	بله
مقاومت در برابر نویز	بله	خیر	بله	خیر	خیر	بله	نسبتاً
نیاز به اسکن قبل از یادگیری	خیر	بله	بله	بله	بله	بله	خیر
پیچیدگی زمانی	O(N)	O(N)	O(Nlog(N))	O(N)	$O(N^2)$	$O(N^2)$	O(N)
نیاز به بافر برای داده‌ها	خیر	بله	بله	بله	بله	بله	بله
مشخص بودن تعداد خوشه‌ها	خیر	بله	خیر	خیر	خیر	خیر	خیر
مدیریت خوشه‌های متغیر	بله	خیر	خیر	خیر	بله	بله	بله
تخمین اتوماتیک شعاع هر خوشه	بله	خیر	خیر	خیر	خیر	بله	خیر
مدل خوشه‌بندی	Network	Centroids	Medoids	Centroids	Network	Network	Network
قابلیت استفاده از معیارهای شباهت متفاوت	بله	خیر	بله	خیر	بله	بله	بله
مبتنی بر افراز/مبتنی بر چگالی	چگالی	فاصله/ افراز	چگالی	افراز	فاصله/افراز	چگالی	فاصله/افراز

۵-۶ جمع‌بندی

در این فصل به ارائه‌ی چهارچوب و معیارهایی برای تخمین کیفیت الگوریتم پرداختیم و کیفیت الگوریتم را از جهات مختلف مورد بررسی قرار دادیم. نتایج، نشان دهنده‌ی کارایی بالای الگوریتم در یادگیری و استخراج اطلاعات است. دلیل اصلی توانایی الگوریتم پیشنهادی در یادگیری با یک بار عبور از داده‌ها، توانمندی ساختار شبکه‌ی ایمنی است.

۶

نتیجه‌گیری و پیشنهادات

مقدمه

ارزیابی الگوریتم پیشنهادی

پیشنهادات برای کارهای آتی

جمع‌بندی

"In my end is my beginning."
-- T.S. Eliot

فصل ۶: نتیجه‌گیری و پیشنهادات

۶-۱ مقدمه

در دو فصل قبلی، الگوریتمی به نام AISWUM برای کاوش در داده‌های WUM عرضه و نتایج اجرای آن بر روی داده‌های واقعی مورد بررسی و ارزیابی قرار گرفت. در این فصل به جمع‌بندی کار انجام شده پرداخته شده و با توجه به کارایی الگوریتم نتایجی کلی از اجرای این پروژه ارائه خواهد شد. همچنین بر اساس آزمایشاتی که بر روی الگوریتم انجام گرفته و تجربیاتی که حین توسعه‌ی الگوریتم ایجاد شده است به ارائه‌ی پیشنهاداتی برای توسعه الگوریتم می‌پردازیم.

۶-۲ ارزیابی الگوریتم پیشنهادی

در این پروژه، یک الگوریتم مقاوم و کارا در مقیاس‌های بزرگ به نام AISWUM برای استخراج مجموعه آیتم‌های مکرر از داده‌های دسترسی به وب ارائه شده است. کاربردی که AISWUM، برای آن طراحی شده است بسیار چالش برانگیز است. زیرا سایز وب (منبع داده) بسیار بزرگ است، نویز در اطلاعات وب بسیار زیاد است و مفاهیم در حال تغییراند. AISWUM سیستمی بدیع الهام گرفته از سیستم ایمنی طبیعی است که بر اساس اصول این سیستم طبیعی عمل می‌کند. بخشی از بداعت این سیستم عملکرد خوب آن در شرایط پویا است. از ویژگی‌های منحصر بفرد دیگر این الگوریتم ارائه شده می‌توان به استفاده از مفاهیم و فرآیندهای جدیدی مانند تئوری خطر و جهش هدایت شده است که در عملکرد مناسب الگوریتم تاثیر به سزایی دارند، اشاره کرد. ارائه‌ی معیارهایی به عنوان وزن آیتم و وزن آنتی‌ژن نیز مفاهیم جدیدی هستند که در سیستم ایمنی مصنوعی وارد شده‌اند و باعث می‌شوند علاوه بر استخراج مجموعه آیتم‌های مکرر، مجموعه آیتم‌هایی که دارای وزن زیادی هستند و تعداد تکرار زیادی ندارند، استخراج شوند که تعبیر آن در سیستم ایمنی مصنوعی حفظ و بقای ژن‌های قوی در طبیعت است. نکته‌ی مهم دیگر در طراحی الگوریتم، استخراج اطلاعات با یک بار عبور از داده‌های

ورودی است دقیقاً مانند سیستم ایمنی طبیعی که در یک دوره‌ی زیستی موجودات طبیعی، برای مبارزه با آنتی‌ژن‌ها وفق پیدا می‌کند. معنی یک بار عبور از داده‌ها در الگوریتم ارائه شده اینست که در این الگوریتم، حلقه‌ی تکرار روی داده‌های ورودی وجود ندارد و با یک‌بار برخورد با یک داده‌ی ورودی الگوریتم می‌تواند اطلاعات درون داده را خلاصه و در خود حفظ کند. بنابراین الگوریتم زمان اجرای مناسبی برای کاربردهای بلادرنگ دارد که اکثر قریب به اتفاق کاربردهای موجود که از اطلاعات خروجی الگوریتم ارائه شده استفاده می‌کنند، کاربردهای بلادرنگ هستند. همچنین ایجاد فایل لاگی که در آن زمان تولد، مرگ و دلیل تولد و مرگ آنتی‌بادی‌ها در آن ذخیره می‌شود، برای تعیین زمان ظهور و افول مسیرهای خاصی که به دلیل اتفاقات زمانی‌که در دنیای واقعی می‌افتد، ایجاد می‌شوند، بسیار موثر است و می‌تواند یک داده‌ی مناسب برای مونیتور کردن جریان دسترسی به وب برای مسئول وب باشد.

همان‌طور که گفته شد، هدف از این الگوریتم ارائه شده، یافتن مجموعه آیتم‌های مکرر در داده‌های دسترسی به وب است. مهمترین الگوریتم موجود برای این منظور اپریوری است که در فصل دو مفصلاً توضیح داده شده است، این الگوریتم از نظر زمانی و مکانی برای داده‌های حجیم غیرقابل استفاده است.

از آن جا که ارزیابی و مقایسه‌ی متدهای مختلف در چهارچوب داده‌های پویا و حجیم که امکان بیش از یک‌بار پیمایش داده‌ها وجود ندارد بسیار مشکل و شاید غیرممکن است. معیارهایی برای نمایش توانایی یادگیری سیستم ارائه کردیم و همچنین ویژگی‌های الگوریتم ارائه شده را تحت جدولی با الگوریتم‌های مشابه که از جهات گوناگون در کلاس الگوریتم پیشنهادی قرار می‌گیرند مقایسه کردیم الگوریتم پیشنهادی مجموعه‌آیتم‌های مکرر کمتر و قوی‌تری نسبت به اپریوری ایجاد می‌کند که تفسیر آن‌ها نیز معقول‌تر از مجموعه‌آیتم‌های به‌دست آمده از الگوریتم اپریوری است، ضمن اینکه این الگوریتم با یک بار عبور از داده می‌تواند چنین مجموعه آیتم‌های مکرری تولید کند که این خصوصیت

الگوریتم آن را برای استفاده در کاربردهای بلادرنگ مناسب می‌سازد. همچنین این الگوریتم به شرایط اولیه حساس نیست و مقادیر پارامترهای مختلف تاثیری در نتایج نهایی الگوریتم به‌وجود نمی‌آورد.

۶-۳ پیشنهادات برای کارهای آتی

همان‌طور که گفته شد یکی از مهمترین مزیت‌های الگوریتم AISWUM اینست که سیستم تنها با یک‌بار برخورد با داده‌های ورودی می‌تواند با داده‌ها وفق پیدا کرده و اطلاعات مفید آن‌ها را در خود حفظ کند. بنابراین به نظر می‌رسد الگوریتم به آسانی با کمی تغییر قابل اعمال بر روی داده‌های جریانی است و یکی از مسیرهایی که در امتداد این پروژه می‌توان به آن پرداخت، تست الگوریتم بر روی داده‌های جریانی مختلف و ارزیابی الگوریتم در این دامنه و در صورت لزوم ایجاد تغییرات بر روی الگوریتم برای دریافت نتایج بهتر در دامنه‌ی داده‌های جریانی است.

یک پیشنهاد دیگر برای بهبود الگوریتم، استفاده از آنتولوژی برای تعیین میزان شباهت بین دو صفحه در یک نشست است، که از این میزان شباهت در الگوریتم پیشنهادی برای تعیین معیار هم‌نواختی یک نشست استفاده شد. استفاده از آنتولوژی‌ها می‌تواند انواع ارتباطات بین دو صفحه‌ی وب را کشف کند که این ارتباط می‌تواند ارتباط بالامجموعه‌ای، زیرمجموعه‌ای، مترادف و ... باشد. برای مثال با استفاده از WordNet می‌توان ارتباطات عجیب و جالب ، مانند متضاد بودن را هم استخراج کرد. و به این وسیله شباهت‌های موجود بین دو صفحه را به‌دست آورد.

پیشنهاد دوم، تخصصی کردن این الگوریتم برای کاربردهای خاصی است که در فصل دو به آن‌ها اشاره شد از قبیل شخصی‌سازی، وب‌سایت‌های وفقی و ... . با تخصصی کردن الگوریتم برای کاربردهای خاص، می‌توان ویژگی‌های دیگری به الگوریتم افزود، برای مثال برای استفاده از خروجی این الگوریتم برای سیستم‌های پیشنهاددهی اتوماتیک، می‌توان الگوریتم را با روش‌های استخراج کلمات کلیدی از اسناد مربوط به یک URL ادغام کرد و به این‌صورت در یک الگوریتم، مکانیزمی برای ارائه‌ی پیشنهاد به کاربران آتی بر اساس دسترسی کاربران قبلی که دارای سلایق مشابه به کاربران آتی هستند ارائه داد.

از آنجا که مقایسه‌ی چنین سیستمی، که شرایط پویا دارد و روی مجموعه داده‌های بزرگ پیاده می‌شود با متدهای دیگر کار بسیار سخت و تقریباً ناممکنی است، پیشنهاد سوم و آخر به عنوان مسیر کار در آینده ارائه‌ی روش‌های ارزیابی مناسب‌تر برای نتایج الگوریتم است که خاصیت پویایی و تغییر نتایج در آن ملموس‌تر باشد.

۶-۴ جمع‌بندی

طی پنج فصل گذشته، به ارائه‌ی مفاهیم موجود در دو زمینه‌ی WUM و AIS که دو مبحث کلیدی در این رساله بوده‌اند، الگوریتم پیشنهادی با جزئیات و آزمایشات مختلفی که برای ارزیابی الگوریتم انجام شده است پرداخته شد. نهایتاً در فصل حاضر نتیجه‌گیری از کل رساله ارائه شد و نقاط قوت و ضعف الگوریتم تشریح شد، جمع‌بندی کلی اثبات کارایی بالای الگوریتم و برتری نتایج حاصل از آن نسبت به کارهای موجود است.

مراجع

مراجع

[1] P.-N. Tang, T. M. Steinbach, V. Kumar, ***"Introduction to Data Mining",*** *Addison Wesley*, (2005).

[2] A. Secker, Dissertation Title: ***"Artificial Immune Systems for Web Content Mining: Focusing on the Discovery of Interesting Information"***, *University of Kent in Canterbury*, Uk. (2006).

[3] L. N. de Castro, J. Timmis, ***"Artificial Immune Systems: A New Computational Intelligence Approach"***, *Springer-Verlag*, (2002).

[4] D. Dasgupta, ***"Artificial Immune Systems and Their Applications"***, *Springer-Verlag*, (1999).

[5] R. Baeza-Yates, B. Ribeiro-Neto, ***"Modern Information Retrieval"***, *Harlow: Addison Wesley Longman*, (1999).

[6] D. Hand, H. Mannila, P. Smyth, ***"Principles of Data Mining"***, *MIT Press*, (2001).

[7] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy, ***"Advances in Knowledge Discovery and Data Mining"***, *MIT Press*, (1996).

[8] A. A. Freitas, ***"Data Mining and Knowledge Discovery with Evolutionary Algorithms"***, *Springer-Verlag*, (2002).

[9] S. Chakrabarti, ***"Mining the web (Discovering Knowledge from Hypertext Data)"***, *San Francisco: Morgan Kaufmann*, (2003).

[10] S. Brin, L. Page, ***"The anatomy of a large-scale hypertextual Web search engine"***, *Computer Networks and ISDN Systems* 30 (1–7), pp. 107–117, (1998).

[11] Configuration file of W3C httpd, http://www.w3.org/Daemon/User/Config/ (1995).

[12] W3C Extended Log File Format, http://www.w3.org/TR/WD-logfile.html (1996).

[13] J.R. Punin, M.S. Krishnamoorthy, M.J. Zaki, ***"Logml: Log markup language for web usage mining"***, in: *R. Kohavi, B. Masand, M. Spiliopoulou, J. Srivastava (Eds.), WEBKDD 2001—Mining Web Log Data Across All Customers Touch Points, Third International Workshop*, San

Francisco, CA, USA, August 26, 2001. Revised Papers, vol. 2356 of Lecture Notes in Computer Science, Springer, pp. 88–112, (2002).

[14] A. Nanopoulos, M. Zakrzewicz, T. Morzy, Y. Manolopoulos, *"**Indexing web access-logs for pattern queries**"*, in: *fourth ACM CIKM International Workshop on Web Information and Data Management (WIDM_02)*, (2002).

[15] K.P. Joshi, A. Joshi, Y. Yesha, *"**On using a warehouse to analyze web logs**"*, *Distributed and Parallel Databases*, 13(2), pp. 161–180, (2003).

[16] D.M. Kristol, *"**Http cookies: standards, privacy, and politics**"*, *ACM Transactions on Internet Technology* (TOIT)1(2), pp. 151–198, (2001).

[17] B. Berendt, B. Mobasher, M. Nakagawa, M. Spiliopoulou, *"**The impact of site structure and user environment on session reconstruction in web usage analysis**"*, in: *Proceedings of the 4th WebKDD 2002 Workshop, at the ACMSIGKDD Conference on Knowledge Discovery in Databases (KDD_2002)*, (2002).

[18] K.D. Fenstermacher, M. Ginsburg, *"**Mining client-side activity for personalization**"*, in: *Fourth IEEE International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems (WECWIS_02)*, pp. 205–212, (2002).

[19] Pilot Software, Web site analysis, Going Beyond Traffic Analysis http://www.marketwave.com/productssolutions/ hitlist.html (2002).

[20] S. Ansari, R. Kohavi, L. Mason, Z. Zheng, *"**Integrating e-commerce and data mining: Architecture and challenges**"*, in: *N. Cercone, T.Y. Lin, X. Wu (Eds.), Proceedings of the 2001 IEEE International Conference on Data Mining (ICDM 2001)*, IEEE Computer Society, (2001).

[21] C. Shahabi, F. Banaei-Kashani, *"**A framework for efficient and anonymous web usage mining based on client-side tracking**"*, in: *R. Kohavi, B. Masand, M. Spiliopoulou, J. Srivastava (Eds.), WEBKDD 2001—Mining Web Log Data Across All Customers Touch Points, Third International Workshop*, San Francisco, CA, USA, August 26, 2001. Revised papers, vol. 2356 of Lecture Notes in Computer Science, Springer, 2002, pp. 113–144, (2005)

[22] L.D. Catledge, J.E. Pitkow, *"**Characterizing browsing strategies in the World-Wide Web**"*, *Computer Networks and ISDN Systems* 27 (6), pp. 1065–1073, (1995).

[23] C.R. Anderson, ***"A machine learning approach to web personalization"***, Ph.D. thesis, *University of Washington*, (2002).

[24] R. Cooley, B. Mobasher, J. Srivastava, ***"Data preparation for mining world wide web browsing patterns"***, *Knowledge and Information Systems* 1 (1), pp. 5–32, (1995).

[25] B. Diebold, M. Kaufmann, ***"Usage-based visualization of web localities"***, in: *Australian symposium on information visualization*, pp. 159–164, (2001).

[26] P.-N. Tan, V. Kumar, ***"Modeling of web robot navigational patterns"***, in: *WEBKDD 2000 —Web Mining for Ecommerce— Challenges and Opportunities, Second International Workshop*, (2000).

[27] P.-N. Tan, V. Kumar, ***"Discovery of web robot sessions based on their navigational patterns"***, *Data Mining and Knowledge Discovery* 6 (1), pp. 9–35, (2002).

[28] J. Srivastava, R. Cooley, M. Deshpande, P.-N. Tan, ***"Web usage mining: discovery and applications of usage patterns from web data"***, *SIGKDD Explorations* 1 (2), pp. 12–23, (2000).

[29] R. Cooley, ***"Web usage mining: discovery and application of interesting patterns from web data"***, Ph.D. thesis, *University of Minnesota*, (2000).

[30] B. Mobasher, R. Cooley, J. Srivastava, ***"Automatic personalization based on web usage mining"***, *Communications of the ACM* 43 (8), pp. 142–151, (2000).

[31] IBM, SurfAid Analytics http://surfaid.dfw.ibm.com (2003).

[32] M. Chen, A.S. LaPaugh, J.P. Singh, ***"Predicting category accesses for a user in a structured information space"***, in: *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 65–72, (2002).

[33] G. Stumme, A. Hotho, B. Berendt, ***"Usage mining for and on the semantic web"***, in: *National Science Foundation Workshop on Next Generation Data Mining*, (2002).

[34] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, S. Slattery, ***"Learning to construct knowledge bases from the world wide web"***, *Artificial Intelligence* 118 (1–2), pp. 69–113, (2000).

[35] A. Banerjee, J. Ghosh, ***"Clickstream clustering using weighted longest common subsequences"***, in: *Proceedings of the Web Mining Workshop at the 1st SIAM Conference on Data Mining*, (2001).

[36] E.H. Chi, P. Pirolli, K. Chen, J.E. Pitkow, ***"Using information scent to model user information needs and actions and the web"***, in: *Proceedings of ACM CHI 2002 Conference on Human Factors in Computing Systems*, ACM Press, pp. 490–497, (2001).

[37] R. Cooley, ***"The use of web structure and content to identify subjectively interesting web usage patterns"***, *ACM Transactions on Internet Technology* (TOIT) 3 (2), pp. 93–116, (2003).

[38] J. Andersen, A. Giversen, A.H. Jensen, R.S. Larsen, T.B. Pedersen, J. Skyt, ***"Analyzing clickstreams using subsessions"***, in: *International Workshop on Data Warehousing and OLAP (DOLAP 2000)*, (2000).

[39] J. Pei, J. Han, B. Mortazavi-asl, H. Zhu, ***"Mining access patterns efficiently from web logs"***, in: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 396–407, (2000).

[40] E. Menasalvas, S. Millan, J. Pena, M. Hadjimichael, O. Marban, ***"Subsessions: a granular approach to click path analysis"***, in: *Proceedings of FUZZ-IEEE Fuzzy Sets and Systems Conference, at the World Congress on Computational Intelligence*, Honolulu, HI, pp. 12–17, (2002).

[41] J.Z. Huang, M. Ng, W.-K. Ching, J. Ng, D. Cheung, ***"A cube model and cluster analysis for web access sessions"***, in: *R. Kohavi, B. Masand, M. Spiliopoulou, J. Srivastava (Eds.), WEBKDD 2001—Mining Web Log Data Across All Customers Touch Points, Third International Workshop*, San Francisco, CA, USA, August 26, 2001. Revised papers, vol. 2356 of Lecture Notes in Computer Science, Springer, pp. 48–67, (2002).

[42] J. Han, M. Kamber, ***"Data Mining Concepts and Techniques"***, *Morgan Kaufmann*, (2001).

[43] A. Nanopoulos, D. Katsaros, Y. Manolopoulos, ***"Exploiting web log mining for web cache enhancement"***, in: *R. Kohavi, B. Masand, M. Spiliopoulou, J. Srivastava (Eds.), WEBKDD 2001—Mining Web Log Data Across All Customers Touch Points, Third International Workshop*, San Francisco, CA, USA, August 26, 2001. Revised papers, vol. 2356 of Lecture Notes in Computer Science, Springer, pp. 68–87, (2002).

[44] X. Huang, N. Cercone, A. An, ***"Comparison of interestingness functions for learning web usage patterns"***, in: *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, ACM Press, pp. 617–620, (2002).

[45] S.S.C. Wong, S. Pal, ***"Mining fuzzy association rules for web access case adaptation"***, in: *Workshop on Soft Computing in Case-Based Reasoning*, *International Conference on Case-Based Reasoning (ICCBR_Ol)*, 2001. 238 F.M. Facca, P.L. Lanzi / Data & Knowledge Engineering 53, pp. 225–241, (2005).

[46] E.S. Nan-Niu, M. El-Ramly, ***"Understanding web usage for dynamic web-site adaptation: A case study"***, in: *Proceedings of the Fourth International Workshop on Web Site Evolution (WSE_02)*, IEEE, pp. 53-64, (2002).

[47] B. Mortazavi-Asl, ***"Discovering and mining user web-page traversal patterns"***, Master thesis, *Simon Fraser University*, (2001).

[48] J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal, M. Hsu, ***"FreeSpan: frequent pattern-projected sequential pattern mining"***, in: *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD_2000)*, Boston, MA, (2000).

[49] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, M.-C. Hsu, ***"Mining sequential patterns by pattern growth: the PrefixSpan Approach"***, *IEEE Transactions on Knowledge and Data Engineering*, in press.

[50] S.E. Jespersen, J. Thorhauge, T.B. Pedersen, ***"A hybrid approach to web usage mining"***, in: *Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery*, Springer-Verlag, pp. 73–82, (2002).

[51] J. Borges, ***"A data mining model to capture User Web navigation patterns"***, Ph.D. thesis, *Department of Computer Science, University College London*, (2000).

[52] J. Heer, E.H. Chi, ***"Mining the structure of user activity using cluster stability"***, in: *Proceedings of the Workshop on Web Analytics, Second SIAM Conference on Data Mining*, ACM Press, (2002).

[53] Y. Xie, V.V. Phoha, ***"Web user clustering from access log using belief function"***, in: *Proceedings of the First International Conference on Knowledge Capture (K-CAP 2001)*, ACM Press, pp. 202–208, (2001).

[54] B. Hay, G. Wets, K. Vanhoof, ***"Clustering navigation patterns on a website using a sequence alignment method"***, In: *Intelligent Techniques for Web Personalization: IJCAI 2001, 17th Int. Joint Conf. on Artificial Intelligence, Seattle*, WA, USA, pp. 1–6, (2001).

[55] C. Shahabi, Y.-S. Chen, ***"Improving user profiles for e-commerce by genetic algorithms"***, *E-Commerce and Intelligent Methods Studies in Fuzziness and Soft Computing* 105 (8), (2002).

[56] O. Nasraoui, F. Gonzalez, D. Dasgupta, ***"The fuzzy artificial immune system: Motivations, basic concepts, and application to clustering and web profiling"***, in: *Proceedings of the World Congress on Computational Intelligence (WCCI) and IEEE International Conference on Fuzzy Systems*, pp. 711–716, (2002).

[57] S. Oyanagi, K. Kubota, A. Nakase, ***"Application of matrix clustering to web log analysis and access prediction"***, in: *WEBKDD2001—Mining Web Log Data Across All Customers Touch Points, Third International Workshop*, (2001).

[58] B. Mobasher, H. Dai, M. Tao, ***"Discovery and evaluation of aggregate usage profiles for web personalization***", Data Mining and Knowledge Discovery 6, pp. 61–82, (2006).

[59] T. Zhang, R. Ramakrishnan, M. Livny, ***"Birch: An Efficient Data Clustering Method for Very Large Databases",*** ACM SIGMOD International Conference on Management of Data, pp. 103-114, (1996).

[60] M. Ester, H. P. Kriegel, J. Sander, X. Xu, ***"A density-based algorithm for discovering clusters in large spatial databases with noise",*** In Proceeding of the 2nd international conf. on Knowledge Discovery and Data Mining (KDD96), (1996).

[61] P. Bradley, U. Fayyad, and C. Reina, ***"Scaling clustering algorithms to large databases"***, In Proceedings of the 4th international conf. on Knowledge Discovery and Data Mining (KDD98), 1998.

[62] H.R. Kim, P.K. Chan, ***"Learning implicit user interest hierarchy for context in personalization"***, in: *Proceedings of the 2003 International Conference on Intelligent User Interfaces*, ACM Press, pp. 101-108, (2003).

[63] W. Lin, S.A. Alvarez, C. Ruiz, ***"Efficient adaptive–support association rule mining for recommender systems"***, *Data Mining and Knowledge Discovery 6* (1), pp. 83–105, (2002).

[64] T. Li, ***"Web-document prediction and presenting using association rule sequential classifiers"***, Masters thesis, *Simon Fraser University*, (2001).

[65] Y. Fu, M. Creado, C. Ju, ***"Reorganizing web sites based on user access patterns"***, in: *Proceedings of the Tenth International Conference on Information and Knowledge Management*, ACM Press, pp. 583–585, (2001).

[66] C. Bounsaythip, E. Rinta-Runsala, ***"Overview of data mining for customer behavior modeling"***, Technical Report TTE1-2001-18, *VTT Information Technology*, (2001).

[67] H.K. Dai, B. Mobasher, ***"Using ontologies to discover domain-level web usage profiles"***, in: *Proceedings of the 2nd Semantic Web Mining Workshop at ECML/PKDD 2002*, Helsinki, Finland, (2002).

[68] T. Kamdar, ***"Creating adaptive web servers using incremental web log mining"***, Master thesis, *Computer Science Department, University of Maryland*, Baltimore County, (2001).

[69] M. Eirinaki, M. Vazirgiannis, I. Varlamis, ***"Sewep: using site semantics and a taxonomy to enhance the web Personalization process"***, in: *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM Press, pp. 99–108, (2003).

[70] F. Bonchi, F. Giannotti, C. Gozzi, G. Manco, M. Nanni, D. Pedreschi, C. Renso, S. Ruggieri, ***"Web log data warehousing and mining for intelligent web caching"***, *Data Knowledge Engineering* 39 (2), pp. 165–189, (2001).

[71] B. Lan, S. Bressan, B.C. Ooi, K.-L. Tan, ***"Rule-assisted prefetching in web-server caching"***, in: *Proceedings of the Ninth International Conference on Information and Knowledge Management (CIKM 2000)*, ACM Press, pp. 504–511, (2000).

[72] A. Nanopoulos, D. Katsaros, Y. Manolopoulos, ***"A data mining algorithm for generalized web prefetching"***, *IEEE Transactions on Knowledge and Data Engineering* 15 (5), pp. 1155–1169, (2003).

[73] Y.-H. Wu, A.L.P. Chen, *"**Prediction of web page accesses by proxy server log**"*, *World Wide Web* 5 (1), pp. 67–88, (2002).

[74] Q. Yang, H.H. Zhang, *"**Web-log mining for predictive web caching**"*, *IEEE Transactions on Knowledge and Data Engineering* 15 (4), pp. 1050–1054, (2003).

[75] C.R. Anderson, P. Domingos, D.S. Weld, *"**Relational markov models and their application to adaptive web navigation**"*, in: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2002)*, (2002).

[76] B. Berendt, *"**Using site semantics to analyze, visualize, and support navigation**"*, *Data Mining and Knowledge Discovery* 6 (1), pp. 37–59, (2002).

[77] M. Spiliopoulou, C. Pohle, *"**Data mining for measuring and improving the success of web sites**"* (1–2), pp. 85–114, (2001).

[78] R. Srikant, Y. Yang, *"**Mining web logs to improve website organization**"*, *World Wide Web*, pp. 430 437, (2001).

[79] J. Zhu, J. Hong, J.G. Hughes, *"**Using markov chains for link prediction in adaptive web sites**"*, in: *D.W. Bustard, W. Liu, R. Sterritt (Eds.), Soft-Ware 2002: Computing in an Imperfect World, First International Conference, SoftWare 2002*, Belfast, Northern Ireland, 8–10 April 2002, Proceedings, vol. 2311 of Lecture Notes in Computer Science, Springer, pp. 60–73, (2002).

[80] W.-L. Chang, S.-T. Yuan, *"**A synthesized learning approach for web-based crm**"*, in: *WEBKDD 2000—Web Mining for E-Commerce—Challenges and Opportunities*, Second International Workshop, (2000).

[81] M. Zeidenberg, *"**Neural Network Models in Artificial Intelligence**"*: Ellis Horwood, 1990.

[82] M. Dorigo, T. Stützle, *"**Ant Colony Optimization**"*: *MIT Press*, (2004).

[83] J. Holland, *"**Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence**"*: *MIT Press*, (1992).

[84] M. Mitchell, *"**An Introduction to Genetic Algorithms**"*: *MIT Press*, (1996).

[85] K. de Jong, *"**Genetic Algorithms: A 30 Year Perspective**"*, *Festschrift Conference in Honour of John H. Holland*, University of Michigan, (1999).

[86] R. N.Germain, *"**An Innately Interesting Decade of Research in Immunology**"*, *Nature Medicine*, 10(12), pp. 1307-1320, (2004).

[87] P. Matzinger, *"**The Danger Model: A Renewed Sense of Self**"*, *Science*, 296, pp. 301-305, (2002).

[88] U. Aickelin, S. Cayzer, *"**The Danger Theory and Its Application to Artificial Immune Systems**"*, *1st International Conference on Artificial Immune Systems (ICARIS 2002)*, Canterbury, UK. pp. 141-148, (2002).

[89] U. Aickelin, P. Bentley, S. Cayzer, J. Kim, J. McLeod, *"**Danger Theory: The Link between AIS and IDS**"*, *2nd International Conference on Artificial Immune Systems (ICARIS 2003)*, Edinburgh, UK. Lecture Notes in Computer Science 2787. Springer-Verlag, pp. 147-155, (2003).

[90] A. Secker, A. A. Freitas, J. Timmis, *"**A Danger Theory Inspired Approach to Web Mining**"*, *2nd International Conference on Artificial Immune Systems (ICARIS 2003)*, Edinburgh, UK. Lecture Notes in Computer Science 2787. Springer-Verlag, pp. 156-167, (2003).

[91] J. Greensmith, U. Aickelin, S. Cayzer, *"**Introducing Dendritic Cells as a novel Immune-Inspired Algorithm for Anomaly Detection**"*, *4th International Conference on Artificial Immune Systems (ICARIS 2005)*, Banff, Canada. Lecture Notes in Computer Science 3627. Springer-Verlag, pp. 153-167, (2005).

[92] J. Twycross, U. Aickelin, *"**Towards a Conceptual Framework for Innate Immunity**"*, *4th International Conference on Artificial Immune Systems (ICARIS 2005)*, Banff, Canada. Lecture Notes In Computer Science 3627. Springer-Verlag, pp. 112-125, (2005).

[93] F. M. Burnett, *"**The Clonal Selection Theory of Acquired Immunity**"*, *Cambridge University Press*, (1959).

[94] L. N. de Castro, F. J. Von Zuben, *"**The Clonal Selection Algorithm with Engineering Applications**"*, *GECCO 2000*, *Workshop on Artificial Immune Systems and Their Applications*, Las Vegas, USA. pp. 36-39, (2000).

[95] N. K. Jerne, "Towards a Network Theory of the Immune System", *Annals of Immunology*, 125(C), pp. 373-389, (1974).

[96] J. Timmis, *"**Artificial Immune Systems: A Novel Data Analysis Technique Inspired by The Immune Network Theory**"*, Ph.D. Thesis, *University of Wales*, Aberystwyth, Wales, (2000).

[97] P. Matzinger, *"**The Danger Model: A Renewed Sense of Self**"*, *Science*, 296, pp. 301-305, (2002).

[98] P. Matzinger, *"**The Real Function of The Immune System or Tolerance and The Four D's**"*. Retrieved 30/10/2002, from http://cmmg.biosci.wayne.edu/asg/polly.html, (2002).

[99] C. C. Anderson, P. Matzinger, *"**Danger: The view from the bottom of the cliff**"*, *Seminars in Immunology*, 12(3), pp. 231-238, (2000).

[100] S. Gallucci, P. Matzinger, *"**Danger signals: SOS to the immune system**"*, *Current Opinion in Immunology*, 13(1), pp. 114-119, (2001).

[101] P. Bretscher, M. Cohn, *"**A theory of self-nonself discrimination**"*, *Science* 169, pp. 1042-1049, (1970).

[102] P. Matzinger, *"**Tolerance, Danger and the Extended Family**"*, *Annual Review of Immunology*, 12, pp. 991-1045, (1994).

[103] J. D. Farmer, N. H. Packard, A.S. Perelson, *"**The Immune System, Adaptation and Machine Learning**"*, *Physica*, 22(D), pp. 187-204, (1986).

[104] L. N. de Castro, J. Timmis, *"**Artificial Immune Systems as a Novel Soft Computing Paradigm**"*, *Soft Computing*, 7(8), pp. 526-544, (2003).

[105] A. S. Perelson, G. F. Oster, *"**Theoretical Studies of Clonal Selection: Minimal Antibody Repertoire Size and Reliability of Self Non-Self Discrimination**"*, *Journal of Theoretical Biology*, 81(4), pp. 645-670, (1979).

[106] L. de Castro, F. Von Zuben, *"**Learning and Optimization Using the Clonal Selection Principle**"*, *IEEE Transactions on Evolutionary Computation*, Special Issue on Artificial Immune Systems, 6(3), pp. 239-251, (2001).

[107] V. Cutello, G. Narzisi, G. Nicosia, M. Pavone, *"**Clonal Selection Algorithms: A Comparative Case Study Using Effective Mutation Potentials**"*, *ICARIS 2005*, Banff, Canada. Lecture Notes In Computer Science 3627. Springer-Verlag, pp. 13-28, (2005).

[108] J. A. White, S. M. Garret, *"**Improved Pattern Recognition with Artificial Clonal Selection**"*, *2nd International Conference on Artificial*

Immune Systems (ICARIS 2003), Edinburgh, UK. Lecture Notes in Computer Science 2787. Springer-Verlag, pp. 181-193, (2003).

[109] U. Aickelin, S. Cayzer, ***"The Danger Theory and Its Application to Artificial Immune Systems"***. *In the proceedings of 1st International Conference on Artificial Immune Systems (ICARIS)*, University of Kent at Canterbury, UK, September 9-11, (2002).

[110] J. Greensmith, U. Aickelin, S. Cayzer, ***"Introducing Dendritic Cells as a Novel Immune-Inspired Algorithm for Anomaly Detection"***. *In the Proceedings of ICARIS-2005, 4th International Conference on Artificial Immune Systems*, Banff, Canada. (2005).

[111] L. N. de Castro, F. J. Von Zuben, ***"aiNet: An Artificial Immune Network for Data Analysis"***, In Abbass, H., Sarker, R. & Newton, C. (Eds.), *Data Mining: A Heuristic Approach,* pp. 231-259, (2002).

[112] D. Dasgupta, Z. Ji, F. Gonzalez, ***"Artificial Immune Systems (AIS) Research in the Last Five Years"***, *CEC 2003*, Canberra, Australia. IEEE, pp. 123-130, (2003).

[113] J. E. Hunt, D. E. Cooke, ***"Learning Using an Artificial Immune System"***, *Journal of Network and Computer Applications*, 19(2), pp. 189-212, (1996).

[114] M. Neal, J. Hunt, J. Timmis, ***"Augmenting an Artificial Immune Network"***, *International Conference on Systems and Man and Cybernetics*, San Diego, USA. pp. 3821-3826, (1998).

[115] J. Timmis, ***"Artificial Immune Systems: A Novel Data Analysis Technique Inspired by The Immune Network Theory"***, Ph.D. Thesis, *University of Wales*, Aberystwyth, Wales, (2000).

[116] C. L. Blake, C. J. Merz, ***"UCI Repository of Machine Learning Databases"***, Retrieved May 2003 from http://www.ics.uci.edu/~mlearn/MLRepository.html, (1998).

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

[118] T. Knight, J. Timmis, ***"AINE: An Immunological Approach to Data Mining"***, *IEEE International Conference on Data Mining*, San Jose, USA. IEEE Press, pp. 297-304, (2001).

[119] O. Nasraoui, D. Dasgupta, F. Gonzalez. *"**An Novel Artificial Immune System Approach to Robust Data Mining**". In the proceedings of the International Conference Genetic and Evolutionary Computation (GECCO)*, New York, July 9-13, (2002).

[120] N. Cruz-Cortés, D. Trejo-Pérez, C. A. Coello, *"**Handling Constraints in Global Optimisation Using an Artificial Immune System**", 4th International Conference on Artificial Immune Systems (ICARIS 2005)*, Banff, Canada. Lecture Notes in Computer Science 3627. Springer-Verlag, pp. 234-247, (2005).

[121] J. Kelsy, J. Timmis, A. Hone, *"**Chasing Chaos**"*, *CEC2003*, Canberra, Australia. IEEE Press, pp. 413-419, (2003).

[122] J. Timmis, C. Edmonds, J. Kelsey, *"**Assessing the Performance of Two Immune Inspired Algorithms and a Hybrid Genetic Algorithm for Function Optimisation**"*, *Congress on Evolutionary Computation (CEC 2004)*, Portland, USA. IEEE Press, pp. 1044-1051, (2004).

[123] E. Clark, A. Hone, J. Timmis, *"**A Markov Chain Model of the B-Cell Algorithm**", 4th International Conference on Artificial Immune Systems (ICARIS2005)*, Banff, Canada. Lecture Notes in Computer Science 3627. Springer- Verlag, pp. 318-330, (2005).

[124] A. Watkins, *"**AIRS: A Resource Limited Artificial Immune Classifier**"*, Masters Dissertation, *Mississippi State University*, MS. USA, (2001).

[125] G. Marwah, A. Watkins, *"**Artificial Immune Systems for Classification: Some Issues**", 1st International Conference on Artificial Immune Systems (ICARIS 2002)*, Canterbury, UK. pp. 149-153, (2002).

[126] A. Watkins, L. Boggess, *"**A New Classifier Based on Resource Limited Artificial Immune Systems**"*, *Congress on Evolutionary Computation (CEC 2002)*, Honolulu, USA. IEEE Press, pp. 1546-1551, (2002).

[127] A. Watkins, L. Boggess, *"**A Resource Limited Artificial Immune Classifier**"*, *Congress on Evolutionary Computation (CEC 2002)*, Honolulu, USA. IEEE, pp. 926-931, (2002).

[128] A. Watkins, J. Timmis, *"**Artificial Immune Recognition System (AIRS): Revisions and Refinements**", 1st International Conference on Artificial Immune Systems (ICARIS 2002)*, Canterbury, UK. pp. 173-181, (2002).

[129] A. Watkins, J. Timmis, ***"Exploiting Parallelism Inherent in AIRS"***, *3rd International Conference on Artificial Immune Systems (ICARIS 2004)*, 247 Catania, Sicily. Lecture Notes in Computer Science 3239. Springer-Verlag, pp. 427-438, (2004).

[130] A. Watkins, J. Timmis, L. Boggess, ***"Artificial Immune Recognition System(AIRS): An Immune-Inspired Supervised Learning Algorithm"***, *Genetic Programming and Evolvable Machines*, 3(5), pp. 291-317, 2004.

[131] T. Knight, J. Timmis, ***"A Multi-Layered Immune Inspired Approach to Data Mining"***, *4th International Conference on Recent Advances in Soft Computing*, Nottingham, UK. pp. 266-271, (2002).

[132] T. Knight, J. Timmis, ***"A Multi-layered Immune Inspired Machine Learning Algorithm"***, In Lotfi, A., Garibaldi, M. (Eds.), *Applications and Science in Soft Computing*: Springer-Verlag, pp. 195-202, (2003).

[133] R. T. Alves, M. R. Delgado, H. S. Lopes, A. A. Freitas, ***"An Artificial Immune System for Fuzzy-Rule Induction in Data Mining"***, *Parallel Problem Solving from Nature* (PPSN-2004). Lecture Notes in Computer Science 3242. Springer-Verlag, pp. 1011-1020, (2004).

[134] A. Secker, A. Freitas, J. Timmis, ***"Towards a danger theory inspired artificial immune system for web mining"***, *In A Scime, editor, Web Mining: applications and techniques*, pp. 145-168. Idea Group, January (2005).

[135] E. Hart, P. Ross, ***"Improving SOSDM: Inspirations from the Danger Theory"***, In J. Timmis et al. (Eds.): *ICARIS 2003*, LNCS 2787, Springer-Verlag Berlin Heidelberg, pp. 194–203, (2003).

[136] B. Mihaljevic, A. Cvitas, M. Zagar, ***"Recommender System Model Based on Artificial Immune System"***, *28th Int. Conf. Information Technology Interfaces ITI 2006*, June 19-22, Cavtat, Croatia, pp. 367-372, (2006).

[137] J. Twycross, S. Cayzer, ***"An Immune System Approach to Document Classification"***, (HP Labs Technical Reports HPL-2002-292): HP Labs Bristol, UK, (2002).

[138] J.Twycross, S.Cayzer, ***"An Immune-based Approach to Document Classification"***, *International Conference on Intelligent Information Processing and Web Mining 2003*, Zakopane, Poland. Springer-Verlag, pp. 33-46, (2003).

[139] J. Twycross, *"**An Immune System Approach to Document Classification**"*, Masters Dissertation, *University of Sussex*, UK, (2002).

[140] J. Twycross, *"**An Immune System Approach to Document Classification**"*, (HP Labs Technical Reports HPL-2002-288): HP Labs Bristol, UK, (2002).

[141] J. Greensmith, "New ***Frontiers for an Artificial Immune System***", Masters Dissertation, *University of Leeds*, Leeds, (2003).

[142] N. Nanas, V. Uren, A. d. Roeck, *"**Nootropia: A User Profiling Model Based on a Self-Organizing Term Network**"*, *3rd International Conference on Artificial Immune Systems (ICARIS 2004)*, Catania, Italy. Lecture Notes in Computer Science 3239. Springer-Verlag, pp. 146-160, (2004).

[143] X. Hang, H. Dai, *"**An Immune Network Approach for Web Document Clustering**"*, *2004 IEEE/WIC/ACM International Conference on Web intelligence*, Beijing, China, (2004).

[144] N. Tang, V. R. Vemuri, *"**An Artificial Immune System Approach to Document Clustering**"*, *20th ACM Symposium on Applied Computing (SAC 2005)*, Santa Fe, USA. ACM Press, pp. 918–922, (2005).

[145] T. Morrison, U. Aickelin, *"**An Artificial Immune System as a Recommender for Web Sites**"*, *1st International Conference on Artificial Immune Systems (ICARIS 2002)*, Canterbury, UK, pp. 161-169, (2002).

[146] O. Nasraoui, D. Dasgupta, F. Gonzalez. *"**The Promise and Challenges of Artificial Immune System Based Web Usage Mining: Preliminary Results**"*, *Presented at the workshop on Web Analytics at Second SIAM International Conference on Data Mining (SDM)*, Arlington, VA, April 11-13, (2002).

[147] J. Timmis, M. Neal, *"**Investigating the Evolution and Stability of a Resource Limited Artificial Immune System**"*, *GECCO 2000 Workshop on Artificial Immune Systems and Their Applications*, Las Vegas, USA, pp. 40-41, (2000).

[148] B. Berendt, B. Mobasher, M. Spiliopoulou, J. Wiltshire, *"**Measuring the accuracy of sessionizers for web usage analysis**"*, in: *Proceedings of the Workshop on Web Mining at the First SIAM International Conference on Data Mining*, Chicago, IL, USA, (2001).

[149] M. Spiliopoulou, B. Mobasher, B. Berendt, M. Nakagawa, ***"A framework for the evaluation of session reconstruction heuristics in web-usage analysis"***, *INFORMS Journal on Computing* 15 (2), pp. 171–190, (2003).

[150] M. Morita, Y. Shinoda, ***"Information filtering based on user behavior analysis and best match text retrieval"***, in: *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Springer-Verlag New York, Inc., Dublin, Ireland, pp. 272–281, (1994).

[151] O. Nasraoui, F. González, C. Cardona, C. Rojas, D. Dasgupta. ***"A Scalable Artificial Immune System Model for Dynamic Unsupervised Learning"***. *In the proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, LNCS 2723, pp. 219-230, July 12-16, (2003).

[152] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa, ***"Improving the Effectiveness of Collaborative Filtering on Anonymous Web Usage Data"***, *In Proceedings of the IJCAI 2001 Workshop on Intelligent Techniques for Web Personalization (ITWP01)*, Seattle, WA, (2001).

[153] C.H. Cai, A.W.C. Fu, C.H. Cheng, W.W. Kwong, ***"Mining association rules with weighted items"***, *In Database Engineering and Applications Symposium*, *IDEAS'98*, pp. 68–77, (1998).

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

[155] J. Timmi s, M. Neal, J. Hunt, ***"An artificial immune system for data analysis",*** *Biosystems*, 55(1/3), pp. 143–150, (2000).

پیوست الف- کدهای الگوریتم پیشنهادی به زبان C++

```
#include "AISWUM.h"
#include "General.h"
#include "Preprocessing.h"

//---------------------------------------------------------------------------

void CalcPrecisionRecall (vector<vector<SessionEntry> > & sessions,
vector<AntiBody> & antibodies, vector<string> & urls);
void CalcPrecisionRecallWithOtherMethods (vector<vector<SessionEntry> > &
sessions, vector<string> & urls);
void CalcPrecisionRecallWithAntibodies (vector<vector<SessionEntry> > &
sessions, vector<string> & urls);
void GeneralKMeans(vector<string> & urls, int numofclustersingeneralkmeans,
bool writeinfile = false);
void CalcPrcCvgWithGTP(vector<string> & urls);
void CalcPrcCvgAntigensWithGTP(vector<string> & urls, int deltat);

//---------------------------------------------------------------------------

int main (int argc, char * argv [])
{
	vector<string> urls;
	vector<vector<SessionEntry> > sessions;
	map<string, double> urlsizes;

	ofstream fout ((default_input_file_name + ".urls.txt").c_str());
	cerr << "Starting the preprocessing phase..." << endl;
	preprocessing2 (argc, argv, sessions, urls);
	//preprocessing (argc, argv, sessions, urls, urlsizes);
	cerr << "Preprocessing done!" << endl;
	cerr << "Number of Sessions: " << sessions.size() << endl << "Number of
URLs: " << urls.size() << endl;
	cerr << endl;

	for (int i = 0; i < urls.size(); i++)
		fout << urls[i] << endl;

#ifdef DO_LOAD_ANTIGEN
	GeneralKMeans (urls, NumOfClusteringGeneralKmeans, true);
#endif

	vector<AntiBody> antibodies;
	AISWUM_Main (sessions, urls, antibodies);

	GeneralKMeans(urls, NumOfClusteringGeneralKmeans);

	CalcPrecisionRecallWithOtherMethods (sessions, urls);

	CalcPrecisionRecallWithAntibodies (sessions, urls);

	CalcPrcCvgWithGTP(urls);

	CalcPrcCvgAntigensWithGTP(urls, DeltaT);

	return 0;
}

//---------------------------------------------------------------------------

void CalcPrecisionRecall (vector<vector<SessionEntry> > & sessions,
vector<AntiBody> & antibodies, vector<string> & urls)
{
	vector<double> precisions, recalls;
```

```
	int i, session_idx = -1;
	double presum = 0.0, recsum = 0.0;

	map<string, int> url_to_index;
	for (i = 0; i < urls.size(); i++)
		url_to_index[urls[i]] = i;

	while (++session_idx < sessions.size())
	{
		// Checking, ... Danger Theory!
		if (Validity(sessions[session_idx]) < ValidityThreshold)
			continue;
		if (sessions[session_idx].size() < MinSessionSize)
			continue;

		// Creating the antigen,
		AntiGen tmp ((int)urls.size());
		for (i = 0; i < sessions[session_idx].size(); i++)

	tmp.cell[url_to_index[sessions[session_idx][i].weblog.url]] = 1;

		int min_idx = -1; double min_dist = 1.01;
		for (i = 0; i < antibodies.size(); i++)
		{
			double dista = AntibodyAntigenDistance (antibodies[i],
tmp, sessions, urls, false);
			if (dista < min_dist)
				min_dist = dista, min_idx = i;
		}

		int ones1 = 0, ones2 = 0, common = 0;
		for (i = 0; i < urls.size(); i++)
		{
			if (antibodies[min_idx].cell[i] > 0 && tmp.cell[i] > 0)
				ones1++, ones2++, common++;
			else if (antibodies[min_idx].cell[i] > 0)
				ones1++;
			else if (tmp.cell[i] > 0)
				ones2++;
		}

		precisions.push_back ((double)common/ones1);
		recalls.push_back ((double)common/ones2);

		presum += precisions[precisions.size()-1];
		recsum += recalls[recalls.size()-1];
	}

	ofstream fout
((string("precisions_recalls_mean.")+default_input_file_name).c_str());

	for (i = 0; i < precisions.size(); i++)
		fout << precisions[i] << ", " << recalls[i] << endl;
	fout << presum / precisions.size() << ", " << recsum / recalls.size() <<
endl;
}

//-----------------------------------------------------------------------------

void CalcPrecisionRecallWithOtherMethods (vector<vector<SessionEntry> > &
sessions, vector<string> & urls)
{
	vector<double> precisions, recalls;
	int i, session_idx = -1;
	double presum = 0.0, recsum = 0.0;

	map<string, int> url_to_index;
	for (i = 0; i < urls.size(); i++)
```

```
		url_to_index[urls[i]] = i;

	string line;
	vector<AntiBody> cells;
	ifstream fin ((default_input_file_name + ".kmeans.output").c_str());

	while (getline(fin, line))
	{
		cells.push_back(AntiBody());
		stringstream ss (line);
		int t;
		for (i = 0; i < urls.size(); i++)
			cells[cells.size()-1].cell.push_back(0);
		while (ss >> t)
			cells[cells.size()-1].cell[t] = 1;
	}

	while (++session_idx < sessions.size())
	{
		// Checking, ... Danger Theory!
		if (Validity(sessions[session_idx]) < ValidityThreshold)
			continue;
		if (sessions[session_idx].size() < MinSessionSize)
			continue;

		// Creating the antigen,
		AntiGen tmp ((int)urls.size());
		for (i = 0; i < sessions[session_idx].size(); i++)

	tmp.cell[url_to_index[sessions[session_idx][i].weblog.url]] = 1;

		int min_idx = -1; double min_dist = 1.01;
		for (i = 0; i < cells.size(); i++)
		{
			double dista = AntibodyAntigenDistance (cells[i], tmp,
sessions, urls, false);
			if (dista < min_dist)
				min_dist = dista, min_idx = i;
		}

		int ones1 = 0, ones2 = 0, common = 0;
		for (i = 0; i < urls.size(); i++)
		{
			if (cells[min_idx].cell[i] > 0 && tmp.cell[i] > 0)
				ones1++, ones2++, common++;
			else if (cells[min_idx].cell[i] > 0)
				ones1++;
			else if (tmp.cell[i] > 0)
				ones2++;
		}

		precisions.push_back ((double)common/ones1);
		recalls.push_back ((double)common/ones2);

		presum += precisions[precisions.size()-1];
		recsum += recalls[recalls.size()-1];
	}

	ofstream fout
((string("precisions_recalls_mean_")+"kmeans."+default_input_file_name).c_str(
));

	for (i = 0; i < precisions.size(); i++)
		fout << precisions[i] << ", " << recalls[i] << endl;
	fout << presum / precisions.size() << ", " << recsum / recalls.size() <<
endl;
}
```

```
//-------------------------------------------------------------

void CalcPrecisionRecallWithAntibodies (vector<vector<SessionEntry> > &
sessions, vector<string> & urls)
{
	vector<double> precisions, recalls;
	int i, session_idx = -1;
	double presum = 0.0, recsum = 0.0;

	map<string, int> url_to_index;
	for (i = 0; i < urls.size(); i++)
		url_to_index[urls[i]] = i;

	string line;
	vector<AntiBody> cells;
	ifstream fin ((default_input_file_name +
".FinalAntibodyLog.out").c_str());

	while (getline(fin, line))
	{
		cells.push_back(AntiBody());
		stringstream ss (line);
		int t;
		for (i = 0; i < urls.size(); i++)
			cells[cells.size()-1].cell.push_back(0);
		while (ss >> t)
			cells[cells.size()-1].cell[t] = 1;
	}

	while (++session_idx < sessions.size())
	{
		// Checking, ... Danger Theory!
		if (Validity(sessions[session_idx]) < ValidityThreshold)
			continue;
		if (sessions[session_idx].size() < MinSessionSize)
			continue;

		// Creating the antigen,
		AntiGen tmp ((int)urls.size());
		for (i = 0; i < sessions[session_idx].size(); i++)

	tmp.cell[url_to_index[sessions[session_idx][i].weblog.url]] = 1;

		int min_idx = -1; double min_dist = 1.01;
		for (i = 0; i < cells.size(); i++)
		{
			double dista = AntibodyAntigenDistance (cells[i], tmp,
sessions, urls, false);
			if (dista < min_dist)
				min_dist = dista, min_idx = i;
		}

		int ones1 = 0, ones2 = 0, common = 0;
		for (i = 0; i < urls.size(); i++)
		{
			if (cells[min_idx].cell[i] > 0 && tmp.cell[i] > 0)
				ones1++, ones2++, common++;
			else if (cells[min_idx].cell[i] > 0)
				ones1++;
			else if (tmp.cell[i] > 0)
				ones2++;
		}

		precisions.push_back ((double)common/ones1);
		recalls.push_back ((double)common/ones2);

		presum += precisions[precisions.size()-1];
		recsum += recalls[recalls.size()-1];
```

```
	}

	ofstream fout ((string("precisions_recalls_mean_")+"final-
antibody."+default_input_file_name).c_str());

	for (i = 0; i < precisions.size(); i++)
		fout << precisions[i] << ", " << recalls[i] << endl;
	fout << presum / precisions.size() << ", " << recsum / recalls.size() <<
endl;
}
//-------------------------------------------------------------
void CalcPrcCvgWithGTP(vector<string> & urls)
{
	int i, j, k, c, cellsnum = 0;
	string line;
	//vector<AntiBody> cells;
	vector <vector<int> > cells;
	vector <vector<int> > kmeanscells;
	vector<int> cellscount;

	ifstream ais_profiles ((default_input_file_name +
".AntibodyLog.out").c_str());
	ifstream kmeans_profiles ((default_input_file_name +
".kmeans.output").c_str());

	ofstream fout ((default_input_file_name +
".EvaluationAntibodiesWithGTP.output").c_str());
	ofstream fout1 ((default_input_file_name +
".EvaluationAntibodiesPrcWithGTP.output").c_str());
	ofstream fout2 ((default_input_file_name +
".EvaluationAntibodiesCvgWithGTP.output").c_str());

	cellscount.push_back(0);
	getline(ais_profiles, line);
	while (getline(ais_profiles, line))
	{
		cellsnum++;
		cells.push_back(vector<int>());
		if (line.substr(0,4) == "Time"){
			cellscount.push_back(cellsnum-1);
			getline(ais_profiles, line);
			cellsnum = 1;
		}
		stringstream ss (line);
		int t;

		while (ss >> t)
			cells[cells.size()-1].push_back(t);
	}

	cellscount.push_back(cellsnum);
	while (getline(kmeans_profiles, line))
	{
		kmeanscells.push_back(vector<int>());
		stringstream ss (line);
		int t;

		while (ss >> t)
			kmeanscells[kmeanscells.size()-1].push_back(t);
	}

	int cellindex = 0;
	for (i = 0; i < kmeanscells.size(); i++)
	{
		cellindex = 0;
		for (j = 0; j < (cellscount.size()-1); j++ )
		{
			cellindex += cellscount[j];
```

```
                    double max_prc = 0.0; double max_cvg = 0.0;
                    for (c = 0; c < cellscount[j+1]; c++)
                    {
                           int common = 0;
                           for (k = 0; k < kmeanscells[i].size(); k++)
                           {
                                  for (int l = 0; l <
cells[cellindex+c].size(); l++)
                                         if (kmeanscells[i][k] ==
cells[cellindex+c][l])
                                                common++;
                           }
                           if ((double)common/cells[cellindex+c].size() >
max_prc)
                                         max_prc =
(double)common/cells[cellindex+c].size();
                           if ((double)common/kmeanscells[i].size() > max_cvg)
                                         max_cvg =
(double)common/kmeanscells[i].size();

                    }
                    if ( max_prc >= PrcCvgTh && max_cvg >= PrcCvgTh )
                           fout << j << ", " << i << endl;
                    if (max_prc >= PrcCvgTh)
                           fout1 << j << ", " << i << endl;
                    if (max_cvg >= PrcCvgTh)
                           fout2 << j << ", " << i << endl;
             }
      }

}

//-----------------------------------------------------------------
void GeneralKMeans(vector<string> & urls, int numofclustersingeneralkmeans,
bool writeinfile)
{
      vector<vector<int> > clusters1;
      vector<AntiBody> centroids1 (numofclustersingeneralkmeans);
      string line;
      vector<AntiBody> cells1;
      int i, j;

      ifstream fin ((default_input_file_name +
".ais_cells_not_csv.data").c_str());

      while (getline(fin, line))
      {
             cells1.push_back(AntiBody());
             stringstream ss (line);
             int t;

             while (ss >> t)
                    cells1[cells1.size()-1].cell.push_back(t);

             if (cells1[cells1.size()-1].cell.size() != urls.size())
             {
                    cout << "Error in input file...\nThe number of urls does
not match the line entries in file..." << endl;
                    return;
             }
      }

      kmeans (cells1, clusters1, centroids1, int(urls.size()),
MaxKmeansIterationsAllowed, true);

      ofstream fout  ((default_input_file_name + ".kmeans.output").c_str());
      ofstream fout2 ((default_input_file_name +
".URLsOfkmeans.output").c_str());
```

```
	ofstream fout3 ((default_input_file_name +
".OrderedAntigens.output").c_str());

	for (i = 0; i < centroids1.size(); i++)
	{
		fout2 << "Profile number: " << i << endl;
		for (j = 0; j < urls.size(); j++)
		{
			if(centroids1[i].cell[j] > 0)
			{
				fout << j << " ";
				fout2 << urls[j] << endl;
			}
		}
		fout << endl;
	}

	if (writeinfile)
	{
		for (i = 0; i < clusters1.size(); i++)
			for (j = 0; j < clusters1[i].size(); j++)
			{
				for (int k = 0; k <
cells1[clusters1[i][j]].cell.size(); k++)
					fout3 << cells1[clusters1[i][j]].cell[k] << "
";
				fout3 << endl;
			}
	}

	return;
}

//-----------------------------------------------------------------------
void CalcPrcCvgAntigensWithGTP(vector<string> & urls, int deltat)
{
	int i, j, k, c,l, antigenscount = 0;
	string line;

	vector<vector<int> > cells;
	vector<vector<int> > aiscells;
	vector<vector<int> > kmeanscells;
	vector<int> cellscount;

	ifstream ais_antigens ((default_input_file_name +
".ais_cells_short.data").c_str());
	ifstream kmeans_profiles ((default_input_file_name+
".kmeans.output").c_str());
	ifstream AISWUM_profiles ((default_input_file_name +
".FinalAntibodyLog.out").c_str());

	ofstream fout ((default_input_file_name +
".EvaluationAntigensWithGTP.output").c_str());
	ofstream fout1 ((default_input_file_name +
".EvaluationAntigensPrcWithGTP.output").c_str());
	ofstream fout2 ((default_input_file_name +
".EvaluationAntigensCvgWithGTP.output").c_str());

	ofstream fiout ((default_input_file_name +
".EvaluationAntibodiesWithAntigens.output").c_str());
	ofstream fiout1 ((default_input_file_name +
".EvaluationAntibodiesPrcWithAntigens.output").c_str());
	ofstream fiout2 ((default_input_file_name +
".EvaluationAntibodiesCvgWithAntigens.output").c_str());

	while (getline(ais_antigens, line))
	{
		cells.push_back(vector<int>());
```

```
		stringstream ss (line);
		int t;

		while (ss >> t)
			cells[cells.size()-1].push_back(t);
	}

	while (getline(kmeans_profiles, line))
	{
		kmeanscells.push_back(vector<int>());
		stringstream ss (line);
		int t;

		while (ss >> t)
			kmeanscells[kmeanscells.size()-1].push_back(t);
	}

	int loopcriterion = 0;

	for (i = 0; i < kmeanscells.size(); i++)
		for (j = 0 /*deltat-1*/; j < cells.size(); j++)
		{
			double max_prc = 0.0; double max_cvg = 0.0;
			if (j < deltat)
				loopcriterion = j;
			else loopcriterion = deltat;
			for (c = 0; c <= loopcriterion; c++)
			{
				int common = 0;
				for (k = 0; k < kmeanscells[i].size(); k++)
				{
					for (l = 0; l < cells[j-c].size(); l++)
						if (kmeanscells[i][k] == cells[j-
c][l])
							common++;
				}
				if ((double)common/cells[j-c].size() > max_prc)
						max_prc = (double)common/cells[j-
c].size();
				if ((double)common/kmeanscells[i].size() > max_cvg)
						max_cvg =
(double)common/kmeanscells[i].size();
			}

			if ( max_prc >= PrcCvgTh && max_cvg >= PrcCvgTh )
				fout << j << ", " << i << endl;
			if (max_prc >= PrcCvgTh)
				fout1 << j << ", " << i << endl;
			if (max_cvg >= PrcCvgTh)
				fout2 << j << ", " << i << endl;

		}


	while (getline(AISWUM_profiles, line))
	{
		aiscells.push_back(vector<int>());
		stringstream ss (line);
		int t;

		while (ss >> t)
			aiscells[aiscells.size()-1].push_back(t);
	}

	for (i = 0; i < aiscells.size(); i++)
		for (j = 0 /*deltat-1*/; j < cells.size(); j++)
		{
			double max_prc = 0.0; double max_cvg = 0.0;
```

```
                    if (j < deltat)
                           loopcriterion = j;
                    else loopcriterion = deltat;
                    for (c = 0; c <= loopcriterion; c++)
                    {
                           int common = 0;
                           for (k = 0; k < aiscells[i].size(); k++)
                           {
                                  for (l = 0; l < cells[j-c].size(); l++)
                                         if (aiscells[i][k] == cells[j-c][l])
                                                common++;
                           }
                           if ((double)common/cells[j-c].size() > max_prc)
                                         max_prc = (double)common/cells[j-
c].size();
                           if ((double)common/aiscells[i].size() > max_cvg)
                                         max_cvg =
(double)common/aiscells[i].size();
                    }
                    if ( max_prc >= PrcCvgTh && max_cvg >= PrcCvgTh )
                           fiout << j << ", " << i << endl;
                    if (max_prc >= PrcCvgTh)
                           fiout1 << j << ", " << i << endl;
                    if (max_cvg >= PrcCvgTh)
                           fiout2 << j << ", " << i << endl;

             }

       return;
}

//-----------------------------------------------------------------


#ifndef __GENERAL_H__

       #define __GENERAL_H__

//-------------------------------------------------------------------------

#pragma warning (disable : 4018)

//-------------------------------------------------------------------------

#include <set>
#include <map>
#include <list>
#include <ctime>
#include <cmath>
#include <queue>
#include <vector>
#include <string>
#include <limits>
#include <cassert>
#include <fstream>
#include <sstream>
#include <numeric>
#include <iostream>
#include <algorithm>

using namespace std;

//-------------------------------------------------------------------------

const string default_input_file_name  = /**"OneWeekTest";//*/ /*"m.990421";*/
"OneWeekUofS_access_log";
//const string default_input_file_name  = "OneMonthTest-April";
const string default_output_file_name = "output.txt";
```

```
//const string default_url_sizes_file = "url_sizes.txt";

const int default_session_time_interval = 30 * 60;  // In Seconds,

//-----------------------------------------------------------------------------

#define MIN_PRC_CVG_DIST_USAGE

//#define VALIDITY_WEIGHTED_STIMULATION

#define HARMONIC_AVG

#define SUPPRESSION_COSTIMULATION

#define STIMULATION_REMOVE_CRITERION

#define LOG_AIS_CELLS_IN_FILE

#define DO_LOAD_ANTIGEN

//-----------------------------------------------------------------------------

#define SQR(a) (a)*(a)


//#define cerr logfile
ofstream logfile ("logfile.txt");

//-----------------------------------------------------------------------------
// Problem Parameters:

const int           NbInit                          = 100;
const int           NbMax                                 = 130;
const int           MaxDist                               = 7;
const double  ScaleInit                             = 0.1;
const int           MinSessionSize                  = 4;
const double  ValidityThreshold                 = 0.3;
const double  MinWeightTh                       = 0.3;
const double  Alpha                             = 0.7;
const int           RehabilitationPeriod            = 10;
const double  CloningFactor               = 0.76;
const int           StagnationTime                  = 15;
const int           MinStimulationNumber            = 3;
const int           MinDiffRequiredForMutation = 5;
const int           MaxKmeansIterationsAllowed = 4;
const double  ProbToAttachToAnotherCluster= 0.5;
const int           ValidUrlLenInNumberOfSlashes= 5;
const int           MinAge                      = 5;
const int           MaxAge                      = 30;
const double  BaseMutationProbability           = 0.1;
const double    MutationFactor                  = 0.5;
const double  ContentmentTh               = 0.0011;
const int           ValidNumOnesAfterMutation  = 3;
const double  PrcCvgTh                          = 0.4
const int           NumOfClusteringGeneralKmeans     = 20;
const int           DeltaT                      = 5;
//-----------------------------------------------------------------------------

struct WebLogEntry
{
       time_t utc_datetime;
       // Number of seconds elapsed since midnight (00:00:00), January 1, 1970
       // Coordinated Universal Time (UTC)

       vector<string> data;
       // 0: server,       1: IP,              2: method and url,  3: mimetype,
              4: str1
```

```
	// 5: date & time,  6: str2,		7: str3,		8:
str4,		9: str5
	// 10: referer		11: device

	string url;
	double freq, duration, interest;
	double pagesize;

	WebLogEntry () : data (12), utc_datetime (0) {}
	WebLogEntry (const string & line) : data (12)
	{
		int i = 0;
		string part;
		stringstream ss (line);
		while (getline(ss, part, '|'))
		{
//			assert(i<12);
			if (i >= 12) break;
			data[i++] = part;
		}

		// Setting entry date-time in UTC format.
		string s = data[5];
		for (int i = 0; i < s.size (); i++)
			if (s[i] == ':' || s[i] == '-' || s[i] == '/')
				s[i] = ' ';
		stringstream ss1 (s);
		int hou, min, sec, dd, mm, yy;
		ss1 >> yy >> mm >> dd >> hou >> min >> sec;
		struct tm tt;
		tt.tm_year = yy - 1900;
		tt.tm_mon  = mm - 1;
		tt.tm_mday = dd;
		tt.tm_hour = hou;
		tt.tm_min  = min;
		tt.tm_sec  = sec;
		tt.tm_isdst= 0;

		if (tt.tm_year < 0 && tt.tm_year > 110) tt.tm_year = 100;
		if (tt.tm_mon  < 0 && tt.tm_mon  > 11 ) tt.tm_mon  = 11 ;
		if (tt.tm_mday < 1 && tt.tm_mday > 31 ) tt.tm_mday = 1 ;
		if (tt.tm_hour < 0 && tt.tm_hour > 23 ) tt.tm_hour = 23 ;
		if (tt.tm_min  < 0 && tt.tm_min  > 59 ) tt.tm_min  = 59 ;
		if (tt.tm_sec  < 0 && tt.tm_sec  > 59 ) tt.tm_sec  = 59 ;

		utc_datetime = mktime (&tt);
	}
};

//-----------------------------------------------------------------------------

struct SessionEntry
{
	string browses;
	WebLogEntry weblog;

	SessionEntry () {}
	SessionEntry (const SessionEntry & se) : browses (se.browses), weblog
(se.weblog) {}
	SessionEntry (const string & s, const WebLogEntry & w) : browses (s),
weblog (w) {}
};

//-----------------------------------------------------------------------------

struct AntiGen
{
	vector<short> cell;
```

```
	int index_in_sessions;

	AntiGen () {};
	AntiGen (const AntiGen & c) : cell (c.cell) {};
	AntiGen (const vector<short> & c) : cell (c) {};
	AntiGen (const int n) : cell (n, 0) {};
};

//---------------------------------------------------------------------------

struct AntiBody
{
	vector<short> cell;
	double stimulation_level, scale, weight, dummy;
	int birth, last_stimulation, number_of_stimulations;


	AntiBody () : stimulation_level (0.0), scale (ScaleInit), weight (0.0),
birth (0), last_stimulation (0) , number_of_stimulations (0) {};
	AntiBody (const int n, int b = 0) : cell (n, 0), stimulation_level
(0.0), scale (ScaleInit), weight (0.0), birth (b), last_stimulation (0) ,
number_of_stimulations (0){};
	AntiBody (const AntiBody & c) : cell (c.cell), stimulation_level
(c.stimulation_level), scale (c.scale), weight (c.weight), birth (c.birth),
last_stimulation (c.last_stimulation) , number_of_stimulations
(c.number_of_stimulations), dummy (c.dummy) {};
	AntiBody (const vector<short> & c, int b) : cell (c), stimulation_level
(0.0), scale (ScaleInit), weight (0.0), birth (b), last_stimulation (0) ,
number_of_stimulations (0) {};

	bool operator < (const AntiBody & ab) const
	{
#ifdef STIMULATION_REMOVE_CRITERION
		return stimulation_level < ab.stimulation_level;
#else
		return birth > ab.birth;
#endif
	}
};

//---------------------------------------------------------------------------

ostream & operator << (ostream & os, AntiBody & ab)
{
	os << "URLs in session:" << endl;
	for (int i = 0; i < ab.cell.size(); i++)
		if (ab.cell[i] > 0)
			os << i << ", ";
	os << endl;
	os << "Stimulation Level: " << ab.stimulation_level << endl;
	os << "Scale: " << ab.scale << endl;
	os << "Weight: " << ab.weight << endl;
	os << "Birth: " << ab.birth << endl;
	os << "Last Stimulation: " << ab.last_stimulation << endl;
	os << "Number of Stimulations: " << ab.number_of_stimulations << endl;
	os << endl;

	return os;
}

/*---------------------------------------------------------------------------
---------------------------------------------------------------------------
--------------------------Function Definitions-----------------------
---------------------------------------------------------------------------*/

bool ReadUrlSizes (map<string, double> & urlsizes)
{
	ifstream fin ((default_input_file_name + ".url_sizes.txt").c_str());
```

```
	string url; double sz;

	if (fin.fail())
		return false;

	while (fin >> url >> sz)
		urlsizes[url] = sz;

	return true;
}

//-----------------------------------------------------------------------------

bool UrlsEqual (const string & url1, const string & url2)
{
	int a = int(url1.size()) - int(url2.size()), b = int(url2.size()) -
int(url1.size());
	return ((a >= 0 && url2 == url1.substr(a)) ||(b >= 0 && url1 ==
url2.substr(b)));
}

//-----------------------------------------------------------------------------

void  CalcDurations (const vector<SessionEntry> & session, vector<double> &
durations)
{
	durations.resize (session.size());

	int i;
	for (i = 0; i < session.size() - 1; i++)
		durations[i] =  abs ((double)session[i].weblog.utc_datetime -
session[i+1].weblog.utc_datetime) / session[i].weblog.pagesize;

	double sum = 0.0, max_duration = *max_element (durations.begin(),
durations.end());
	for (i = 0; i < session.size() - 1; i++)
		durations[i] /= max_duration,
		sum += durations[i];

	durations[i] = sum / (session.size()-1);
}

//-----------------------------------------------------------------------------

// Modifies 'session': assigning the session duration and frequency.
// Returns: the 'session' interest.
// Remarks: Deletes duplicate URLs in 'session'.
double CalcSessionDurationsFrequenciesAndInterest (vector<SessionEntry> &
session)
{
	string url;
	vector<double> durations;
	int i, j, cnt, n = (int)session.size();
	double sum_durations, sum_interest = 0.0;

	CalcDurations (session, durations);
	for (j = 0; j < session.size(); j++)
	{
		cnt = 1; sum_durations = durations[j];
		url = session[j].weblog.url;
		for (i = j+1; i < session.size(); i++)
			if (UrlsEqual(url, session[i].weblog.url))
			{
				sum_durations += durations[i];
				cnt++;
				session.erase (session.begin() + i);
				durations.erase (durations.begin() + i);
				i--;
```

```
                        }
                session[j].weblog.freq = (double)cnt/n;
                session[j].weblog.duration = sum_durations/cnt;             // An
average of mutiple pages with the same URL.

#ifdef HARMONIC_AVG
        session[j].weblog.interest =
(2*session[j].weblog.freq*session[j].weblog.duration)/(session[j].weblog.freq+
session[j].weblog.duration);
#else
        session[j].weblog.interest =
(session[j].weblog.freq+session[j].weblog.duration)/2;
#endif

                sum_interest += session[j].weblog.interest;
        }

        return sum_interest/session.size();
}

//-----------------------------------------------------------------------------
// Returns the distance between two URLs on a tree-basis principle

int dist (const string & url1, const string & url2)
{
        int i = -1, j = -1, i_new, j_new;
        while (0 == 0)
        {
                i_new = (int)url1.find_first_of ('/', i+1);
                j_new = (int)url2.find_first_of ('/', j+1);

                if ((size_t)i_new == string::npos || (size_t)j_new ==
string::npos)
                        break;
                if (url1.substr(i+1, i_new-i-1) != url2.substr(j+1, j_new-j-1))
                        break;

                i = i_new; j = j_new;
        }

        int cnt1 = 0, cnt2 = 0;
        if (i_new != string::npos)
                for ( ; i < url1.size()-1; i++)
                        if (url1[i] == '/')
                                cnt1++;
        if (j_new != string::npos)
                for ( ; j < url2.size()-1; j++)
                        if (url2[j] == '/')
                                cnt2++;

        return cnt1 + cnt2;
}

//-----------------------------------------------------------------------------

double Similarity (const string & url1, const string & url2)
{
        return 1.0 - (double)dist(url1, url2) / MaxDist;
}

//-----------------------------------------------------------------------------

// Returns session consistancy.
double Consistancy (const vector<SessionEntry> & session)
{
        int i, j, P = (int)session.size();
        double sum = 0.0;
        for (i = 0; i < P - 1; i++)
```

```
            for (j = i + 1; j < P; j++)
                    sum += Similarity (session[i].weblog.url,
session[j].weblog.url);

        return sum / (P*(P-1)/2);
}

//-----------------------------------------------------------------------

// Returns session validiuty.
double Validity (vector<SessionEntry> & session)
{
        double consistancy = Consistancy (session);
        double interest = CalcSessionDurationsFrequenciesAndInterest (session);

#ifdef HARMONIC_AVG
        return (2*interest*consistancy)/(interest+consistancy);
#else
        return (interest+consistancy)/2;
#endif

}

//-----------------------------------------------------------------------

double AntibodyAntigenDistance (AntiBody & ab, const AntiGen & ag, const
vector<vector<SessionEntry> > & sessions, const vector<string> & urls, bool
weighted_dist = false)
{
        int i, abc = 0, agc = 0;
        double dist, sum = 0.0;
        for (i = 0; i < ab.cell.size(); i++)
        {
                if (ab.cell[i] > 0) abc++;
                if (ag.cell[i] > 0) agc++;

                if (weighted_dist)
                {
                        int j;
                        for (j = 0; j < sessions[ag.index_in_sessions].size();
j++)
                                if (sessions[ag.index_in_sessions][j].weblog.url ==
urls[i])
                                        break;
//                      assert (j >= sessions[ag.index_in_sessions]);

                        if (j < sessions[ag.index_in_sessions].size())
                                sum += (ab.cell[i] > 0 && ag.cell[i] > 0 ?
sessions[ag.index_in_sessions][j].weblog.interest : 0.0);
                }
                else
                        sum += (ab.cell[i] > 0 && ag.cell[i] > 0 ? 1.0 : 0.0);
        }

        double prc = sum / abc;
        double cvg = sum / agc;

#ifdef MIN_PRC_CVG_DIST_USAGE
        dist= 1.0 - min(prc, cvg);
#else
        dist = 1.0 - sqrt(prc * cvg);
#endif

        return dist;
}

//-----------------------------------------------------------------------
```

```
double AntibodyAntibodyDistance (const AntiBody & ab1, const AntiBody & ab2)
{
	int i, abc = 0, agc = 0;
	double dist, sum = 0.0;
	for (i = 0; i < ab1.cell.size(); i++)
	{
		if (ab1.cell[i] > 0) abc++;
		if (ab2.cell[i] > 0) agc++;

		sum += (ab1.cell[i] > 0 && ab2.cell[i] > 0 ? 1.0 : 0.0);
	}

	double prc = sum / abc;
	double cvg = sum / agc;

#ifdef MIN_PRC_CVG_DIST_USAGE
	dist= 1.0 - min(prc, cvg);
#else
	dist = 1.0 - sqrt(prc * cvg);
#endif

	return dist;
}

//-----------------------------------------------------------------------------

double Weight (AntiBody & ab, const AntiGen & ag, vector<vector<SessionEntry>
> & sessions, const vector<string> & urls)
{
	double dist = AntibodyAntigenDistance (ab, ag, sessions, urls);
	return exp (-1 * (dist * dist / (2 * ab.scale)));
}

//-----------------------------------------------------------------------------

double Weight (AntiBody & ab1, const AntiBody & ab2, double & dist)
{
	dist = AntibodyAntibodyDistance (ab1, ab2);
	return exp (-1 * (dist * dist / (2 * ab1.scale)));
}

//-----------------------------------------------------------------------------

// Culsters the 'antibodies' vector to 'centroids.size()' , using
void kmeans (const vector<AntiBody> & antibodies, vector<vector<int> > &
clusters, vector<AntiBody> & centroids, int antibodysize, int
num_of_iterations = MaxKmeansIterationsAllowed, bool first_time = false)
{
	int i, j, c, k, kmeans_itr = 0, n = (int)centroids.size();

	if (first_time)
		for (i = 0 ; i < n; i++)
			centroids[i] = antibodies[i];

	i = 0; j = 0;
	while (kmeans_itr++ < num_of_iterations)
	{
		clusters.clear ();
		clusters = vector<vector<int> > (n, vector<int> ());

		// Simply determine which antibody belongs to which cluster,
		for (i = 0; i < antibodies.size(); i++)
		{
			int min_dist = numeric_limits<int>::max(); int min_idx = -
1; int dist = 0;
			for (j = 0; j < n; j++)
			{
				dist = 0;
```

```
                        for (k = 0; k < antibodysize; k++)
                                if ( (centroids[j].cell[k] <= 0 &&
antibodies[i].cell[k] >=1) || (centroids[j].cell[k] >= 1 &&
antibodies[i].cell[k] <= 0)  ) dist++;

                        if (min_dist > dist)
                                min_dist = dist, min_idx = j;
                        if (min_dist == dist && (double)rand()/RAND_MAX <
ProbToAttachToAnotherCluster)    // A simple probability to attach to another
cluster!
                                min_idx = j;
                }
                clusters[min_idx].push_back (i);
        }

        // Erasing empty clusters.
        for (i = 0; i < n; i++)
                if (clusters[i].size() == 0)
                        clusters.erase (clusters.begin()+i),
                        centroids.erase(centroids.begin()+i),
                        i--, n--;

        // Updating cluster centriods,
        for (c = 0; c < clusters.size(); c++)
        {
                int min_dist = numeric_limits<int>::max(); int min_idx = -
1; int cnt = 0;
                for (i = 0; i < clusters[c].size(); i++)
                {
                        cnt = 0;
                        for (j = 0; j < clusters[c].size(); j++)
                        {
                                if( i == j )
                                        continue;

                                for (k = 0; k < antibodysize; k++)
                                        if
((antibodies[clusters[c][i]].cell[k] <= 0 &&
antibodies[clusters[c][j]].cell[k] >= 1)
                                        || (antibodies[clusters[c][i]].cell[k]
>= 1 && antibodies[clusters[c][j]].cell[k] <= 0))
                                                cnt++;

                        } // for j

                        if ( cnt < min_dist )
                                min_dist = cnt, min_idx = i;
                } // for i

                centroids[c] = antibodies[clusters[c][min_idx]];

        }//for c

    }//while
}

//-----------------------------------------------------------------------------
void UpdateCentroidofCluster (const vector<AntiBody> & antibodies,
vector<vector<int> > & clusters, vector<AntiBody> & centroids, int clusternum,
int antibodysize)
{
    int i, j, k;
    int min_dist = numeric_limits<int>::max(); int min_idx = -1; int cnt =
0;
    for (i = 0; i < clusters[clusternum].size(); i++)
    {
        cnt = 0;
        for (j = 0; j < clusters[clusternum].size(); j++)
```

```
		{
			if( i == j )
				continue;

			for (k = 0; k < antibodysize; k++)
				if ((antibodies[clusters[clusternum][i]].cell[k] <=
0 && antibodies[clusters[clusternum][j]].cell[k] >= 1)
				|| (antibodies[clusters[clusternum][i]].cell[k] >= 1
&& antibodies[clusters[clusternum][j]].cell[k] <= 0))
					cnt++;

		} // for j

		if ( cnt < min_dist )
			min_dist = cnt, min_idx = i;
	} // for i

	centroids[clusternum] = antibodies[clusters[clusternum][min_idx]];

}
//---------------------------------------------------------------------------

double CalcClusterDissimilarity (const vector<AntiBody> & antibodies, const
vector<int> & cluster)
{
	int i, j;

	double sum = 0.0;
	double sumdis = 0.0;
	for (i = 0; i < cluster.size(); i++)
	{
		sum = 0.0;
		for (j = 0; j < cluster.size(); j++)
			if (i != j)
				sum += AntibodyAntibodyDistance
(antibodies[cluster[i]], antibodies[cluster[j]]);
		sumdis += sum / cluster.size();
	}
	return (double)sumdis / cluster.size();
}

//---------------------------------------------------------------------------

#endif
#ifndef __PREPROCESSING_H__

	#define __PREPROCESSING_H__

//---------------------------------------------------------------------------

#include "General.h"

//---------------------------------------------------------------------------
int preprocessing2 (int argc, char * argv [], vector<vector<SessionEntry> > &
sessions, vector<string> & urls)
{
	string in_file  = default_input_file_name ;
	string out_file = default_output_file_name;

	if (argc > 1)
		in_file  = argv[1];
	if (argc > 2)
		out_file = argv[2];
	else
		out_file = in_file + ".out";

	ifstream fin  ( in_file.c_str());
	ofstream fout (out_file.c_str());
```

```
	int idx, i;
	string line;
	vector<string> browses;
	vector<WebLogEntry> weblog;

	while (getline(fin, line))
	{
		WebLogEntry t;
		int i1, i2, i3, i4, i5;
		string t2;
		i1 = (int)line.find_first_of (' ');
		i2 = (int)line.find_first_of ('[', i1);
		i3 = (int)line.find_first_of (']', i2);
		i4 = (int)line.find_first_of ('\"', i3);
		i5 = (int)line.find_last_of  ('\"');

		t.data[1] = (line.substr (0, i1));
		// data[1] is the referer IP or internet address.

		string mons [] = {"Jan", "Feb", "Mar", "Apr", "May", "Jun",
"Jul", "Aug", "Sep", "Oct", "Nov", "Dec"};
		t2 = line.substr (i2+1, i3-i2-1);
		for (int ii = 0; ii < t2.size (); ii++)
			if (t2[ii] == ':' || t2[ii] == '-' || t2[ii] == '/')
				t2[ii] = ' ';

		stringstream ss1 (t2);
		int hou, min, sec, dd, yy; string mm;
		ss1 >> dd >> mm >> yy >> hou >> min >> sec;
		struct tm tt;
		tt.tm_year = yy - 1900;
		tt.tm_mon  = (int)(find(mons, mons+12, mm)-mons); // Month index.
		tt.tm_mday = dd;
		tt.tm_hour = hou;
		tt.tm_min  = min;
		tt.tm_sec  = sec;
		tt.tm_isdst= 0;

		if (tt.tm_year < 0 && tt.tm_year > 110) tt.tm_year = 100;
		if (tt.tm_mon  < 0 && tt.tm_mon  > 11 ) tt.tm_mon  = 11 ;
		if (tt.tm_mday < 0 && tt.tm_mday > 31 ) tt.tm_mday = 1  ;
		if (tt.tm_hour < 0 && tt.tm_hour > 23 ) tt.tm_hour = 23 ;
		if (tt.tm_min  < 0 && tt.tm_min  > 59 ) tt.tm_min  = 59 ;
		if (tt.tm_sec  < 0 && tt.tm_sec  > 59 ) tt.tm_sec  = 59 ;

		t.utc_datetime = mktime (&tt);

		t2 = line.substr (i4+1, i5-i4-1);
		int i6 = (int)t2.find_first_of('/');
		t.url = t2.substr (i6, i5-i6);
		t.url = t.url.substr (0, t.url.find_first_of(' '));
		t.url = t.url.substr (0, t.url.find_first_of('?'));

		if (t.url.find ("gif") == string::npos && t.url.find ("jpg") ==
string::npos)
		if (t.url.find ("..") == string::npos && t.url.find ("//") ==
string::npos)
//		URL is valid:
		{
			browses.push_back (line);
			if (t.url.size()>10 && "index.html" ==
t.url.substr(t.url.size()-10))
				t.url = t.url.substr(0, t.url.size()-10);
			if (t.url[t.url.size()-1] != '/')
				t.url += '/';

			int slashcnt = 0;
```

```
                        for (i = 0; i < t.url.size(); i++)
                        {
                                if (slashcnt >= ValidUrlLenInNumberOfSlashes) break;
                                if (t.url[i] == '/') slashcnt++;
                        }
                        if (i < t.url.size())
                                t.url = t.url.substr (0, i);

                        t.pagesize = 1;
                        weblog.push_back (t);
                        for (i = 0; i < urls.size(); i++)
                                if (UrlsEqual(urls[i], t.url))
                                        break;
                        if (i >= urls.size())
                                urls.push_back (t.url);
                }
        }

// Putting browsed pages in sessions, based on their refered IP and time,
//      vector<vector<SessionEntry> > sessions;
        for (idx = 0; idx < weblog.size(); idx++)
        {
                for (i = (int)sessions.size()-1; i > -1; i--)
                        // If a same IP address and the difference is less than 30
minutes
                        if (sessions[i][0].weblog.data[1] == weblog[idx].data[1]
&& abs(long(sessions[i][0].weblog.utc_datetime - weblog[idx].utc_datetime)) <
default_session_time_interval)
                        {
                                sessions[i].push_back(SessionEntry(browses[idx],
weblog[idx]));
                                break;
                        }

                if (i <= -1)
                {
                        vector<SessionEntry> tmp (1, SessionEntry(browses[idx],
weblog[idx]));
                        sessions.push_back(tmp);
                }
        }

        // Writing output,
        for (idx = 0; idx < sessions.size(); idx++)
        {
                fout << "Session " << idx+1 << ":" << endl;
                for (i = 0; i < sessions[idx].size(); i++)
                        fout << sessions[idx][i].browses << " --- " <<
sessions[idx][i].weblog.utc_datetime << endl;
                fout << endl;
        }

        return 0;
}

//-----------------------------------------------------------------------------

int preprocessing (int argc, char * argv [], vector<vector<SessionEntry> > &
sessions, vector<string> & urls, map<string, double> & urlsizes)
{
        string in_file  = default_input_file_name ;
        string out_file = default_output_file_name;

        if (argc > 1)
                in_file  = argv[1];
        if (argc > 2)
                out_file = argv[2];
        else
```

```
		out_file = in_file + ".out";

	ifstream fin  ( in_file.c_str());
	ofstream fout (out_file.c_str());

	int idx, i;
	string line;
	vector<string> browses;
	vector<WebLogEntry> weblog;

	// Keeping only Text Content in browses,
	while (getline(fin, line))
	{
		WebLogEntry t (line);
		if (t.data[3].find ("text") != string::npos) // If found,
		if (t.data[2].find ("gif") == string::npos && t.data[2].find
("jpg") == string::npos)
		if (t.data[2].find ("..") == string::npos && t.data[2].find
("//") == string::npos)
		{
			browses.push_back (line);
			int int_tmp = 4;
			if (t.data[2].substr(0,4) == "HEAD")
				int_tmp = 5;
			t.url = t.data[2].substr(int_tmp, t.data[2].size() - 9 -
int_tmp);
			t.url = t.url.substr (0, t.url.find_first_of ('?'));
			if (t.url.size()>10 && "index.html" ==
t.url.substr(t.url.size()-10))
				t.url = t.url.substr(0, t.url.size()-10);

			if (t.url[t.url.size()-1] != '/')
				t.url += '/';
			if ("/music/machines" == t.url.substr(0, 15))
				t.url = t.url.substr(15);
			if ("/music" == t.url.substr(0, 6))
				t.url = t.url.substr(6);
			if ("/machines" == t.url.substr(0, 9))
				t.url = t.url.substr(9);

			int slashcnt = 0;
			for (i = 0; i < t.url.size(); i++)
			{
				if (slashcnt >= ValidUrlLenInNumberOfSlashes) break;
				if (t.url[i] == '/') slashcnt++;
			}
			if (i < t.url.size())
				t.url = t.url.substr (0, i);

			t.pagesize = urlsizes[t.url];
			if (t.pagesize == 0) t.pagesize =
floor(((double)rand()/RAND_MAX)*11.5+1.0); // To avoid its being zero!
			weblog.push_back  (t);
			for (i = 0; i < urls.size(); i++)
			{
				if (UrlsEqual(urls[i], t.url))
					break;
			}
			if (i >= urls.size()) urls.push_back (t.url);
		}
	}

// Putting browsed pages in sessions, based on their refered IP and time,
	for (idx = 0; idx < weblog.size(); idx++)
	{
//		for (i = 0; i < sessions.size(); i++)
		for (i = (int)sessions.size()-1; i > -1; i--)
```

```
                    // If a same IP address and the difference is less than 30
minutes
                    if (sessions[i][0].weblog.data[1] == weblog[idx].data[1]
&& abs(long(sessions[i][0].weblog.utc_datetime - weblog[idx].utc_datetime)) <
default_session_time_interval)
                    {
                           sessions[i].push_back(SessionEntry(browses[idx],
weblog[idx]));
                           break;
                    }

             if (i <= -1)
             {
                    vector<SessionEntry> tmp (1, SessionEntry(browses[idx],
weblog[idx]));
                    sessions.push_back(tmp);
             }
       }

       // Writing output,
       for (idx = 0; idx < sessions.size(); idx++)
       {
              fout << "Session " << idx+1 << ":" << endl;
              for (i = 0; i < sessions[idx].size(); i++)
                     fout << sessions[idx][i].browses << " --- " <<
sessions[idx][i].weblog.utc_datetime << endl;
              fout << endl;
       }

       return 0;
}


//---------------------------------------------------------------------------

#endif
#ifndef __AISWUM_H__

       #define __AISWUM_H__

//---------------------------------------------------------------------------

#include "General.h"

//---------------------------------------------------------------------------

int AISWUM_Main (vector<vector<SessionEntry> > & sessions, vector<string> &
urls, vector<AntiBody> & antibodies)
{
//     vector<AntiBody> antibodies;
       map<string, int> url_to_index;
       vector<AntiBody> old_matures;
       int written_old_matures_num = 0;
       int i, j, k, session_idx = -1;
       ofstream resultslog ((default_input_file_name +
".OutputLog.out").c_str());
       ofstream resultantibodies ((default_input_file_name +
".AntibodyLog.out").c_str());
       ofstream resultantibodies2 ((default_input_file_name +
".FinalAntibodyLog.out").c_str());

#ifdef LOG_AIS_CELLS_IN_FILE
       ofstream flog ((default_input_file_name + ".ais_cells.data").c_str());
       ofstream flog1 ((default_input_file_name +
".ais_cells_not_csv.data").c_str());
       ofstream flog2 ((default_input_file_name +
".ais_cells_short.data").c_str());
```

```
	flog << "# Valid Session Data in \'" << default_input_file_name << "\'"
<< endl;
#endif

	for (i = 0; i < urls.size(); i++)
		url_to_index[urls[i]] = i;

#ifdef DO_LOAD_ANTIGEN
	ifstream findata
((default_input_file_name+".OrderedAntigens.output").c_str());
#endif

	// Inserting the Nb valid antibodies in a list, for further processes.
	while (antibodies.size() < NbInit)
	{
#ifndef DO_LOAD_ANTIGEN
		if (++session_idx >= sessions.size())
			break;

		cerr << "Processing " << session_idx << "th AntiBody" << endl;

		// Checking, ... Danger Theory!
		if (Validity(sessions[session_idx]) < ValidityThreshold)
			continue;
		if (sessions[session_idx].size() < MinSessionSize)
			continue;

		// Creating the antibody,
		AntiBody tmp ((int)urls.size());
		for (i = 0; i < sessions[session_idx].size(); i++)

	tmp.cell[url_to_index[sessions[session_idx][i].weblog.url]] = 1;
#else
		AntiBody tmp /*((int)urls.size())*/;
		string line;
		if (!getline(findata, line))
			break;
		stringstream ss (line);
		int t;

		while (ss >> t)
			tmp.cell.push_back(t);
#endif

#ifdef LOG_AIS_CELLS_IN_FILE
		for (i = 0; i < tmp.cell.size(); i++)
		{
			flog << (i > 0 ? ", " : "") << tmp.cell[i];
			flog1 << (i > 0 ? " " : "") << tmp.cell[i];
			//if (tmp.cell[i] > 0)
			//	flog2 << i << " ";
		}
		//flog2 << endl;
		flog1 << endl;
		flog << endl;
#endif

		antibodies.push_back (tmp);
	}

	//////////////right in file///////////////

	resultantibodies << "Time 0" << endl;
	for (k = 0; k < antibodies.size(); k++)
	{
		for (i = 0; i < urls.size(); i++)
			if (antibodies[k].cell[i] > 0)
				resultantibodies << i << " ";
```

```
		resultantibodies << endl;
	}

	//////// K-means clustering algorithm,//////
	vector<vector<int> > clusters;
	vector<AntiBody> centroids ((int)sqrt((double)NbInit));
	kmeans (antibodies, clusters, centroids, (int)urls.size(),
MaxKmeansIterationsAllowed, true);


	int time_since_world_created = 0, init_idx = session_idx;

	////////////////main loop/////////////////
	while (0 == 0)
	{
		double session_validity = 0.0;

#ifndef DO_LOAD_ANTIGEN
		if (++session_idx >= sessions.size())
			break;
		cerr << "Processing " << session_idx - init_idx << "th AntiGen"
<< endl;

		// Checking, ...
		if ((session_validity = Validity(sessions[session_idx])) <
ValidityThreshold)
			continue;
		if (sessions[session_idx].size() < MinSessionSize)
			continue;

		// Creating the antigen,
		AntiGen antigen ((int)urls.size());
		for (i = 0; i < sessions[session_idx].size(); i++)

	antigen.cell[url_to_index[sessions[session_idx][i].weblog.url]] = 1;
		antigen.index_in_sessions = session_idx;
#else
		AntiGen antigen /*((int)urls.size())*/;
		string line;
		if (!getline(findata, line))
			break;
		stringstream ss (line);
		int t;

		while (ss >> t)
			antigen.cell.push_back(t);
#endif


#ifdef LOG_AIS_CELLS_IN_FILE
		for (i = 0; i < antigen.cell.size(); i++){
			flog << (i > 0 ? ", " : "") << antigen.cell[i];
			flog1 << (i > 0 ? " " : "") << antigen.cell[i];
			if (antigen.cell[i] > 0)
				flog2 << i << " ";
		}
		flog2<< endl;
		flog1<< endl;
		flog << endl;
#endif
		time_since_world_created++;

		cerr << "Time of system is: " << time_since_world_created <<
endl;
		cerr << "Checking With Cluster Centroids..." << endl;
		// Finding the most activated cluster,
		double weight, max_weight = 0;
		int max_w_index = -1;
```

```
            double w_validity = 1.0;
            double dist = 0.0;

#ifdef VALIDITY_WEIGHTED_STIMULATION
                    w_validity = session_validity;
#endif
            for (i = 0; i < centroids.size(); i++)
            {
                    weight = Weight (centroids[i], antigen, sessions, urls);

//-------------------
// Updaitong Scale and Weight for Centroids.
                    dist = AntibodyAntigenDistance (centroids[i], antigen,
sessions, urls);
                    centroids[i].scale = w_validity * (centroids[i].scale *
centroids[i].weight + weight * SQR(dist)) / (2 * ( centroids[i].weight +
weight));

                    centroids[i].weight += weight;
//-------------------

                    if (max_weight < weight)
                            max_weight  = weight,
                            max_w_index = i;
                    if (max_weight == weight && (double)rand()/RAND_MAX <
ProbToAttachToAnotherCluster)
                            max_w_index = i;
            }

            bool outlier = true;
            for (i = 0; i < clusters[max_w_index].size(); i++)
            {
                    AntiBody & cur_antibody =
antibodies[clusters[max_w_index][i]];
                    weight = Weight (cur_antibody, antigen, sessions, urls);
                    if (weight < MinWeightTh)
                            continue;
                    outlier = false;
            }

            if (outlier)
            {
                    antibodies.push_back (AntiBody(antigen.cell,
time_since_world_created));
                    clusters[max_w_index].push_back((int)antibodies.size()-1);
                    UpdateCentroidofCluster(antibodies, clusters, centroids,
max_w_index, (int)urls.size());
            }
            else
            {
                    for (i = 0; i < clusters[max_w_index].size(); i++)
                    {
                            AntiBody & cur_antibody =
antibodies[clusters[max_w_index][i]];
                            weight = Weight (cur_antibody, antigen, sessions,
urls);

                            // Directed Mutation Peparation,
                            for (j = 0; j < urls.size(); j++)
                                    if (cur_antibody.cell[j] > 0 &&
antigen.cell[j] <= 0)
                                            cur_antibody.cell[j]++;
                                    else if (cur_antibody.cell[j] <= 0 &&
antigen.cell[j] > 0)
                                            cur_antibody.cell[j]--;

                            if (weight >= MinWeightTh)
                            {
```

```
                                cur_antibody.last_stimulation =
time_since_world_created;
                                cur_antibody.number_of_stimulations++;
                        }

#ifdef SUPPRESSION_COSTIMULATION
                        dist = AntibodyAntigenDistance (cur_antibody,
antigen, sessions, urls, false);
                        double w_sum = 0.0, wd_sum = 0.0;
                        for (j = 0; j < clusters[max_w_index].size(); j++)
                        {
                                if (i == j) continue;
                                double d_in, w_in;
                                w_in = Weight (cur_antibody,
antibodies[clusters[max_w_index][j]], d_in);
                                w_sum += w_in;
                                wd_sum += w_in * SQR(d_in);
                        }
                        double dissim = CalcClusterDissimilarity
(antibodies, clusters[max_w_index]);
                        double Beta  =  double((14 - dissim)/20);
                        cur_antibody.stimulation_level = w_validity *
(cur_antibody.weight + weight)/ cur_antibody.scale + (Alpha * w_sum /
cur_antibody.scale) - (Beta * w_sum / cur_antibody.scale);
                        cur_antibody.scale = w_validity * (
cur_antibody.scale * (cur_antibody.weight + weight) * SQR(dist) + Alpha *
wd_sum - Beta * wd_sum) / (2 * ( cur_antibody.weight + weight + Alpha * w_sum
- Beta * w_sum));
#else
                        dist = AntibodyAntigenDistance (cur_antibody,
antigen, sessions, urls);
                        cur_antibody.stimulation_level = w_validity *
(cur_antibody.weight + weight)/ cur_antibody.scale;
                        cur_antibody.scale = w_validity *
(cur_antibody.scale * (cur_antibody.weight + weight) * SQR(dist)) / (2 *
(cur_antibody.weight + weight));
#endif
                        cur_antibody.weight += weight;
                }
        }

        //Calculating average stimulation of system.
        double st_sum = 0.0, st_avg;
        for (j = 0; j < antibodies.size(); j++)
                st_sum += antibodies[j].stimulation_level;
        st_avg = st_sum / antibodies.size();
        cerr << "Average stimulation level of system before cloning is: "
<< st_avg << endl;

        // Starting Clone phases.
        cerr << "Clonning..." << endl;
        for (j = 0; j < antibodies.size(); j++)
        {
                if (!(time_since_world_created -
antibodies[j].last_stimulation <= StagnationTime &&
                        antibodies[j].number_of_stimulations >=
MinStimulationNumber))
                        continue;

                int NClone = int (CloningFactor *
(antibodies[j].stimulation_level) / st_avg);
                if (NClone > 0)
                        antibodies[j].number_of_stimulations = 0; //
last_stimulation = ?

                cerr << "Stimulation level of cloned Antibody: " <<
antibodies[j].stimulation_level << endl;
```

```
                    cerr << NClone << " new samples added via cloning." <<
endl;

                    for (k = 0; k < NClone; k++)
                    {
                                AntiBody new_ab (antibodies[j].cell,
time_since_world_created);
                                antibodies.push_back (new_ab);
                    }

                    st_avg = (st_avg * antibodies.size() - NClone) /
(antibodies.size());
                    cerr << "Average stimulation level after cloning: " <<
st_avg << endl;
             }

             cerr << "Population size: " << antibodies.size () << endl;
             cerr << "Average stimulation level after cloning: " << st_avg <<
endl;

             // Preparing to Remove Excess Antibodies.
             if (antibodies.size() > NbMax)
             {

                    // Checking to transfer old mature antibodies, to
sencondary memory.
                    int excess = (int)antibodies.size()-NbMax;
                    int numofmatures = 0;
                    for (j = 0; j < antibodies.size()&& excess > 0; j++)
                    {
                           if (time_since_world_created -
antibodies[j].last_stimulation > StagnationTime &&
                                antibodies[j].stimulation_level > st_avg)
                           {
                                old_matures.push_back (antibodies[j]);
                                antibodies.erase (antibodies.begin()+j);
                                excess--;
                                j--;
                                numofmatures++;
                           }
                    }
                    cerr << "Number of matured erased antibodies " <<
numofmatures << endl;
                    cerr << "Population size: " << antibodies.size () << endl;

             }
             if (antibodies.size() > NbMax)
             {
                    for (k = 0; k < antibodies.size(); k++)
                    {
                           if (time_since_world_created - antibodies[k].birth
<= MinAge && antibodies[k].stimulation_level <= st_avg)
                           {
                                antibodies[k].dummy =
antibodies[k].stimulation_level;
                                antibodies[k].stimulation_level = st_avg;
                           }
                           else antibodies[k].dummy = -1.0;
                    }

                    sort (antibodies.begin(), antibodies.end());
#ifdef STIMULATION_REMOVE_CRITERION
                    reverse (antibodies.begin(), antibodies.end());
#endif
                    // We should redo the stimulation_levels of !mature
antibodies to 0.
                    for (k = 0; k < antibodies.size(); k++)
                           if (antibodies[k].dummy == -1.0)
```

```
                                continue;
                        else
                        {
                                antibodies[k].stimulation_level =
antibodies[k].dummy ;
                                antibodies[k].dummy = -1.0;
                        }

                // Removing Excess Antibodies.
                antibodies.erase (antibodies.begin() + min(NbMax,
(int)antibodies.size()), antibodies.end());
        }
        st_sum = 0.0;
        for (j = 0; j < antibodies.size(); j++)
                st_sum += antibodies[j].stimulation_level;
        st_avg = st_sum / antibodies.size();
        cerr << "Population size: " << antibodies.size () << endl;
        cerr << "Average stimulation level after removing excess
antibodies " << st_avg << endl;

        // This is to update 'clusters' vector, removing invalid
pointers.
        kmeans (antibodies, clusters, centroids, (int)urls.size(), 1);

        // write antibodies to file for evaluation
        resultantibodies << "Time: " << time_since_world_created << endl;
        for (j = 0; j < clusters.size(); j++)
        {
                for (k = 0; k < clusters[j].size(); k++){
                        for (i = 0; i < urls.size(); i++)
                                if (antibodies[clusters[j][k]].cell[i] > 0)
                                        resultantibodies << i << " ";
                        resultantibodies << endl;
                }

        }

        if (time_since_world_created % RehabilitationPeriod == 0)
        {
                cerr << "Periodical Check..." << endl;


                // Mutation
                st_sum = 0.0;
                for (j = 0; j < antibodies.size(); j++)
                        st_sum += antibodies[j].stimulation_level;
                st_avg = st_sum / antibodies.size();
                cerr << "Average stimulation level of system before
mutation is: " << st_avg << endl;

                int antibodiesnum = (int)antibodies.size();
                for (j = 0; j < antibodiesnum; j++)
                {
                        if ( antibodies[j].last_stimulation >=
(time_since_world_created - RehabilitationPeriod)  &&
                                antibodies[j].stimulation_level <= st_avg)
                        {
                                int cntmutate = 0; int cntone = 0;
                                for (k = 0; k < urls.size(); k++)
                                {
                                        if (
(abs((double)antibodies[j].cell[k])/RehabilitationPeriod) >= MutationFactor )
cntmutate++;
                                        if ( antibodies[j].cell[k] >= 1 )
cntone++;
                                }
                                cerr << (double)cntmutate/urls.size() <<
endl;
```

```
                                if ( (double)cntmutate/urls.size() >=
ContentmentTh ) // decide to mutate antibody or not
                                {
                                        AntiBody new_ab (antibodies[j].cell,
time_since_world_created);

                                        for (k = 0; k < urls.size(); k++)
                                        {
                                                if (
(double)new_ab.cell[k]/RehabilitationPeriod > MutationFactor)
                                                {
                                                        double p =
(double)rand()/RAND_MAX;
                                                        if (p <= (new_ab.cell[k]-
1) * BaseMutationProbability)
                                                                new_ab.cell[k] =
0, cntone--, antibodies[j].cell[k] = 1;
                                                        else new_ab.cell[k] = 1,
antibodies[j].cell[k] = 1;
                                                }
                                                else if (
(double)new_ab.cell[k]/RehabilitationPeriod < -1.0*MutationFactor)
                                                {
                                                        double p =
(double)rand()/RAND_MAX;
                                                        if ( p <= -1 *
new_ab.cell[k] * BaseMutationProbability)
                                                                new_ab.cell[k] =
1, cntone++, antibodies[j].cell[k] = 0;
                                                        else new_ab.cell[k] = 0,
antibodies[j].cell[k] = 0;
                                                }
                                                else if ( new_ab.cell[k] > 1 )
                                                        new_ab.cell[k] = 1,
antibodies[j].cell[k] = 1;
                                                else if ( new_ab.cell[k] < 0 )
                                                        new_ab.cell[k] = 0,
antibodies[j].cell[k] = 0;

                                        }// for k
                                        if (cntone >
ValidNumOnesAfterMutation)
                                                antibodies.push_back (new_ab);

                                }
                        }
                        else
                        {
                                for (k = 0; k < urls.size(); k++)
                                        if ( antibodies[j].cell[k] > 1 )
                                                antibodies[j].cell[k] = 1;
                                        else if ( antibodies[j].cell[k] < 0 )
                                                antibodies[j].cell[k] = 0;
                        }
                }
                cerr << antibodies.size()-antibodiesnum << " samples added
via mutation." << endl;

                // Reclustering
                kmeans (antibodies, clusters, centroids,
(int)urls.size());
                cerr << "Done..." << endl;
                resultslog << "-------------------------------------------
--------------------" << endl;
                resultslog << "Time: " << time_since_world_created <<
endl;
                for (j = 0; j < clusters.size(); j++)
                {
```

```
                            resultslog << "Cluster " << j << "(size=" <<
clusters[j].size() << "): " << endl << "Centroid: " << endl << centroids[j] <<
endl;
                            for (k = 0; k < clusters[j].size(); k++){
                                    resultslog << k << "th Antibody:" << endl <<
antibodies[clusters[j][k]] << endl;
                            }

                    }
            }
    }
kmeans (antibodies, clusters, centroids, (int)urls.size(),
MaxKmeansIterationsAllowed);
    resultslog << "-------------------------------------------------------
------" << endl;
    resultslog << "-------------------------------------------------------
------" << endl;
    resultslog << "Final results in  time: " << time_since_world_created <<
endl;
    for (j = 0; j < clusters.size(); j++)
    {
            resultslog << "Cluster " << j << "(size=" << clusters[j].size()
<< "): " << endl << "Centroid: " << endl << centroids[j] << endl;
            for (k = 0; k < clusters[j].size(); k++)
                    resultslog << k << "th Antibody:" << endl <<
antibodies[clusters[j][k]] << endl;
    }

    for (i = 0; i < centroids.size(); i++)
    {
            for (j = 0; j < urls.size(); j++)
                    if (centroids[i].cell[j] >= 1)
                            resultantibodies2 << j << " ";
            resultantibodies2 << endl;
    }
    centroids.clear();
    centroids.resize(15);
    if ( old_matures.size() > 30 )
    {
            kmeans (old_matures, clusters, centroids, (int)urls.size(),
MaxKmeansIterationsAllowed, true);
            for (i = 0; i < centroids.size(); i++)
            {
                    for (j = 0; j < urls.size(); j++)
                            if (centroids[i].cell[j] >= 1)
                                    resultantibodies2 << j << " ";
                    resultantibodies2 << endl;
            }
    }
    else
            for (i = 0; i < old_matures.size(); i++)
            {
                    for (j = 0; j < urls.size(); j++)
                            if (old_matures[i].cell[j] >= 1)
                                    resultantibodies2 << j << " ";
                    resultantibodies2 << endl;
            }

    return 0;
}

//-------------------------------------------------------------------------

#endif
```

Abstract:

WITH the ever expanding Web and the information published on it, effective tools for managing such data and presenting information to users based on their needs and taste are becoming necessary. Web Usage Mining is one of the applications of data mining to discover interesting usage patterns from Web data and its goal is to leverage data collected as a result of user interactions with the Web in order to learn user models and to use these models for different applications that aim to ease the use of Web. There are a number of attributes of the web, firstly the content of the web is forever changing and so are the expectations of the user, secondly because of the convenient addition and deletion of data on the Web, it is full of irrelevant noise, thirdly Web is huge. Any Web mining system that is to retrieve an acceptable set of results must adapt to the condition of the Web. Artificial Immune System is a new, biologically inspired, paradigm for learning. There are a number of motivations for using the immune system as inspiration for Web usage mining which include recognition, diversity, memory, self regulation, and learning and the most important feature of AIS is its dynamic nature that is so similar to the dynamic nature of Web.

In this thesis, an AIS-based model for Web usage mining is designed, implemented and evaluated. Processes like artificial immune network and danger theory are used to extract frequent itemsets from usage data. Some of the novelties in the proposed algorithm are the directed mutation that is designed to avoid the random nature of mutation that make the system nondeterministic, presenting a new model for learning new antigen instead of using the hypermutation which highly cost and defining weight for each item in antigen and calculating the system parameters based on weighted items. Experimental results confirm the pre-mentioned prediction that AIS fit well to the WUM application and algorithm is able to produce good results.

Iran University of Science and Technology
Department of Computer Engineering

Artificial Immune based Approach to Assocation Rule Mining

By:
Bentol Hoda Helmi

A Thesis Submitted in Partial Fulfillment of the Requirement for the Degree of Master of Engineering in Artificial Intelligence-Computer Engineering

Supervisor:
Dr. Adel T. Rahmani

January 2008