

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**  
**KHOA KHOA HỌC MÁY TÍNH**

**NGUYỄN VĨNH KHA**

**NGHIÊN CỨU KỸ THUẬT KẾT HỢP LỘC CỘNG  
TÁC VÀ CÁC PHƯƠNG PHÁP SUY DIỄN DỰA TRÊN  
CƠ SỞ TRI THỨC CHO KHUYẾN NGHỊ TÀI NGUYÊN  
HỌC TẬP TRONG HỌC TRỰC TUYẾN**

**LUẬN VĂN THẠC SĨ**  
**NGÀNH KHOA HỌC MÁY TÍNH**

**MÃ SỐ: 60.48.01.01**

**TP. HỒ CHÍ MINH, 2015**

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**  
**KHOA KHOA HỌC MÁY TÍNH**



**NGUYỄN VĨNH KHA**

**NGHIÊN CỨU KỸ THUẬT KẾT HỢP LỘC CỘNG**  
**TÁC VÀ CÁC PHƯƠNG PHÁP SUY DIỄN DỰA TRÊN**  
**CƠ SỞ TRI THỨC CHO KHUYẾN NGHỊ TÀI NGUYÊN**  
**HỌC TẬP TRONG HỌC TRỰC TUYẾN**

**LUẬN VĂN THẠC SĨ**  
**NGÀNH KHOA HỌC MÁY TÍNH**

**MÃ SỐ: 60.48.01.01**

**NGƯỜI HƯỚNG DẪN KHOA HỌC:**  
**PGS. TS. VŨ THANH NGUYỄN**

**TP. HỒ CHÍ MINH, 2015**

## **DANH SÁCH HỘI ĐỒNG BẢO VỆ KHÓA LUẬN**

Hội đồng chấm khóa luận tốt nghiệp, thành lập theo Quyết định số ..... ngày ..... của Hiệu trưởng Trường Đại học Công nghệ Thông tin.

1. PGS.TS. Đỗ Văn Nhơn - ĐH CNTT – Chủ tịch.
2. TS. Nguyễn Minh Sơn - ĐH CNTT – Thư ký.
3. PGS.TS. Đặng Trần Khánh - ĐH Bách Khoa – Phản biện 1.
4. TS. Nguyễn Đình Thuận - ĐH CNTT – Phản biện 2.
5. PGS.TS. Trần Công Hùng - Học viện BCVT – Ủy viên.

## LỜI CẢM ƠN

Trước tiên, em xin gửi lời cảm ơn chân thành đến thầy PGS. TS. Vũ Thanh Nguyên. Trong suốt quá trình làm luận văn, thầy đã dành nhiều công sức giúp đỡ và hướng dẫn em tận tình để em có thể hoàn tất đề tài này một cách thuận lợi nhất.

Bên cạnh đó, em xin cảm ơn cô ThS. Đỗ Thị Minh Phụng trong thời gian em thực hiện luận văn đã truyền đạt và đóng góp những ý kiến hết sức quý báu, giúp em hoàn thành tốt đề tài.

Em cũng xin gửi lời cảm ơn đến các thầy cô tại trường ĐH Công nghệ Thông tin, ĐHQG HCM đã tạo mọi điều kiện cho em có thể học tập và hoàn tất luận văn này.

Em xin chân thành cảm ơn.

TP. Hồ Chí Minh, tháng 10 năm 2015

Nguyễn Vĩnh Kha

## **LỜI CAM ĐOAN**

Tôi xin cam đoan những nội dung trong luận văn là kết quả nghiên cứu thực sự của cá nhân dưới sự hướng dẫn của thầy PGS. TS. Vũ Thanh Nguyên. Tôi xin hoàn toàn chịu trách nhiệm về luận văn của mình.

Học viên

Nguyễn Vĩnh Kha

**NHẬN XÉT**  
**(Của giảng viên phản biện)**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

## MỤC LỤC

DANH SÁCH HỘI ĐỒNG BẢO VỆ KHÓA LUẬN .....	3
LỜI CẢM ƠN .....	4
LỜI CAM ĐOAN .....	5
NHẬN XÉT (Của giảng viên phản biện).....	6
MỤC LỤC .....	7
DANH MỤC CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT .....	10
DANH MỤC CÁC BẢNG .....	12
DANH MỤC HÌNH VẼ, ĐỒ THỊ .....	13
TÓM TẮT LUẬN VĂN .....	14
Chương 1. TỔNG QUAN.....	15
1.1. Đặt vấn đề .....	15
1.2. Bài toán khuyến nghị .....	16
1.3. Bài toán khuyến nghị tài nguyên học tập.....	17
1.4. Đóng góp chính của luận văn.....	18
Chương 2. CÁC NGHIÊN CỨU LIÊN QUAN .....	20
2.1. Phương pháp khuyến nghị lọc cộng tác .....	20
2.2. Phương pháp kết hợp lọc cộng tác (Collaborative Filtering - CF) với các phương pháp suy luận dựa trên cơ sở tri thức (Knowledge- based Reasoning - KBR) .....	22
Chương 3. CÁC PHƯƠNG PHÁP NỀN TẢNG.....	23
3.1. Tổng quan về các hệ khuyến nghị thuộc nhóm lọc cộng tác (Collaborative Filtering - CF).....	23
3.2. Matrix Factorization.....	24

3.3.	Non-negative Matrix Factorization.....	26
3.4.	Các hệ cơ sở tri thức .....	28
3.5.	Phương pháp Rule-based reasoning.....	29
3.5.1.	Thuật toán xây dựng cây quyết định ID3.....	32
3.5.2.	Các phiên bản cập nhật của ID3 - C4.5 và C5.0 .....	38
3.6.	Phương pháp Case-based Reasoning .....	40
3.6.1.	Cách thức biểu diễn case.....	43
3.6.2.	Phương pháp rút trích các case tương đồng .....	44
3.6.3.	Tái sử dụng case.....	46
3.6.4.	Điều chỉnh và lưu trữ case.....	47
3.6.5.	Ưu và nhược điểm của suy luận theo tình huống.....	48
Chương 4.	THUẬT TOÁN KHUYẾN NGHỊ LAI KẾT HỢP CƠ SỞ TRI THỨC VÀ LỘC CỘNG TÁC.....	50
4.1.	Mô hình tổng quan .....	50
4.2.	Rút trích đặc trưng tường minh của người dùng.....	52
4.3.	Kết hợp đặc trưng tường minh và không tường minh - vector đặc trưng người dùng .....	54
4.4.	Áp dụng thuật toán phân cụm để xác định các cụm người dùng ...	55
4.5.	Rút trích các quy tắc suy luận ra cụm người dùng dựa trên Rule- based Reasoning .....	55
4.6.	Suy luận cụm người dùng dựa trên Case-based Reasoning.....	55
4.7.	Kết hợp lọc cộng tác và các phương pháp suy diễn dựa trên tri thức .....	56
Chương 5.	DỮ LIỆU KIỂM THỬ VÀ TIỀN XỬ LÝ DỮ LIỆU .....	59
5.1.	Bộ dữ liệu kiểm thử.....	59



5.2.	Thống kê đặc tính của các tập dữ liệu kiểm thử và lựa chọn các trường dữ liệu phù hợp .....	62
5.3.	Các vấn đề về tiền xử lý dữ liệu – Lựa chọn biểu cách biểu diễn một item.....	64
Chương 6. THỰC NGHIỆM VÀ PHÂN TÍCH KẾT QUẢ.....		67
6.1.	Lựa chọn dữ liệu và mô hình thực nghiệm .....	67
6.2.	Kết quả thực nghiệm và đánh giá.....	68
Chương 7. KẾT LUẬN .....		70
7.1.	Kết quả đạt được .....	70
7.2.	Hướng phát triển .....	70
TÀI LIỆU THAM KHẢO.....		71

## DANH MỤC CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT

ALS	Alternating Least Squares
CBF	Content-based Filtering
CBR	Case-based Reasoning
CBRR	Case-based Reasoning Recommendation
CF	Collaborative Filtering
CRN	Case Retrieval Net
HR	Hybrid Recommendation
IG	Information Gain
KB	Knowledge-base
KBR	Knowledge-Based Reasoning
KB	Knowledge-based System
KC	Knowledge Component
MF	Matrix Factorization
NMF	Non-negative matrix factorization
RBR	Rule-based Reasoning
RBR	Rule-based Reasoning
RBRR	Rule-based Reasoning Recommendation
RCBRR	Rule- and Case- Based Reasoning Recommendation
RS	Recommendation System
SGD	Stochastic Gradient Descent
SVD-based	Singular Value Decomposition based

TF	Tensor Factorization
WHRCBR-MF	Weighted Hybrid system of Rule-Case Based Reasoning and Matrix Factorization

## DANH MỤC CÁC BẢNG

<b>Bảng 3.1.</b>	Dữ liệu ví dụ một bài toán có thể giải quyết bằng phương pháp xây dựng cây quyết định
<b>Bảng 3.2.</b>	Ưu và nhược điểm của thuật toán ID3
<b>Bảng 3.3.</b>	Các cải tiến của C4.5
<b>Bảng 3.4.</b>	Các cải tiến của C5.0 so với C4.5
<b>Bảng 3.5.</b>	Mô tả một case cụ thể trong bài toán quản lý quỹ tín dụng
<b>Bảng 5.1.</b>	Mô tả các trường dữ liệu của bộ dữ liệu Cognitive Tutor
<b>Bảng 5.2.</b>	Thống kê các đặc tính của hai tập dữ liệu “algebra 2008 2009” và “bridge to algebra 2008 2009”
<b>Bảng 5.3.</b>	Bảng phân tích hai giải pháp xác định item trong bộ dữ liệu Cognitive Tutor
<b>Bảng 5.4.</b>	Thống kê tình trạng các tập dữ liệu khi sử dụng KC để xác định item (step)
<b>Bảng 6.1.</b>	Thiết lập hệ số cho thực nghiệm hệ WHRCBR-MF trên hai tập dữ liệu “algebra 2008 2009” và “bridge to algebra 2008 2009”
<b>Bảng 6.2.</b>	So sánh kết quả dự đoán của WHRCBR-MF với các thuật toán dự thi KDD Cup 2010 và các thuật toán cơ sở

## DANH MỤC HÌNH VẼ, ĐỒ THỊ

- Hình 1.1.** Tương quan giữa khuyến nghị sản phẩm và khuyến nghị tài nguyên học tập
- Hình 3.1.** Ma trận rating/mark
- Hình 3.2.** Nonnegative Matrix Factorization trong khuyến nghị
- Hình 3.3.** Mô tả hệ cơ sở tri thức
- Hình 3.4.** Cây quyết định cho bài toán dự đoán số lượng người chơi golf dựa trên điều kiện thời tiết
- Hình 3.5.** Cây quyết định (phiên bản trực quan hơn) cho bài toán dự đoán số lượng người chơi golf dựa trên điều kiện thời tiết
- Hình 3.6.** Chu trình của phương pháp suy luận theo tình huống
- Hình 3.7.** Thể liên tục các mô hình hiệu chỉnh
- Hình 4.1.** Mô hình tổng quan (thành phần 1) hệ khuyến nghị lai WHRCBR-MF
- Hình 4.2.** Mô hình tổng quan (thành phần 2 và 3) hệ khuyến nghị lai WHRCBR-MF
- Hình 4.3.** Vec-tơ đặc trưng  $ufv_u$
- Hình 5.1.** Một mẫu file dữ liệu \*\_train.txt

## TÓM TẮT LUẬN VĂN

Luận văn này tập trung vào bài toán dự đoán thành tích học tập của người học dựa trên các kỹ thuật lọc cộng tác và suy luận dựa trên cơ sở tri thức. Kết quả của kỹ thuật các kỹ thuật này có thể được sử dụng để đánh giá mức độ tiếp thu của người học cũng như đưa ra những khuyến nghị học tập mang tính định hướng, phù hợp với năng lực người học.

Luận văn đã đề xuất một hệ khuyến nghị lai, kết hợp việc sử dụng các phương pháp suy luận dựa trên cơ sở tri thức với phương pháp lọc cộng tác nhằm tăng cường độ chính xác của quá trình dự đoán trình độ của người học.

Song song đó, một phương pháp rút trích đặc trưng người học cũng được tìm hiểu và triển khai. Các cách tiếp cận khác nhau đã được sử dụng để đồng thời tìm ra các vec-tơ đặc trưng tường minh và không tường minh (ẩn) nhằm tối ưu hóa việc đặc tả người dùng. Các kết quả thực nghiệm cho thấy hệ khuyến nghị đề xuất mang lại sự cải thiện về độ chính xác so với các kỹ thuật truyền thống.

## **Chương 1. TỔNG QUAN**

### **1.1. Đặt vấn đề**

Hệ khuyến nghị - Recommendation System (RS) là những công cụ phần mềm và kỹ thuật đưa ra lời khuyên giúp người dùng lựa chọn các sản phẩm (đối tượng) phù hợp [2]. Các kỹ thuật giải quyết khuyến nghị có thể được chia thành ba nhóm chính: lọc cộng tác (Collaborative Filtering - CF), lọc nội dung (content-based filtering - CBF) và phương pháp lai (hybrid). Trong đó CF là kỹ thuật được sử dụng phổ biến và rộng rãi nhất trong các hệ khuyến nghị hiện nay, ý tưởng chính của kỹ thuật này là sử dụng những đánh giá của người dùng trên một số sản phẩm để đưa ra dự đoán và khuyến nghị những sản phẩm mới. Kỹ thuật CF thể hiện rõ tính cộng đồng và khuyến nghị dựa trên xu hướng của cộng đồng mà không quan tâm đến bản chất người dùng cũng như đặc tính sản phẩm.

Mặt khác, các kỹ thuật suy luận dựa trên tri thức (Knowledge-Based Reasoning KBR) có thể đưa ra khuyến nghị dựa trên những suy luận tri thức về những đặc tính của sản phẩm đáp ứng nhu cầu, sở thích và hữu ích với người dùng hoặc dựa trên việc mô hình hóa các trường hợp, đưa ra các tập luật để đưa ra các suy luận khi các trường hợp mới xuất hiện.

Quá trình phát triển của khuyến nghị gắn liền với sự ra đời và phát triển của các hệ thương mại (e-commerce), khuyến nghị ra đời trong các hệ thống này với mục tiêu mang đến cho người dùng sự lựa chọn các sản phẩm phù hợp với nhu cầu một cách nhanh chóng cũng như mang lại lợi ích to lớn cho các hệ thống bán hàng.

Qua quá trình phát triển, ngoài lĩnh vực thương mại điện tử, khuyến nghị còn được áp dụng rộng rãi sang nhiều lĩnh vực khác, cụ thể là lĩnh vực học tập trực tuyến (e-learning). Khái niệm “sản phẩm” trong các hệ thương mại trở thành các khái niệm “bài học, bài tập, tài nguyên học tập, tài liệu tham khảo” trong lĩnh vực học tập trực tuyến, nơi mà người học có khả năng chủ động học tập mọi lúc, mọi nơi. Trong e-learning, các phần mềm dựa trên kết nối mạng được phát triển để tạo điều kiện thuận lợi cho người dùng học ở không gian và thời gian bất kỳ. Không

nằm ngoài sự tăng trưởng nhanh chóng của sản phẩm trong các lĩnh vực e-commerce, các tài nguyên trong e-learning cũng thể hiện xu hướng tăng trưởng về số lượng cũng như chủng loại, điều này mang đến sự khó khăn cho người học trong việc lựa chọn tài nguyên phù hợp. Vì thế, để các ứng dụng e-learning trở thành một ứng dụng thông minh, hệ khuyến nghị cần được triển khai tích hợp với các ứng dụng này.

## 1.2. Bài toán khuyến nghị

Về tổng quan, bài toán khuyến nghị được coi là bài toán ước lượng trước hạng (rating) của các sản phẩm (phim, món ăn, các sản phẩm gia dụng ...) chưa được người dùng xem xét. Việc ước lượng này thường dựa trên những đánh giá đã có của chính người dùng đó hoặc những người dùng khác. Những sản phẩm có hạng cao nhất sẽ được dùng để khuyến nghị.

Bài toán khuyến nghị được mô tả như sau:

Gọi  $U$  là tập tất cả người dùng (users);  $I$  là tập tất cả các sản phẩm (items) có thể tư vấn. Tập  $I$  có thể rất lớn, từ hàng trăm ngàn (sách, cd...) đến hàng triệu (như website).

Hàm  $r(u, i)$  đo độ phù hợp (hay hạng) của sản phẩm  $i$  với user  $u$ :

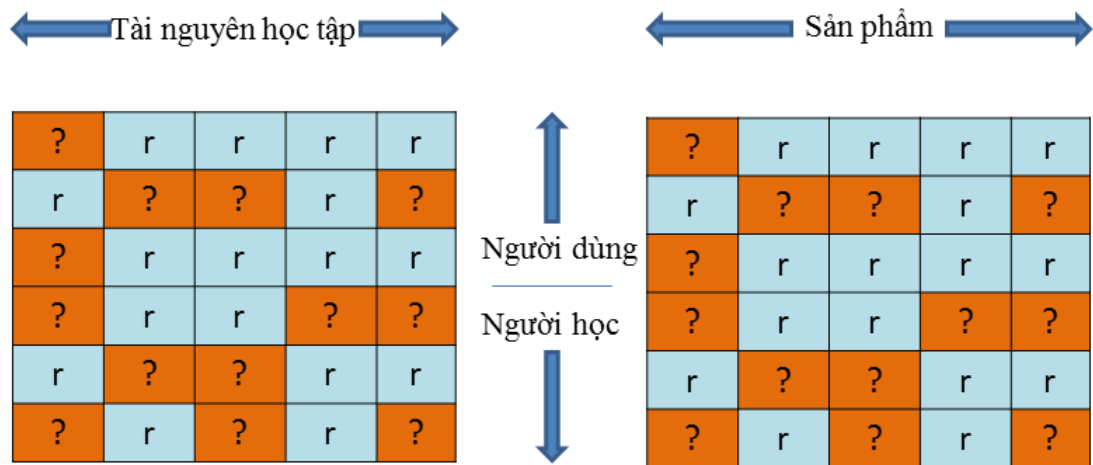
$$r: U \times I \rightarrow R$$

Trong đó  $R$  là tập các đánh giá (rating) được sắp thứ tự. Với mỗi người dùng  $u \in U$ , cần tìm sản phẩm  $i \in I$  sao cho hàm  $r(u, i)$  đạt giá trị lớn nhất.



### 1.3. Bài toán khuyến nghị tài nguyên học tập

Về cơ bản, bài toán khuyến nghị tài nguyên học tập cũng tương tự như các trường hợp khuyến nghị sản phẩm. Trong đó tập sản phẩm  $I$  được hiểu là tập các tài nguyên học tập (bài học, bài tập, bài kiểm tra...) trong khi tập  $R$  có thể là tập kết quả học tập của học viên hoặc đánh giá của học viên trên một tài nguyên học tập cụ thể. Hình 1.1 mô tả sự tương tự của mô hình khuyến nghị sản phẩm và mô hình khuyến nghị tài nguyên học tập.



**Hình 1.1.** Tương quan giữa khuyến nghị sản phẩm và khuyến nghị tài nguyên học tập

Từ đây, tài liệu sẽ sử dụng thuật ngữ user dùng chung cho khái niệm người dùng/người học, thuật ngữ item cho khái niệm sản phẩm/tài nguyên học tập. Vấn đề dự đoán *điểm đánh giá/điểm số đạt được* của khi một user tương tác với một item được biểu diễn lại như sau:

Hàm  $v(u, i)$  đo độ phù hợp của item  $i$  với user  $u$ :

$$v: U \times I \rightarrow V$$

Trong đó  $V$  là tập các *điểm đánh giá/điểm số đạt được* được sắp thứ tự. Với mỗi người dùng  $u \in U$ , cần tìm tài nguyên học tập  $i \in I$  sao cho hàm  $v(u, i)$  đạt giá trị lớn nhất.

#### 1.4. Đóng góp chính của luận văn

Hiện nay, các phương pháp khuyến nghị có ba hướng cơ bản chính:

- Khuyến nghị dựa trên lọc nội dung (Content-based Recommendation): Sử dụng các thông tin sẵn có của user và item để đưa ra các gợi ý phù hợp nhất với các đặc trưng của user.
- Khuyến nghị lọc cộng tác (Collaborative Filtering Recommendation): Thu thập và đánh giá hành vi user, các hoạt động cũng như sở thích của user từ đó đưa ra các dự đoán dựa trên sự tương tự (trong hành vi, sở thích...) của user đang xét với các user khác.
- Khuyến nghị lai (Hybrid Recommendation - HR): Sử dụng kết hợp hai phương pháp cơ bản để tận dụng tối đa điểm mạnh của từng phương pháp đồng thời khắc phục các nhược điểm vốn có khi sử dụng riêng rẽ.

Để tăng độ chính xác của khuyến nghị, việc sử dụng tối đa các đặc trưng, thông tin của user và item là rất quan trọng. Các hệ cơ sở tri thức cho phép tận dụng tốt thông tin thu thập được trong bộ dữ liệu, đồng thời trích xuất ra các quy luật chung rất hữu hiệu trong việc giải quyết các vấn đề mới xuất hiện. Với mục đích này, nhiều nghiên cứu gần đây được tiến hành nhằm kết hợp các phương pháp suy luận dựa trên cơ sở tri thức vào quá trình khuyến nghị và mang lại kết quả rất khả quan.

Với tư tưởng tận dụng tối đa các thông tin thu thập được về sở thích, xu hướng, năng lực học tập của user, luận văn đề xuất một phương pháp rút trích sau đó kết hợp các đặc trưng tường minh và không tường minh của user vào một vec-tơ đặc trưng duy nhất. Quan trọng hơn, luận văn đề xuất một hệ khuyến nghị lai kết hợp giữa lọc cộng tác với các phương pháp suy luận dựa trên cơ sở tri thức. Các bước chính của hệ bao gồm:

- Rút trích các đặc trưng không tường minh của user dựa trên thuật toán Matrix Factorization (MF).
- Rút trích các đặc trưng tường minh của user dựa trên phương pháp mới được đề xuất. Kết hợp các đặc trưng tường minh và không tường minh vào một vec-tơ đặc trưng duy nhất.

- Sử dụng thuật toán gom cụm để bước đầu phân loại user vào các cụm khác nhau.
- Sử dụng thuật toán Rule-based Reasoning (RBR), rút trích các quy tắc xác định cụm của người dùng.
- Áp dụng thuật toán hướng Case-based Reasoning (CBR) để xác định cụm người dùng.
- Kết hợp các kết quả thu được ở hai bước trên và kết quả dự đoán của thuật toán Matrix Factorization, đưa dự đoán cuối cùng về điểm số của người dùng đối với một item cụ thể.

## **Chương 2. CÁC NGHIÊN CỨU LIÊN QUAN**

Trong chương này, học viên sẽ tiến hành khảo sát các nghiên cứu gần đây liên quan đến các bài toán khuyến nghị nói chung và khuyến nghị tài nguyên học tập nói riêng.

### **2.1. Phương pháp khuyến nghị lọc cộng tác**

Hiện có nhiều phương pháp khuyến nghị lọc cộng tác được áp dụng với những điểm mạnh và yếu riêng. Một số phương pháp tiêu biểu như sau:

Hướng tiếp cận xây dựng mô hình các nhân tố ẩn trong lọc cộng tác với ý tưởng chính là xác định các đặc trưng ẩn không được đề cập đến trong bộ dữ liệu tiêu biểu như thuật toán pLSA [1], thuật toán dựa trên mạng nơ-ron [2], thuật toán Latent Dirichlet Allocation [3], và các mô hình phân giải ma trận rating Matrix Factorization (MF) [4](còn được gọi là các mô hình dựa trên phân giải trị đơn SVD-based - Singular Value Decomposition based). Phiên bản bậc cao của MF - Tensor Factorization - cho phép tăng số lượng các yếu tố đánh giá từ 2 (user và rating) lên 3 hoặc hơn nữa cho phép tích hợp các yếu tố ngữ cảnh một cách tường minh.

Tài liệu [5] đề xuất rằng người dùng có kiến thức tốt hơn (ví dụ, những người dùng đạt kết quả tốt hơn trong những bài kiểm tra khác nhau) sẽ có trọng số cao hơn khi tính toán khuyến nghị so với những người dùng khác. Sử dụng memory-based CF kết hợp với những công thức mới để mở rộng việc thu thập và xử lý thông tin liên quan đến điểm số thu được từ mỗi người dùng trong những bài kiểm tra khác nhau. Ý tưởng cơ bản của việc đánh trọng số khuyến nghị là không những sử dụng độ tương đồng truyền thống giữa những người dùng với nhau mà còn xem xét việc những người dùng có điểm số tốt hơn sẽ có trọng số cao hơn những người dùng khác.

Công trình [6] sử dụng thuật toán neighborhood-based CF để khuyến nghị tài liệu cho người học dựa trên đánh giá của người học trên các tài liệu học tập. Công trình cung cấp công cụ tạo các câu hỏi và xác định cấp độ cho từng câu hỏi (mới bắt đầu, trung bình, chuyên gia), công cụ lựa chọn một cách ngẫu nhiên các câu hỏi để tạo

thành bài kiểm tra cho mỗi bài học. Đường dẫn học tập (learning path) cho mỗi người học dựa vào điểm số bài kiểm tra của người học.

Trong tài liệu [7] nhóm tác giả sử dụng user-based CF là thuật toán chính. Tỷ lệ giữa số giờ học thực tế của người học so với tổng số giờ học của khóa học được ghi nhận như là điểm đánh giá ngầm định (implicit rating score) và được quy đổi thành thang đánh giá tương ứng từ 1-5. Dữ liệu được thu thập từ khai phá dữ liệu lịch sử, mối quan tâm của người dùng và tần suất yêu cầu trang web (nhật ký truy cập trang web, thông tin môn học, và điểm đánh giá môn học). Thông tin hữu ích được trích xuất từ nguồn dữ liệu thu thập được và lưu vào kho dữ liệu. Hệ khuyến nghị được thiết kế độc lập với hệ thống e-learning. Khi có yêu cầu khuyến nghị, hệ khuyến nghị phân tích yêu cầu của người dùng và trả về kết quả cho hệ thống e-learning.

Công trình [8] tiến hành thí nghiệm sử dụng người học giả lập (artificial learners) với hai kỹ thuật khuyến nghị model-based và hybrid CF. Thiết lập mô phỏng: tạo ra 500 người học giả lập và 50 bài báo làm tài nguyên học tập. Hệ thống sẽ khuyến nghị 15 bài báo cho mỗi người học. Mỗi người học sẽ đánh giá những bài báo này dựa vào thuộc tính của họ. Sau đó, sẽ tạo thêm 100 người học khác gọi là “target learners”. Sau đó sẽ áp dụng hai kỹ thuật khuyến nghị cho “target learners” từ đó so sánh hai kỹ thuật với nhau.

Tài liệu [9] đề cập đến việc sử dụng thuật toán Matrix Factorization để dự đoán năng lực của sinh viên.

Tài liệu [10] đề xuất hệ khuyến nghị dựa trên các đặc trưng ẩn của tài nguyên. Việc xác định các thuộc tính ngầm định được dựa trên lịch sử đánh giá (historical rating) trong ma trận user x material. Mô hình dự đoán được xây dựng dựa trên những thuộc tính đã được quan sát hoặc thuộc tính ngầm định để cải thiện độ chính xác trong dự đoán. Thuật toán di truyền được sử dụng để tìm ra mối quan hệ giữa toàn bộ đánh giá và vector trọng số của thuộc tính ngầm định cho mỗi người học. Sau khi xác định được trọng số của các thuộc tính, độ tương đồng giữa những người học sẽ được tính dựa trên các thuộc tính đó. Sau đó giá trị đánh giá dự đoán của

người học cho tài nguyên sẽ được tính dựa vào đánh giá của các hàng xóm của người học.

Tài liệu [11] đưa ra các giải pháp khác nhau cho phép tích hợp yếu tố ngữ cảnh thời gian vào việc dự đoán đánh giá người dùng trên các sản phẩm trong hệ thống. Tài liệu đưa ra khái niệm dự đoán mối liên kết thời gian (Temporal Link Prediction) giữa các lượt đánh giá. Các thuật toán MF, Tensor Factorization (TF) và Katz kết hợp với ngữ cảnh thời gian đã được cài đặt và đánh giá hiệu năng. Kết quả cho thấy việc xét đến ngữ cảnh trong bài toán cho ra những kết quả tốt rất khả quan.

## **2.2. Phương pháp kết hợp lọc cộng tác (Collaborative Filtering - CF) với các phương pháp suy luận dựa trên cơ sở tri thức (Knowledge-based Reasoning - KBR)**

Trong tài liệu [12] nhóm tác giả đã đề cập đến cách tích hợp cơ sở tri thức vào hệ khuyến nghị. Xây dựng các phương pháp suy luận dựa trên Rule-based Reasoning (RBR) và Case-based Reasoning (CBR). Đồng thời một giải pháp suy luận kết hợp cả hai phương pháp trên đã được đề xuất.

Công trình [13] tiến hành cài đặt các thuật toán lai giữa KB và CF, cũng như lai giữa các phương pháp CF khác nhau từ đó đưa ra các đánh giá thực nghiệm cho thuật toán này.

Tài liệu [14] sử dụng kết hợp kỹ thuật Constraint-based với CF để xây dựng một hệ khuyến nghị lai. Nhóm tác giả đã đúc kết ra các vấn đề cần quan tâm khi sử dụng kết hợp hai kỹ thuật với nhau.

### **Chương 3. CÁC PHƯƠNG PHÁP NỀN TẢNG**

Phần này của tài liệu tập trung vào các thuật toán nền tảng của hệ khuyến nghị lai, các phần chính bao gồm tổng quan về các thuật toán khuyến nghị thuộc hướng lọc cộng tác, các phương pháp giải bài toán Matrix Factorization, Nonnegative Matrix Factorization, tổng quan về hệ cơ sở tri thức, các phương pháp suy diễn rule-based và case-based reasoning.

#### **3.1. Tổng quan về các hệ khuyến nghị thuộc nhóm lọc cộng tác (Collaborative Filtering - CF)**

Các thuật toán lọc cộng tác (Collaborative Filtering - CF) được phân thành ba nhóm chính:

- Nhóm Memory-based: hướng tiếp cận của nhóm này dựa trên các đánh giá, thành tích của người dùng để xác định sự tương đồng của user và item. Đây là hướng tiếp cận chính được sử dụng trong giai đoạn đầu, khi mà đa số các hệ khuyến nghị được dùng cho các hệ thống e-commerce. Hướng tiếp cận này có điểm mạnh là có khả năng giải thích được bản chất của kết quả khuyến nghị, dễ xây dựng và sử dụng; đồng thời, khả năng tiếp nhận dữ liệu mới cho phép dễ cập nhật dữ liệu cho thuật toán. Tuy nhiên, một trong những điểm yếu lớn nhất của các thuật toán hướng memory-based là hiệu năng giảm dần khi dữ liệu thừa, đây là một trở ngại lớn khi đa phần các tập dữ liệu trên trong các hệ thống web đều là dữ liệu thừa; mặc dù các thuật toán thuộc nhóm này cho phép cập nhật thêm dữ liệu về user mới khá dễ dàng, nhưng vẫn phải phụ thuộc và cấu trúc dữ liệu được lưu trữ.
- Nhóm Model-based: hướng tiếp cận này nhắm đến việc xác định các mô hình (pattern) của dữ liệu thông qua việc sử dụng các thuật toán khai phá dữ liệu, máy học trên tập dữ liệu huấn luyện. Các thuật toán tiêu biểu thuộc nhóm này bao gồm mạng Bayes, các mô hình phân cụm (Clustering Models), các mô hình ngữ nghĩa ẩn (Latent Semantic Models) như

Singular Value Decomposition (SVD), phân tích ngữ nghĩa ẩn dựa hướng xác suất (Probabilistic Latent Semantic Analysis), mô hình nhân tố đa bội (Multiple Multiplicative Factor), phân phối Dirichlet ẩn (Latent Dirichlet Allocation) và các mô hình dựa trên quy trình quyết định Markov (Markov decision process based models). Điểm mạnh của các thuật toán thuộc nhóm này bao gồm khả năng xử lý tốt hơn với các tập dữ liệu thưa, khả năng làm việc tốt với các tập dữ liệu lớn; bên cạnh đó, các thuật toán thuộc nhóm này có khả năng phân tích tốt các nhân tố căn bản ẩn trong tập dữ liệu. Điểm yếu của nhóm thuật toán này bao gồm: tài nguyên sử dụng để xây dựng mô hình cao, do vậy có sự đánh đổi về hiệu năng dự đoán và chi phí cài đặt; trong khi một số thuật toán có khả năng làm thất thoát thông tin do sử dụng các mô hình biến đổi thu gọn (reduction models), một số khác lại khó giải thích được kết quả dự đoán.

- Nhóm thuật toán CF lai (Hybrid CF): Sử dụng kết hợp các thuật toán thuộc hai hướng tiếp cận memory-based và model-based để khắc phục các nhược điểm khi áp dụng từng hướng riêng lẻ. Nhóm thuật toán này cho khả năng dự đoán tốt hơn; quan trọng hơn, các vấn đề như mất mát thông tin và xử lý dữ liệu thưa cũng được xử lý tốt hơn. Tuy nhiên nhược điểm chính của hướng này lại là mô hình phức tạp và chi phí cài đặt cao.

Phần sau sẽ đi sâu tìm hiểu thuật toán Matrix Factorization, một thuật toán thuộc nhánh mô hình ngữ nghĩa ẩn (Latent Semantic Models). Đây là thuật toán được sử dụng khá nhiều trong những năm gần đây, do sự đơn giản trong cài đặt và khả năng xử lý tốt dữ liệu thưa.

### **3.2. Matrix Factorization**

Matrix Factorization hay Matrix Decomposition là phương pháp khuyến nghị thuộc nhóm Latent Semantic Models. Về mặt toán học, Matrix Factorization là phương pháp phân tích ma trận ban đầu  $V^{m \times n}$  thành tích các ma trận thành phần. Có



nhiều phương pháp để triển khai việc phân tích  $V$ , chủ yếu chia làm ba nhóm chính [18]

- Nhóm phương pháp giải các hệ phương trình tuyến tính (Decompositions related to solving systems of linear equations): Các phương pháp thuộc nhóm này bao gồm LU decomposition, LU reduction, Block LU decomposition, Rank factorization, Cholesky decomposition, QR decomposition, RRQR factorization và Interpolative decomposition.
- Nhóm phương pháp dựa trên trị riêng và các khái niệm liên quan (Decompositions based on eigenvalues and related concepts): Bao gồm các phương pháp Eigendecomposition, Jordan decomposition, Schur decomposition, QZ decomposition, Takagi's factorization và Singular value decomposition (SVD)
- Nhóm các phương pháp còn lại: Polar decomposition, Algebraic polar decomposition, Sinkhorn normal form và Sectoral decomposition.

Tuy về mặt toán học, các phương pháp trên đều có thể giải quyết việc phân tích ma trận ban đầu nhưng chúng lại đòi hỏi ma trận cần phân tích phải là ma trận đầy đủ (các vị trí trong ma trận phải được lấp đầy). Điều này là một trở ngại trong việc áp dụng Matrix Factorization vào khuyến nghị bởi các tập dữ liệu khuyến nghị thường là các tập thưa. Để giải quyết vấn đề trên, phương pháp có tên Non-negative Matrix Factorization (NMF) hay Non-negative Matrix Approximation [19][20] đã được đề xuất.

Về tổng quan, NMF phân tích ma trận ban đầu  $V$  thành tích hai ma trận thành phần  $W$  và  $H^T$

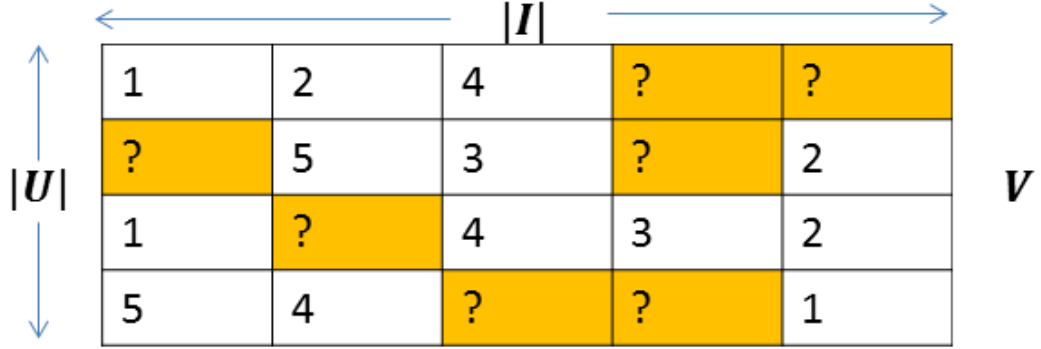
$$V \approx WH^T \quad (3.1)$$

Với  $V \in \mathbb{R}^{m \times n}$ ,  $W \in \mathbb{R}^{m \times p}$  và  $H \in \mathbb{R}^{n \times p}$ .

Phần tiếp theo sẽ đi chi tiết hơn về NMF trong ngữ cảnh của một hệ khuyến nghị.

### 3.3. Non-negative Matrix Factorization

Trong ngữ cảnh khuyến nghị, giả sử danh sách các rating/mark của tập user  $U$  trên tập item  $I$  được trình bày dưới dạng một ma trận  $V^{|U| \times |I|}$  có dạng như sau:



			$ I $		
	1	2	4	?	?
	?	5	3	?	2
	1	?	4	3	2
$ U $	5	4	?	?	1
					$V$

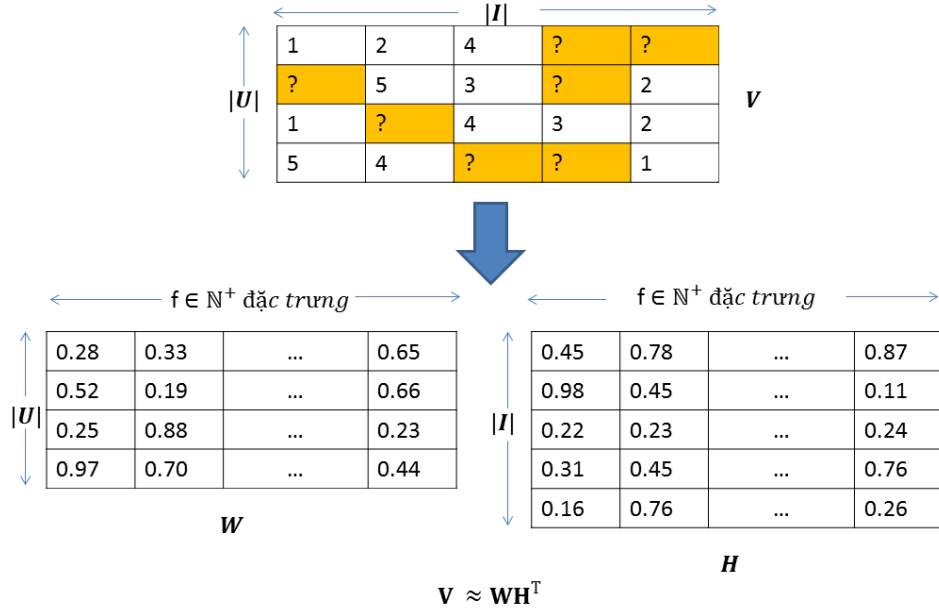
**Hình 3.1.** Ma trận rating/mark

Phương pháp NMF dựa trên ý tưởng phân tích ma trận  $V^{|U| \times |I|}$  thành tích hai ma trận thành phần  $W^{|U| \times f}$  và  $H^{|I| \times f}$  theo công thức (3.1)

$$V \approx WH^T$$

Với các phần tử trong  $V$ ,  $W$  và  $H$  là các giá trị không âm. Khi đó một user  $u$  hay item  $i$  bất kỳ được mô tả bằng vector đặc trưng ẩn (latent factor vector)  $p_u$  hay  $q_i \in \mathbb{R}^f$ . Như vậy,  $W^{|U| \times f}$  là ma trận với mỗi dòng  $p_u$  là vector đặc trưng cho user, trong khi  $H^{|I| \times f}$  là ma trận với mỗi dòng  $q_i$  là vector đặc trưng cho user  $i$ . Việc dự đoán rating/mark khi  $u$  tương tác với  $i$  được xác định nhờ công thức

$$\hat{v}_{ui} = p_u q_i^T \quad (3.2)$$



**Hình 3.2.** *Nonnegative Matrix Factorization trong khuyến nghị*

Vấn đề chính của NMF là tính toán ra tập các vector  $p_u$  và  $q_i$  sao cho giá trị của các  $\hat{v}_{ui}$  càng gần với các  $v_{ui}$  tương ứng. Vấn đề trên tương ứng với bài toán tối thiểu hóa sau:

$$\min_{p^*, q^*} \sum_{(u,i) \in \mathbb{R}^f} (v_{ui} - p_u q_i^T)^2 + \lambda (\|p_u\|^2 + \|q_i\|^2) \quad (3.3)$$

Có hai phương pháp chính để giải bài toán tối thiểu (3.3) là phương pháp leo đồi - Stochastic Gradient Descent (SGD) và Alternating Least Squares (ALS) [4]. Tài liệu sẽ tập trung tìm hiểu phương pháp SGD được đề cập trong nghiên cứu [21].

Để tối thiểu hóa (3.3), nghiên cứu [21] tính toán sai số dự đoán  $e_{ui}$  cho mỗi trường hợp với công thức:

$$e_{ui} = v_{ui} - p_u q_i^T \quad (3.4)$$

Tiếp đó là quá trình tính toán lại  $p_u$  và  $q_i$  như sau:

$$p_u \leftarrow p_u + \gamma \cdot (e_{ui} q_i - \lambda \cdot p_u) \quad (3.5)$$

$$q_i \leftarrow q_i + \gamma \cdot (e_{ui} p_u - \lambda \cdot q_i) \quad (3.6)$$

Quá trình tính toán sai số  $e_{ui}$  và tính toán lại  $p_u$ ,  $q_i$  được lặp lại cho đến khi một ngưỡng được thỏa hoặc dựa trên số lần lặp *iter* cho trước. Trong các công thức (3.5) và (3.6):

- $\gamma$  (tốc độ học – learning rate): quyết định mức độ thay đổi (nhiều hay ít) của  $p_u$  và  $q_i$  trong mỗi bước lặp.
- $\lambda$  (trọng số chuẩn – regularization term hay regularization weight): kiểm soát độ lớn của vector nhân tố tiềm ẩn sao cho ma trận  $W$  và  $H$  sẽ mang lại độ xấp xỉ tốt cho ma trận  $V$  mà không làm cho các hạng tử trong  $V$  phải quá lớn

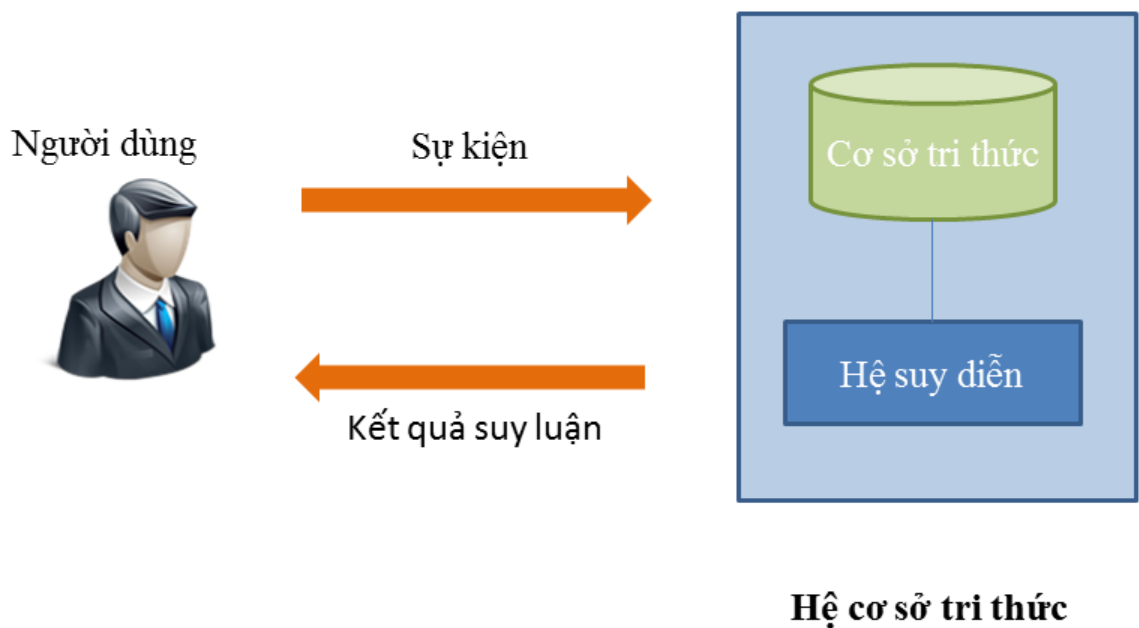
Việc xác định các tham số  $\gamma$ ,  $\lambda$  đóng một vai trò quan trọng và thường được quyết định dựa trên thực nghiệm.

Thực nghiệm cho thấy hiệu năng của NMF là khá tốt so với tính đơn giản và dễ cài đặt của nó. Gần đây, một số cải tiến của NMF đã được tìm hiểu, càng nâng cao hơn tính hiệu quả của nó. Phương pháp bias Matrix Factorization [4] cho kết quả dự đoán tốt hơn nhờ xem xét đến các đặc điểm riêng của từng user và item. Trong khi đó, tích hợp yếu tố thời gian vào NMF là một cải tiến đáng kể do có khả năng biểu diễn quá trình thay đổi thói quen, sở thích hay năng lực học tập của user. Các nghiên cứu về Tensor Factorization, dạng tổng quát và phức tạp hơn của MF, cũng đã được triển khai và cho thấy sự cải thiện rõ rệt về khả năng áp dụng trong dự đoán [11].

Để đơn giản và thống nhất, từ đây tài liệu sẽ sử dụng thuật ngữ Matrix Factorization (MF) để đề cập đến Non-negative Matrix Factorization, do trong ngữ cảnh khuyến nghị hai khái niệm này có thể sử dụng thay thế nhau.

### 3.4. Các hệ cơ sở tri thức

Hệ cơ sở tri thức (Knowledge-based System - KBS) là hệ thống có khả năng suy luận để giải quyết các vấn đề từ đơn giản đến phức tạp dựa trên việc sử dụng cơ sở tri thức (Knowledge-base - KB) có sẵn. Về cơ bản, hệ cơ sở tri thức được cấu thành từ hai thành phần chính là hệ suy diễn (Inference Engine) và cơ sở tri thức. Cơ sở tri thức có thể được xây dựng dựa trên các dữ liệu, thông tin có hoặc không có cấu trúc. Hình 3.3 mô tả một cách tổng quan và đơn giản cách thức một hệ cơ sở tri thức tương tác với người dùng.



**Hình 3.3.** Mô tả hệ cơ sở tri thức

Cơ sở tri thức có nhiều dạng khác nhau, phụ thuộc vào mô hình biểu diễn tri thức. Các mô hình phổ biến bao gồm mô hình *đối tượng – thuộc tính – giá trị*, mô hình *thuộc tính – luật dẫn*, *mạng ngữ nghĩa* và *frame*. Nhiệm vụ chính của cơ sở tri thức là cung cấp thông tin, tri thức nền tảng cho hệ suy diễn làm việc. Do vậy một cơ sở tri thức nhiều dữ kiện và được mô tả hợp lý đóng vai trò quan trọng trong hệ.

Trong khi đó, nhiệm vụ của hệ suy diễn là tiến hành các bước suy luận dựa trên cơ sở tri thức có sẵn và dữ liệu sự kiện do người dùng cung cấp để đưa ra các kết quả mang tính định lượng (ví dụ: bao nhiêu % một người thuộc nhóm nào, khả năng xảy ra mưa của một ngày..). Có nhiều phương pháp để xây dựng hệ suy diễn, trong các phần sau, tài liệu sẽ đi sâu tìm hiểu hai phương pháp thường được dùng hiện nay là phương pháp suy luận dựa trên luật (Rule-based Reasoning - RBR) và phương pháp suy luận dựa trên trường hợp (Case-based Reasoning - CBR).

### 3.5. Phương pháp Rule-based reasoning

Một trong những phương pháp xây dựng hệ suy diễn đầu tiên là phương pháp suy diễn dựa trên luật (Rule-based reasoning - RBR). Phương pháp CBR là nền tảng để xây dựng các hệ cơ sở tri thức cũng như các hệ chuyên gia (Expert System) vào thời

điểm khởi đầu của quá trình phát triển các hệ thống dựa trên cơ sở tri thức. Các hệ dựa trên RBR bắt đầu được phát triển vào những năm bảy mươi của thế kỷ trước, trong đó cơ sở tri thức hay tri thức chuyên gia được biểu diễn thông qua các luật có dạng

*IF <điều kiện> THEN <ứng xử>*

Ví dụ, việc xác định một người là đủ tuổi công dân hay chưa có thể được thực hiện thông qua luật có dạng:

*IF <tuổi  $\geq 18$ > THEN <đủ tuổi công dân>*

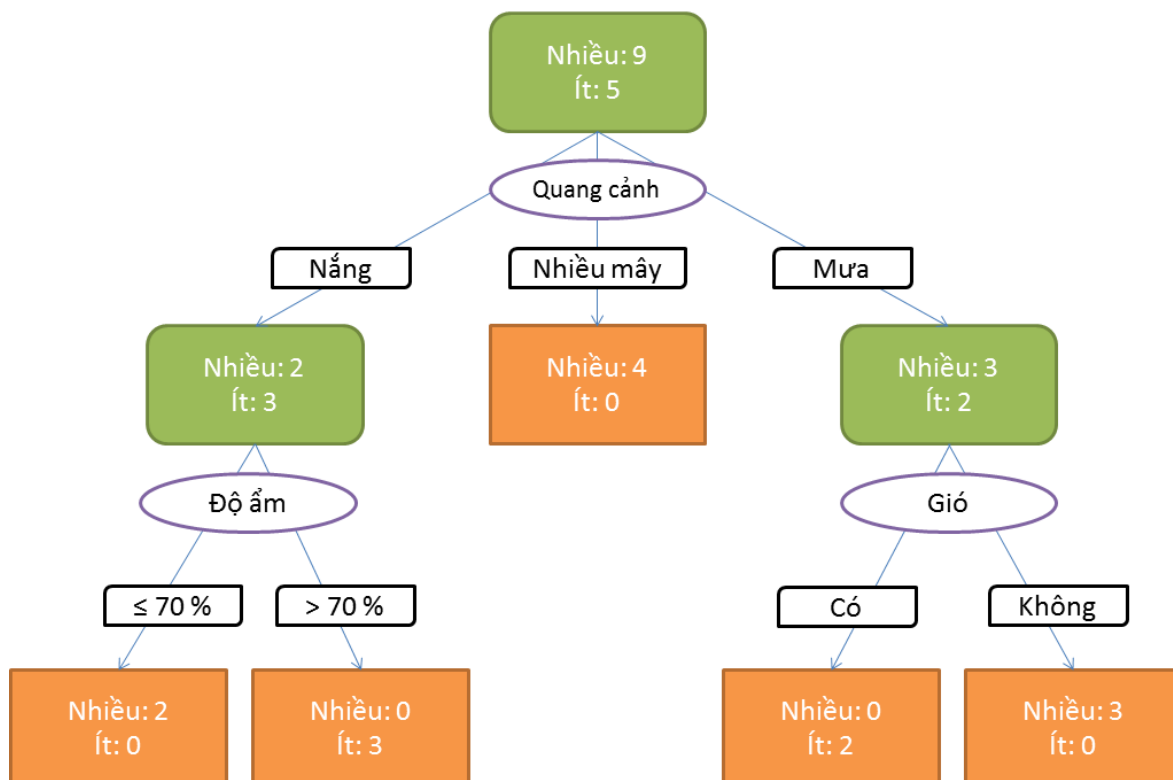
Vấn đề cơ bản trong các hệ dựa trên RBR (Rule-based System) là xác định được tập các luật biểu diễn tri thức chuyên gia của hệ. Một trong những phương pháp tiếp cận truyền thống là xây dựng một cây quyết định (Decision Tree). Cây quyết định là mô hình mang tính dự đoán cho phép đưa ra những kết luận về các giá trị mục tiêu của một đối tượng thông qua việc tính toán trên các tính chất đo lường được của đối tượng đó. Dưới đây là ví dụ về cây quyết định để dự đoán xem có ít hay nhiều người chơi golf dựa trên thông tin về thời tiết.

THAM SỐ ĐỘC LẬP				
Quang cảnh	Nhiệt độ (độ F)	Độ ẩm(%)	Gió	Số người chơi golf
Nắng	85	85	Không	Ít
Nắng	80	90	Có	Ít
Nhiều mây	83	78	Không	Nhiều
Mưa	70	96	Không	Nhiều
Mưa	68	80	Không	Nhiều
Mưa	65	70	Có	Ít
Nhiều mây	64	65	Có	Nhiều
Nắng	72	95	Không	Ít
Nắng	69	70	Không	Nhiều
Mưa	75	80	Không	Nhiều
Nắng	75	70	Có	Nhiều

Nhiều mây	72	90	Có	Nhiều
Nhiều mây	81	75	Không	Nhiều
Mưa	71	80	Có	Ít

**Bảng 3.1.** Dữ liệu ví dụ một bài toán có thể giải quyết bằng phương pháp xây dựng cây quyết định

Với bảng trên ta có cây quyết định như sau:



**Hình 3.4.** Cây quyết định cho bài toán dự đoán số lượng người chơi golf dựa trên điều kiện thời tiết

Dựa trên cây quyết định trên ta có thể đưa ra một số suy luận như sau:

- Suy luận thứ nhất: nếu trời nhiều mây, người ta luôn luôn chơi golf. Và có một số người ham mê đến mức chơi golf cả khi trời mưa.
- Tiếp theo, ta lại chia nhóm trời nắng thành hai nhóm con. Ta thấy rằng khách hàng không muốn chơi golf nếu độ ẩm lên quá 70%.
- Cuối cùng, ta chia nhóm trời mưa thành hai và thấy rằng khách hàng sẽ không chơi golf nếu trời nhiều gió.

Có thể thấy đây là lời giải ngắn gọn cho bài toán mô tả bởi cây phân loại. Hiện nay, có nhiều thuật toán để xây dựng cây phân loại, tiêu biểu như:

- ID3 (Iterative Dichotomiser 3): là thuật toán được sử dụng nhiều hiện nay do tính trực quan và dễ cài đặt. ID3 được giới thiệu bởi Ross Quinlan vào năm 1986[22].
- C4.5 và C5: là các phiên bản tốt hơn của ID3 cũng do Ross Quinlan giới thiệu.
- CART (Classification And Regression Tree)
- CHAID (CHi-squared Automatic Interaction Detector): Cho phép thực hiện các phép phân chia đa bậc khi tính toán cây phân loại[23].
- MARS: tạo ra cây phân loại có khả năng xử lý dữ liệu số tốt hơn.
- Conditional Inference Trees: Thuật toán dựa trên thống kê sử dụng các kiểm định không tham số như điều kiện phân chia. Hướng tiếp cận này tạo ra khả năng lựa chọn khách quan và không đòi hỏi bước xén tỉa như các thuật toán trước[24][25].

Các phần sau sẽ đi sâu tìm hiểu thuật toán ID3 và các phiên bản sau này của nó (C4.5 và C5), làm cơ sở cho hệ khuyến nghị đề xuất.

### **3.5.1. Thuật toán xây dựng cây quyết định ID3**

Thuật toán ID3 có đầu vào và đầu ra được mô tả như sau:

- Đầu vào: Một tập hợp các ví dụ. Mỗi ví dụ bao gồm các thuộc tính mô tả một tình huống, hay một đối tượng nào đó, và một giá trị phân loại của nó.
- Đầu ra: Cây quyết định có khả năng phân loại đúng đắn các ví dụ trong tập dữ liệu huấn luyện, và hy vọng là phân loại đúng cho cả các ví dụ chưa gặp trong tương lai.

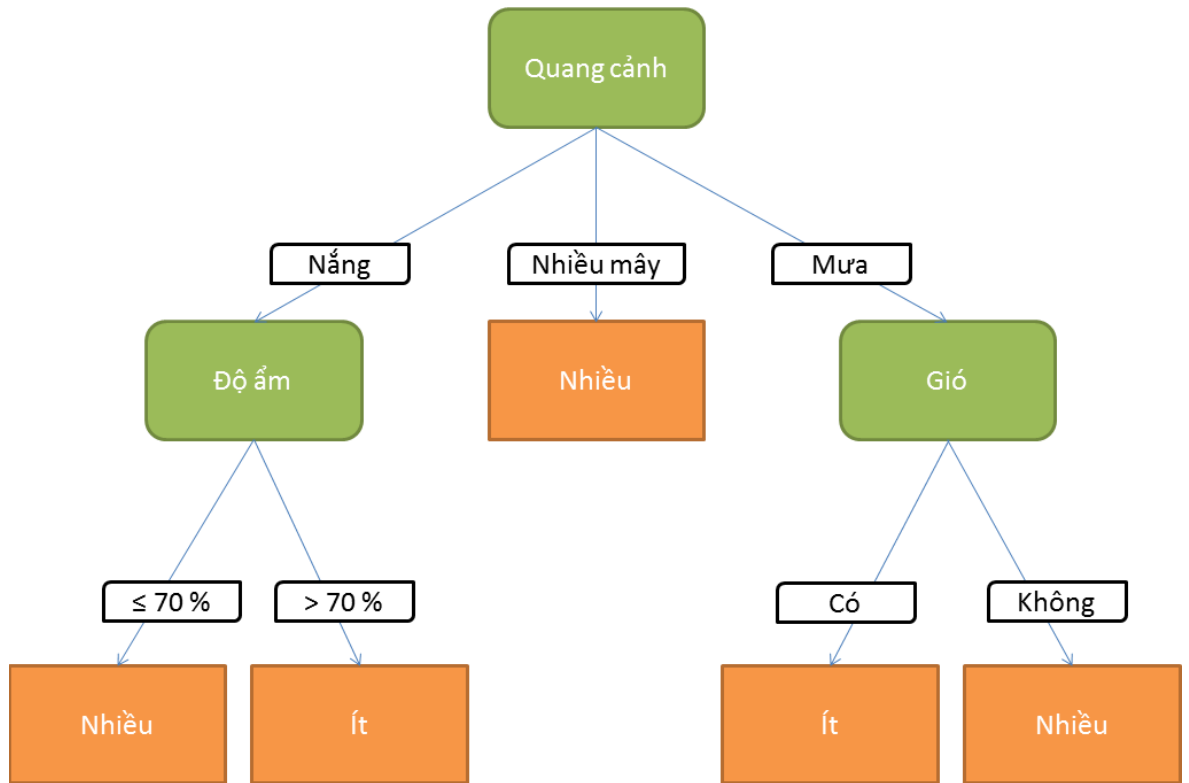
Thuật toán ID3 bắt đầu với ban đầu là các tập  $S$  như là nút gốc. Trên mỗi lần lặp của thuật toán, nó lặp qua tất cả các thuộc tính chưa được sử dụng của tập  $S$  và tính toán độ hỗn loạn Entropy  $H(S)$  (hoặc độ lợi thông tin (Information Gain)  $IG(A)$ ) của thuộc tính đó.



Sau đó chọn các thuộc tính có giá trị Entropy nhỏ nhất (hoặc Information Gain lớn nhất). Tập S sau đó được chia bởi thuộc tính được lựa chọn (ví dụ độ tuổi  $<50$ ,  $50 \leq \text{tuổi} < 100$ ,  $\text{tuổi} = 100$ ) để tạo ra các tập con của dữ liệu. Thuật toán tiếp tục đệ quy trên mỗi tập con, chỉ xem xét các thuộc tính chưa được chọn ở bước trước. Đệ quy trên tập con có thể dừng lại khi bắt gặp một trong những trường hợp sau

- Mỗi phần tử trong tập con thuộc về cùng một lớp (+ hoặc -), sau đó nút biến thành một lá và được dán nhãn với lớp tương ứng.
- Không có nhiều hơn nữa các thuộc tính chưa được chọn, nhưng các mẫu vẫn không thuộc về cùng một lớp (một số được + và một số -), khi đó nút biến thành một lá và dán nhãn với lớp phổ biến nhất của các mẫu trong tập con.
- Không có mẫu trong tập con, điều này sẽ xảy ra khi không có mẫu nào trong tập cha mà phù hợp với một giá trị cụ thể của thuộc tính lựa chọn, ví dụ nếu không có mẫu nào với  $\text{tuổi} \geq 100$ . Sau đó, một lá được tạo ra, và được dán nhãn với lớp phổ biến nhất của các mẫu trong tập cha.

Trong suốt thuật toán, cây quyết định được tạo lập bởi các nút không kết thúc - (non-terminal node) đại diện cho thuộc tính dùng để phân chia dữ liệu và các nút kết thúc (nút lá) biểu diễn nhãn lớp của tập con cuối cùng của nhánh đang xét. Dưới đây là một phiên bản cây quyết định trực quan hơn của bài toán xét xem có nhiều hay ít người chơi golf.



**Hình 3.5.** Cây quyết định (phiên bản trực quan hơn) cho bài toán dự đoán số lượng người chơi golf dựa trên điều kiện thời tiết

Thuật toán ID3 có thể được tóm gọn qua các bước như sau:

- Bước 1: Tính Entropy hoặc Information Gain của từng thuộc tính được sử dụng trong tập S ban đầu.
- Bước 2: Chia tập S thành các tập con, sử dụng thuộc tính có Entropy nhỏ nhất (hoặc Information Gain lớn nhất) làm tiêu chuẩn phân chia.
- Bước 3: Tạo một nút mới của cây đại diện cho thuộc tính phân chia được chọn.
- Bước 4: Lập lại các bước trên trên các tập con, sử dụng các thuộc tính chưa được xét.

Mã giả của thuật toán được trình bày như dưới đây

ID3(Tập mẫu - Examples, Thuộc tính mục tiêu cần xác định - Target\_Attribute, Tập thuộc tính - Attributes)

Tạo nút gốc Root cho cây

If "tất cả mẫu là positive(+)"

Return "cây có một nút duy nhất là Root với nhãn label = +".

If "tất cả mẫu là negative(-)"

Return "cây có một nút duy nhất là Root với nhãn label = -".

If "số lượng các thuộc tính dự đoán (các thuộc tính để phân chia) là rỗng"

Return "cây có một nút duy nhất là Root với nhãn label = giá trị gặp nhiều nhất của thuộc tính mục tiêu (Target\_Attribute) trong tập mẫu".

Otherwise

Begin

A  $\leftarrow$  Thuộc tính có khả năng phân loại tốt nhất đối với tập mẫu  
Thuộc tính cây quyết định cho nút gốc Root = A (lấy A làm giá trị cho nút gốc Root)

Với mỗi giá trị có thể xảy ra,  $v_i$ , của A,

Thêm một nhánh mới bên dưới nút gốc Root,

Tạo tập mẫu Examples( $v_i$ ) là tập mẫu con chứa tất cả các mẫu có thuộc tính A mang giá trị là  $v_i$ .

If "tập mẫu con Examples( $v_i$ ) rỗng"

Bên dưới nhánh thêm một nút lá (leaf) với nhãn label = giá trị mục tiêu gặp nhiều nhất trong tập mẫu

Else

Bên dưới nhánh con thêm một cây con  
ID3(Examples( $v_i$ ), Target\_Attribute, Attributes – {A})

End

Return "Nút gốc Root"

Dưới đây là công thức tính độ hỗn loạn Entropy dùng trong thuật toán ID3.

$$H(S) = -\sum_{x \in X} p(x) \log_2 p(x) \quad (3.7)$$

Trong đó:

- $S$ : tập dữ liệu (tập dữ liệu con) cần tính độ hỗn loạn Entropy (tập  $S$  thay đổi theo từng vòng lặp của thuật toán ID3)
- $X$ : tập các lớp  $x$  phân chia  $S$ .
- $p(x)$  : tỉ lệ giữa số lượng các mẫu thuộc lớp  $x$  với tổng lượng các mẫu thuộc tập  $S$

Các trường hợp đặc biệt:

- Nếu tất cả các mẫu thành viên trong tập  $S$  đều thuộc cùng một lớp thì  $Entropy(S) = 0$ .
- Nếu trong tập  $S$  có số mẫu phân bố đều nhau vào các lớp thì  $Entropy(S) = 1$ .
- Các trường hợp còn lại:  $0 < Entropy(S) < 1$ .

Trong thuật toán ID3, entropy được tính cho từng thuộc tính chưa được xét. Thuộc tính với entropy nhỏ nhất sẽ được chọn để phân chia  $S$  trong bước lặp hiện tại. Entropy càng cao, khả năng cây phân loại có thể được cải thiện càng cao, do vậy các thuộc tính với entropy nhỏ được ưu tiên chọn lựa trước.

Công thức tính độ lợi thông tin IG có dạng:

$$IG(A, G) = H(S) - \sum_{t \in T} p(t)H(t) \quad (3.8)$$

Trong đó:

- $A$ : thuộc tính bất kỳ trong tập các thuộc tính.
- $H(S)$ : Entropy của tập  $S$  (hiện thời) đang xét.
- $T$ : tập các tập con  $t$  được tạo ra khi chia  $S$  bằng thuộc tính  $A$  với  $S = \bigcup_{t \in T} t$
- $p(t)$  : tỉ lệ giữa số lượng các mẫu thuộc tập con  $t$  với tổng lượng các mẫu thuộc tập  $S$ .
- $H(t)$ : Entropy của tập con  $t$ .

Độ lợi thông tin IG có thể được tính toán (thay cho Entropy) đối với từng thuộc tính chưa được xét. Thuộc tính nào có IG lớn nhất sẽ được sử dụng để phân chia tập (tập con)  $S$  hiện thời trong bước lặp đang xét.

Các ưu và nhược điểm của thuật toán ID3 được liệt kê trong bảng 3.2 dưới đây

Ưu điểm	Cây quyết định dễ hiểu. Người ta có thể hiểu mô hình cây quyết định sau khi được giải thích ngắn.
	Cây quyết định có thể xử lý cả dữ liệu có giá trị bằng số và dữ liệu có giá trị là tên thể loại. Các kỹ thuật khác thường chuyên để phân tích các bộ dữ liệu chỉ gồm một loại biến. Chẳng hạn, các luật quan hệ chỉ có thể dùng cho các biến tên, trong khi mạng nơ-ron chỉ có thể dùng cho các biến có giá trị bằng số.
	Việc chuẩn bị dữ liệu cho một cây quyết định là cơ bản hoặc không cần thiết. Các kỹ thuật khác thường đòi hỏi chuẩn hóa dữ liệu, cần tạo các biến phụ (dummy variable) và loại bỏ các giá trị rỗng.
	Cây quyết định là một mô hình hộp trắng. Nếu có thể quan sát một tình huống cho trước trong một mô hình, thì có thể dễ dàng giải thích điều kiện đó bằng logic Boolean. Mạng nơ-ron là một ví dụ về mô hình hộp đen, do lời giải thích cho kết quả quá phức tạp để có thể hiểu được.
	Có thể thẩm định một mô hình bằng các kiểm tra thống kê. Điều này làm cho ta có thể tin tưởng vào mô hình.
	Cây quyết định có thể xử lý tốt một lượng dữ liệu lớn trong thời gian ngắn. Có thể dùng máy tính cá nhân để phân tích các lượng dữ liệu lớn trong một thời gian đủ ngắn để cho phép các nhà chiến lược đưa ra quyết định dựa trên phân tích của cây quyết định.
	Khó giải quyết được những vấn đề có dữ liệu phụ thuộc thời

Nhược điểm	gian liên tục
	Dễ xảy ra lỗi khi có quá nhiều lớp chi phí tính toán để xây dựng mô hình cây quyết định cao

**Bảng 3.2.** Ưu và nhược điểm của thuật toán ID3

### 3.5.2. Các phiên bản cập nhật của ID3 - C4.5 và C5.0

Sau ID3, vào năm 1993, Quinlan đã giới thiệu phiên bản cập nhật đầu tiên của nó là C4.5[26]. Do có thể được sử dụng với mục đích phân loại, thuật toán này có thể được xếp vào nhóm các thuật toán phân loại thống kê (Statistical Classifier).

Về cơ bản thuật toán C4.5 cũng có cùng cách tiếp cận như ID3 là dựa trên lý thuyết về độ hỗn loạn thông tin (Information Entropy). Tại mỗi nút của cây quyết định, C4.5 chọn ra thuộc tính có khả năng phân chia tốt nhất tập mẫu thành các tập con. Chuẩn phân loại được sử dụng trong C4.5 là độ lợi thông tin chuẩn hóa (Normalized Information Gain – NIG). Thuộc tính với NIG lớn nhất sẽ được chọn để phân chia tập mẫu đang xét. Thuật toán có các trường hợp cơ bản sau:

- Tất cả các mẫu trong tập (tập mẫu ban đầu hoặc bất kỳ tập con nào của nó) thuộc cùng một lớp. Khi đó, việc cần làm chỉ là tạo một nút lá và kết luận nút đó thuộc lớp tương ứng.
- Không có bất kỳ thuộc tính nào có thể cung cấp thông tin về độ lợi thông tin. Trong trường hợp này, C4.5 tạo một nút quyết định (decision node) bên trên cây đang xét sử dụng giá trị mong đợi (expected value) của lớp.
- Gặp phải mẫu của một lớp chưa được xác định. Tương tự như trường hợp thứ hai, C4.5 tạo một nút quyết định bên trên cây đang xét sử dụng giá trị mong đợi.

Các cải tiến của thuật toán C4.5 được trình bày trong Bảng 3.3

	Có khả năng xử lý cả các thuộc tính rời rạc và liên tục. Để xử lý các thuộc tính liên tục, C4.5 tạo ra một ngưỡng và phân chia tập đang xét vào hai tập con. Các mẫu có giá trị của thuộc tính đang xét lớn hơn ngưỡng
--	--

Các cải tiến của thuật toán C4.5	được đưa vào tập thứ nhất, các mẫu còn lại thuộc tập thứ hai[27].
	Có khả năng xử lý cả trường hợp thiếu thông tin về giá trị của thuộc tính. C4.5 đánh dấu ? cho các giá trị bị thiếu. Các giá trị bị thiếu sẽ không được sử dụng trong quá trình tính toán Entropy và IG.
	Xử lý các thuộc tính với chi phí khác nhau.
	Xén bớt cây sau quá trình khởi tạo. Sau khi cây quyết định được tạo xong, C4.5 duyệt ngược lại cây và loại bớt các nhánh không có giá trị, thay chúng bằng các nút lá

**Bảng 3.3.** Các cải tiến của C4.5

Dưới đây là mã giả của thuật toán C4.5[28]

Kiểm tra các trường hợp cơ bản có xuất hiện không

Với mỗi thuộc tính A

    Tính tỉ số của giá trị độ lợi thông tin chuẩn hóa cho A

Lựa chọn thuộc tính A<sub>best</sub> là thuộc tính có độ lợi thông tin chuẩn hóa cao nhất

Tạo cây quyết định dựa trên việc phân chia thông qua A<sub>best</sub>

Lặp lại quá trình với các tập mẫu con có được sau khi sử dụng A<sub>best</sub> để phân chia tập mẫu ban đầu, đặt các nút mới phát sinh là con của nút đang xét.

Gần đây, Quinlan đã giới thiệu thuật toán cải tiến thứ hai của ID3 có tên là C5.0[29][30]. So với C4.5, thuật toán C5.0 đạt được những cải tiến sau:

Các cải tiến của C5.0 so với C4.5	
Tốc độ	C5.0 có tốc độ nhanh hơn đáng kể so với C4.5 trên một số phương diện
Bộ nhớ	Về tổng quan, C5.0 có khả năng tận dụng bộ nhớ tốt hơn C4.5
Độ lớn của	C5.0 cho kết quả tương tự như C4.5 với cây quyết định nhỏ hơn đáng

cây quyết định	kể.
Hỗ trợ khả năng boosting	Đối với một thuật toán máy học có giám sát, boosting là khả năng giảm thiểu thiên kiến (bias) và do đó giảm thiểu phương sai[31]. Do có đặc tính này, C5.0 cải thiện cây quyết định và độ chính xác được tăng cường.
Đánh trọng số	C5.0 cho phép các đánh trọng số trên các trường hợp và loại chưa được phân lớp khác nhau.
Khả năng sàng lọc	C5.0 cho phép lựa chọn việc có sàng lọc hay không các thuộc tính không hữu dụng.

**Bảng 3.4.** Các cải tiến của C5.0 so với C4.5

Thuật toán C5.0 được tác giả Quinlan cung cấp miễn phí tại địa chỉ <http://www.rulequest.com/download.html>.

### 3.6. Phương pháp Case-based Reasoning

Phương pháp suy luận theo tình huống (Case-based Reasoning - CBR) là qui trình giải các bài toán mới dựa trên lời giải của các bài toán tương tự đã gặp. Ví dụ, một thợ sửa chữa ô tô đang chữa động cơ của một chiếc ô tô bằng cách nhớ lại một chiếc xe khác cũng có các triệu chứng tương tự, người đó đang sử dụng suy luận theo tình huống. Một luật sư đang bảo vệ một kết quả nào đó trong một phiên tòa dựa trên các tiền lệ pháp lý hay một quan tòa đang sử dụng một phán lệ (case law), người này cũng đang thực hiện lập luận theo tình huống. Cũng như vậy, một kỹ sư đang sao chép các đặc tính hoạt động của thiên nhiên vào trong công trình phỏng sinh học (biomimicry) của mình, anh ta đang coi thiên nhiên như một cơ sở dữ liệu của các giải pháp cho các vấn đề. Lập luận theo tình huống là một dạng nổi bật của phương pháp giải quyết vấn đề dựa trên việc xác định và tạo ra sự tương tự.

Suy luận theo tình huống không chỉ là một phương pháp mạnh cho suy luận máy tính mà còn là một hành vi phổ biến của con người trong cuộc sống hằng ngày khi giải quyết các vấn đề. Hay nói cách khác, mọi lập luận đều dựa trên các tình huống



trong quá khứ (mà đã được trải nghiệm hoặc chấp nhận bằng cách chủ động thực hiện chọn lựa) - lý thuyết nguyên mẫu (prototype theory) - lý thuyết được nghiên cứu sâu nhất trong ngành khoa học nhận thức về con người (human cognitive science)[32].

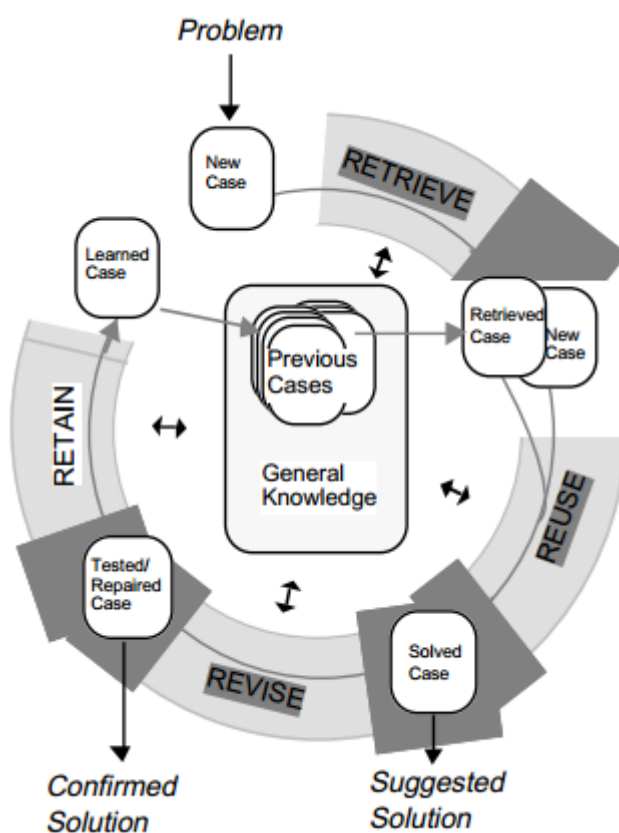
Trong tin học, Case-based Reasoning (CBR) là phương pháp kỹ thuật giải quyết vấn đề, thực hiện giải quyết các vấn đề mới bằng việc sử dụng lại những giải pháp đã có của những vấn đề trước. Những vấn đề trước đây được mã hóa gọi là các case, mỗi case chứa những thuộc tính đặc trưng của vấn đề đó và giải pháp cho nó. Một tập các case được gọi là case-base, là kiến thức nền tảng đã qua trải nghiệm, case-base được sử dụng cho quá trình đưa giải pháp cho vấn đề mới[33].

Suy luận theo tình huống đã được chính hóa phù hợp với việc ứng dụng trong suy luận tin học thành một qui trình gồm bốn bước[34]:

- Rút trích (Retrieve): Đối với một bài toán đích, truy lục từ trong bộ nhớ các tình huống có liên quan tới việc giải bài toán cần giải quyết. Một tình huống bao gồm một vấn đề, giải pháp cho vấn đề đó, và thông thường, các chú thích về lời giải đó đã được tìm ra như thế nào. Ví dụ, giả sử Dậu muốn nấu món cơm gà. Vì là người không thạo nấu ăn, kinh nghiệm gần nhất mà anh ta có thể nhớ đến là một lần anh ta nấu thành công một nồi cơm thường. Qui trình mà anh ta làm theo để nấu cơm thường, cùng với giải thích cho các quyết định mà anh ta đưa ra trong quá trình nấu, hợp thành tình huống thu được của Dậu.
- Tái sử dụng (Reuse): Ánh xạ lời giải cho tình huống trước cho bài toán đích. Bước này có thể dẫn đến việc điều chỉnh lời giải để phù hợp với tình huống mới. Trong ví dụ cơm gà, Dậu phải điều chỉnh giải pháp truy lục được để bao hàm cả phần nguyên liệu thịt gà bổ sung.
- Điều chỉnh (Revise): Sau khi đã ánh xạ lời giải trước vào bài toán đích, kiểm tra lời giải mới trong thế giới thực (hoặc giả lập) và sửa lại nếu cần thiết. Giả sử Dậu điều chỉnh giải pháp nấu cơm gà bằng cách cho thịt gà vào nấu cùng gạo ngay từ đầu. Sau khi cơm chín, anh ta phát hiện ra rằng

món ăn thu được là một món cháo đặc với thịt gà bị quá nhừ. Kết quả này gợi ý việc sửa lại như sau: không cho thịt gà vào ngay từ đầu mà xào trước rồi trộn vào sau, khi cơm đã chín.

- Lưu trữ (Retain): Sau khi lời giải đã được điều chỉnh thành công cho bài toán đích, lưu trữ kinh nghiệm thu được trong bộ nhớ dưới dạng một tình huống mới. Theo đó, Dấu ghi lại qui trình nấu cơm gà mới tìm được, nhờ đó làm giàu thêm tập các kinh nghiệm anh đã tích trữ được, và chuẩn bị tốt hơn cho những lần phải nấu ăn sau này [32].



**Hình 3.6.** Chu trình của phương pháp suy luận theo tình huống

Nói ngắn gọn, chu trình của phương pháp suy luận theo tình huống bao gồm:

- RÚT TRÍCH (RETRIEVE) một (hoặc nhiều) case tương đồng nhất với case đang xét.
- TÁI SỬ DỤNG (REUSE) thông tin và tri thức trong (các) case tìm được để tìm giải pháp cho bài toán đích.
- XEM XÉT và ĐIỀU CHỈNH (REVISE) giải pháp tìm được nếu cần thiết.

- LƯU TRỮ (RETAIN) giải pháp (kinh nghiệm) được tìm ra để sử dụng trong việc giải các bài toán tương lai.

Khi có một vấn đề mới cần phải giải quyết, vấn đề đó sẽ được biểu diễn dưới dạng case. Case mới này sẽ được so sánh với các case trong case-base, những case có độ tương đồng cao nhất với case mới sẽ được trích ra từ case-base. Tập hợp case được trích ra đó sẽ được phân tích để đưa ra giải pháp cho case mới. Giải pháp đưa ra cho case mới có thể sẽ được kiểm tra lại, nếu giải pháp đó chưa được thỏa đáng thì thực hiện tính toán lại để đưa ra giải pháp thỏa đáng hơn. Giải pháp cho vấn đề mới sẽ được lưu lại vào tập hợp các vấn đề đã có giải pháp.

Trong các phần sau, tài liệu sẽ đi sâu tìm hiểu các vấn đề chính liên quan đến phương pháp suy luận theo tình huống, bao gồm:

- Cách thức biểu diễn case.
- Phương pháp rút trích các case tương đồng.
- Tái sử dụng case
- Xem xét, điều chỉnh và lưu trữ case.

### **3.6.1. Cách thức biểu diễn case**

Một case là một mảnh kiến thức đã được ngữ cảnh hóa, biểu diễn một kinh nghiệm. Case chứa đựng bài học trong quá khứ, bao gồm nội dung của case và ngữ cảnh mà bài học có thể được sử dụng. Thông thường, một case được tạo bởi [35]:

- Một vấn đề (problem) mô tả trạng thái của thế giới (world) khi case xuất hiện.
- Một giải pháp (solution) mang tính suy luận giúp giải quyết vấn đề.
- (Và/hoặc) Một kết quả (outcome) diễn tả trạng thái của thế giới sau khi case xuất hiện.

Về tổng quan, mô tả của một case chứa tập các đặc trưng của case, giải pháp (solution) và đôi khi cả kết quả (outcome) sau khi case đó xuất hiện. Những đặc trưng của case được xác định qua một quá trình kiểm tra kiến thức: hệ chuyên gia phỏng vấn trong lĩnh vực mà nó liên quan đến, tập trung vào việc đưa ra những yêu

cầu và việc sử dụng các phương pháp kỹ thuật tập hợp dữ liệu. Ví dụ như một vấn đề về một chương trình quản lý quỹ tín dụng. Một khách hàng tiếp cận với ngân hàng và yêu cầu vay tiền. Người quản lý ngân hàng sẽ quyết định có nên cho vay hay không như thế nào? Giả sử vấn đề này được thực hiện bằng cách sử dụng hệ thống các tri thức hay hệ thống dựa trên các luật (còn gọi là hệ chuyên gia). Trong trường hợp cho ứng dụng này, case biểu diễn một sự trải nghiệm, nó nên biểu diễn những đặc trưng của ứng dụng để xác định nên hay không nên cho khách hàng vay tiền. Trong case sẽ phải chứa số lượng tiền mà khách hàng muốn vay, thời hạn trả tiền, giới tính của khách hàng, tình trạng hôn nhân, tuổi, tình trạng và những chi tiết mô tả việc làm như tiền lương, vị trí đảm trách... mục đích vay tiền làm gì, và có thể thêm một vài đặc trưng khác nữa [33].

Đặc trưng	Giá trị
Số tiền cần vay	2.500 \$
Loại hình vay tiền	Cá nhân
Thời hạn vay	6 tháng
Giới tính người vay	Nam
Tuổi	30
Tình trạng hôn nhân	Đã kết hôn
Tình trạng công việc	Đang đi làm
Thu nhập hàng tuần	260 \$
Số năm làm việc trong cơ quan	1.5 năm
hiện tại	
Đề nghị cho vay	Được chấp nhận
Kết quả (Outcome)	Tốt

**Bảng 3.5.** Mô tả một case cụ thể trong bài toán quản lý quỹ tín dụng

### 3.6.2. Phương pháp rút trích các case tương đồng

Rút trích case tương đồng liên quan đến việc tìm kiếm trong case-base các case gần nhất với vấn đề cần giải quyết. Các case được chọn trên tiêu chí có khả năng

giúp đưa ra các dự đoán tốt nhất cho case đang xét. Rút trích case tương đồng có vai trò cốt lõi trong phương pháp suy luận theo tình huống.

Có hai phương pháp rút trích chính thường được dùng trong CBR, phương pháp đầu tiên là sử dụng thuật toán cây quyết định (decision tree), phương pháp thứ hai là thuật toán k láng giềng gần nhất -- k-NN (k-Nearest Neighbour).

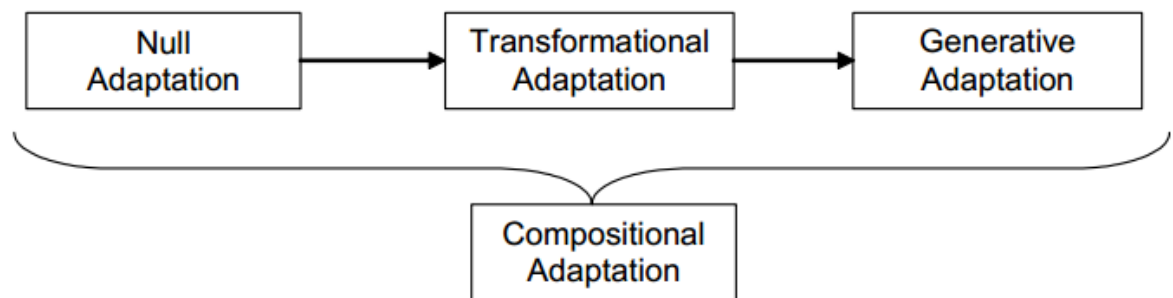
Thuật toán cây quyết định thực hiện phân tích các đặc trưng để tìm ra đặc trưng nào là tốt nhất cho việc so sánh các case với nhau. Các đặc trưng tốt đó được sắp xếp vào vào một cấu trúc cây, đặc trưng tốt nhất được đặt ở đỉnh của cây. Sau đó, các case được quản lý theo cấu trúc cây và thuật toán rút trích sẽ tìm kiếm các nút (node) có mức độ tương đồng cao nhất với case đang xét. Vì các case được sắp xếp theo cấu trúc phân cấp, thời gian rút trích tăng theo hàm logarit của số lượng các case trong case-base (tốt hơn là tăng tuyến tính theo lượng các case), đây là một ưu điểm giúp cách tiếp cận này tỏ ra hiệu quả trong trường hợp case-base lớn. Tuy nhiên, phương pháp này lại đòi hỏi một lượng case đáng kể để có thể phân biệt tốt các đặc trưng đồng thời xác định được cấu trúc có cấp bậc thích hợp. Sự phân tích này là một quy trình tốn thời gian (time consuming) và phải được thực hiện mỗi khi có case mới được đưa vào case-base. Một điều cần lưu ý nữa là cây quyết định thường không xử lý tốt các trường hợp thiếu dữ liệu về đặc trưng.

Thuật toán rút trích dựa trên k-NN thực hiện so sánh từng case trong case-base với case đang xét và tính ra độ tương đồng đối với từng trường hợp. Đơn vị đo độ tương đồng dựa trên khái niệm độ gần “close” của tập đặc trưng của các case được chọn với tập đặc trưng của case mới. Mỗi cặp đặc trưng sẽ được so sánh và cho điểm dựa trên mức độ khác biệt giữa chúng, cặp đặc trưng càng gần nhau, điểm càng cao. Có thể đánh trọng số cho các đặc trưng quan trọng hơn, dựa trên mức độ liên quan của chúng đối với vấn đề cần giải quyết. Tập k các case với độ tương đồng cao nhất sẽ được rút trích. Lời giải cho case mới sẽ được suy ra từ k case láng giềng gần nhất này. Giả sử lời giải ở đây liên quan đến sự phân chia lớp cho case mới, khi đó lớp của case mới sẽ là lớp chứa đa số các case láng giềng. Cách tiếp cận này có nhược điểm chính là thời gian tìm kiếm tăng tuyến tính theo số lượng các case trong case-

base. Tuy nhiên, một giải pháp đã được Lenz et al.(1998) đưa ra có tên là CRN (Case Retrieval Net) có thể giúp cải thiện điều này.

### 3.6.3. Tái sử dụng case

Trong những trường hợp case được rút trích hoàn toàn tương đồng với case đang xét, case được rút trích có thể được sử dụng như giải pháp (solution) cho case đang xét. Thông thường, ít có khả năng các case cũ hoàn toàn trùng khớp với một case mới, do vậy các giải pháp được mô tả trong các case cũ sẽ được hiệu chỉnh lại để tương ứng với mô tả vấn đề (problem specification) của case mới; quá trình này gọi là sự hiệu chỉnh cho phù hợp hay adaptation. Một vài kỹ thuật hiệu chỉnh có thể được sử dụng trong CBR, các kỹ thuật này tạo thành một thể liên tục (continuum) các mô hình hiệu chỉnh (adaptation model) (Wilke and Bergmann 1998, Wilke et al. 1998)



**Hình 3.7.** *Thế liên tục các mô hình hiệu chỉnh*

Như hình 3.7, có bốn mô hình hiệu chỉnh có thể được áp dụng, bao gồm:

- Null Adaptation: Là trường hợp đơn giản nhất khi không cần đến adaptation, khi đó giải pháp có thể được áp dụng trực tiếp, là trường hợp thường gặp trong các bài toán về phân lớp.
- Transformational Adaptation: Sử dụng một tập các luật để điều chỉnh các giải pháp rút trích từ case-base dựa trên sự khác biệt giữa đặc trưng của case rút trích với đặc trưng của case mới.
- Generative Adaptation: Còn được gọi là Derivational Analogy, về cơ bản là phức tạp hơn hai mô hình đầu đồng thời đòi hỏi một hệ giải quyết vấn đề (problem solver) được tích hợp chặt chẽ vào hệ CBR. Hệ giải quyết vấn

đề được sử dụng để sản sinh ra các phần còn khuyết của giải pháp, vốn tồn tại do sự khác biệt giữa đặc trưng của case rút trích với đặc trưng của case mới.

- Compositional Adaptation: tạo ra một giải pháp tổng hợp cho case mới bằng cách kết hợp các thành phần hiệu chỉnh khác nhau.

#### **3.6.4. Điều chỉnh và lưu trữ case**

Như đã đề cập, hai bước cuối trong một hệ CBR là Xem xét, điều chỉnh (Revision) và Lưu trữ (Retention). Cùng nhau, hai bước này cho phép hệ CBR có khả năng tự học. Khi giải pháp thu được trong bước tái sử dụng không đủ tốt, cơ hội để hệ tự học xuất hiện. Trong trường hợp này, giải pháp sẽ được xem xét và điều chỉnh lại để trở thành một giải pháp tốt hơn và nếu phù hợp sẽ được lưu trữ lại trong case-base, cho phép giải pháp mới này có tác động nhất định đến việc giải các vấn đề trong tương lai.

Quá trình xem xét và điều chỉnh (Revision) bao gồm 2 bước, đánh giá giải pháp (evaluation) và chẩn đoán, sửa chữa (repair) các khiếm khuyết nếu cần thiết.

Bước đánh giá liên quan đến việc đánh giá mức độ hoàn thiện của giải pháp được cung cấp bởi case-base. Việc đánh giá này có thể được tiến hành theo một số cách khác nhau. Trước tiên, có thể dựa trên phản hồi từ chuyên gia, người sử dụng, khách hàng... các nguồn phản hồi trong thế giới thật. Ngoài ra, việc đánh giá có thể dựa trên kết quả quá trình mô phỏng việc áp dụng giải pháp. Kết quả đánh giá sẽ cho biết liệu có cần thêm một quá trình adaptation sâu hơn hay bước sửa chữa hay không.

Bước lưu trữ cho phép bổ sung case đã qua quá trình xem xét và sửa chữa vào case-base, qua đó tạo cho hệ thống có khả năng tự học. Khả năng tự học, lúc này, là kết quả phụ của việc giải quyết vấn đề và có thể được xem là khả năng học qua kinh nghiệm. Những giải pháp tốt được thêm vào case-base sẽ làm tăng độ chính xác, khả năng giải quyết tốt các vấn đề tương tự trong tương lai. Trong khi đó, những giải pháp không tốt, hoặc không đúng có thể được lưu lại để tránh việc lặp lại lỗi này một lần nữa.

Tăng số lượng các mẫu đồng nghĩa với việc giúp cho hệ CBR có khả năng bao quát một phạm vi tốt hơn và, dĩ nhiên, sẽ làm việc tốt hơn. Tuy nhiên, gia tăng kích thước case-base một cách bừa bãi sẽ dẫn tới vấn đề về tính hữu dụng (utility problem). Khi vượt quá một ngưỡng nào đó, thêm case vào case-base sẽ dẫn đến giảm hiệu năng của case-base. Lúc này tốc độ rút trích sẽ tăng lên trong khi năng lực hiệu chỉnh cho phù hợp – adaptation – của hệ lại giảm xuống khi case mới được thêm vào. Vấn đề này đã dẫn đến các nghiên cứu về điều chỉnh case-base (case-base editing) nhằm giảm kích thước của case-base trong khi vẫn giữ được hiệu năng của hệ.

### **3.6.5. Ưu và nhược điểm của suy luận theo tình huống**

Phương pháp suy luận theo tình huống về cơ bản là một kỹ thuật thuộc nhóm lazy learning technique (Aha 1997), là nhánh con của lớp các thuật toán học địa phương (local learning algorithm) (Bottou and Vapnik 1992). Do vậy nó thừa hưởng các ưu và nhược điểm sau:

Đầu tiên, do thuộc nhóm lazy learner nên CBR có tính chất hoãn việc thực thi cho đến khi việc giải quyết vấn đề được yêu cầu. Trái với nó, các kỹ thuật thuộc nhóm eager learner phải khởi tạo một mô hình từ các mẫu huấn luyện trước khi yêu cầu giải quyết vấn đề xuất hiện, sau đó các kỹ thuật eager learner sẽ sử dụng mô hình này để đáp ứng khi xuất hiện yêu cầu. Do đó, các kỹ thuật lazy learner có một lợi thế là cho phép các mẫu huấn luyện mới được thêm vào hệ thống một cách dễ dàng hơn rất nhiều. Đây là điều hết sức thuận tiện trong các tình huống mà dữ liệu huấn luyện không thể (hoặc không cần) có sẵn vào lúc hệ thống khởi chạy mà chỉ được thu thập dần dần (ví dụ: trong các hệ thống online). Tuy nhiên, các kỹ thuật lazy learner lại cần không gian lưu trữ lớn, do tất cả các mẫu huấn luyện phải sẵn sàng bất cứ khi nào yêu cầu giải quyết vấn đề xuất hiện. Mặt khác, chi phí cho việc xử lý mỗi khi có yêu cầu lại khá cao. Hiện nay, tốc độ phát triển khá nhanh của phần cứng liên tục cải thiện các nhược điểm này.

Do thuộc nhánh các thuật toán học địa phương, CBR tiếp cận giải pháp thông qua việc lựa chọn các mẫu phù hợp nhất để khởi tạo một mô hình cục bộ cho mỗi yêu



cầu. Trong các trường hợp thiếu sự thống nhất của các mẫu huấn luyện, tính chất này là một lợi thế của các kỹ thuật local learner so với global learner. Cụ thể, nếu dữ liệu huấn luyện có nhiều dạng khác nhau, một kỹ thuật global learner sẽ cố gắng khởi dựng một mô hình tóm gọn các dạng khác nhau lại. Ngược lại, một kỹ thuật local learner chỉ cần sử dụng các mẫu huấn luyện thuộc những dạng tương đồng nhất với yêu cầu để đưa ra giải pháp cho yêu cầu cụ thể đó. Hiển nhiên, các kỹ thuật local learner không mắc các hạn chế mang tính can thiệp dữ liệu (Atkeson et al. 1997). Các mẫu dữ liệu mới, nếu được thêm vào tập dữ liệu huấn luyện, cũng không ảnh hưởng đến hiệu năng giải quyết vấn đề của các mẫu dữ liệu cũ.

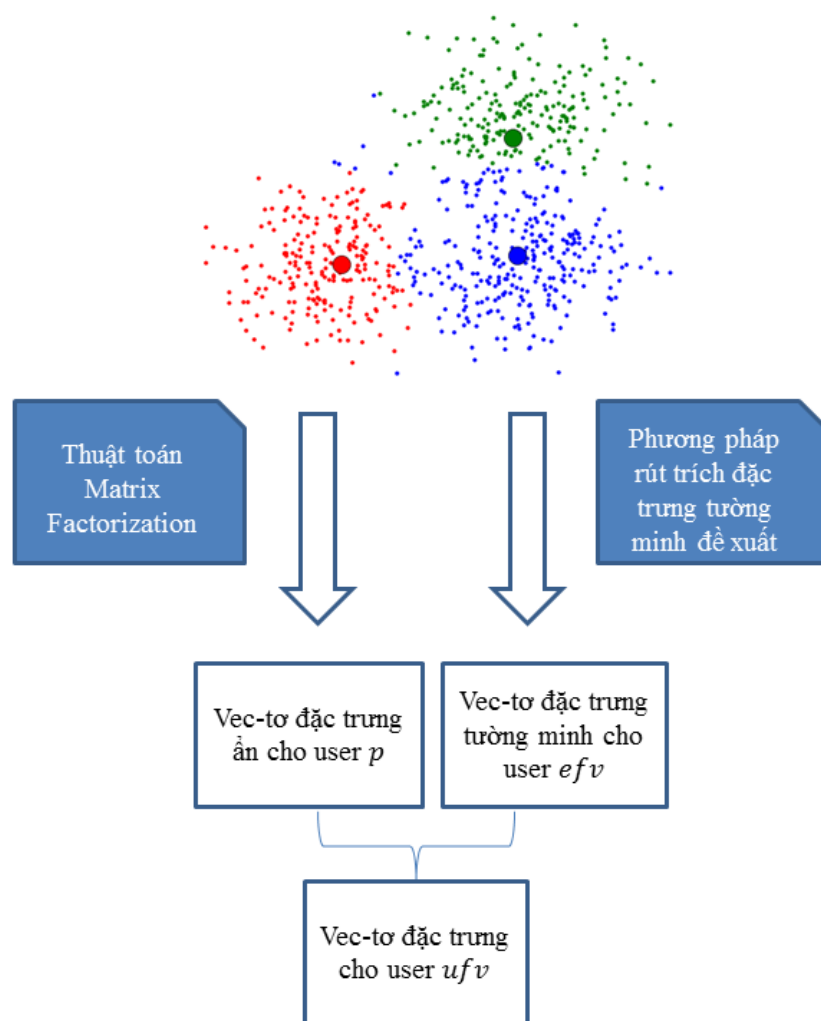
## Chương 4. THUẬT TOÁN KHUYẾN NGHỊ LAI KẾT HỢP CƠ SỞ TRI THỨC VÀ LỌC CỘNG TÁC

### 4.1. Mô hình tổng quan

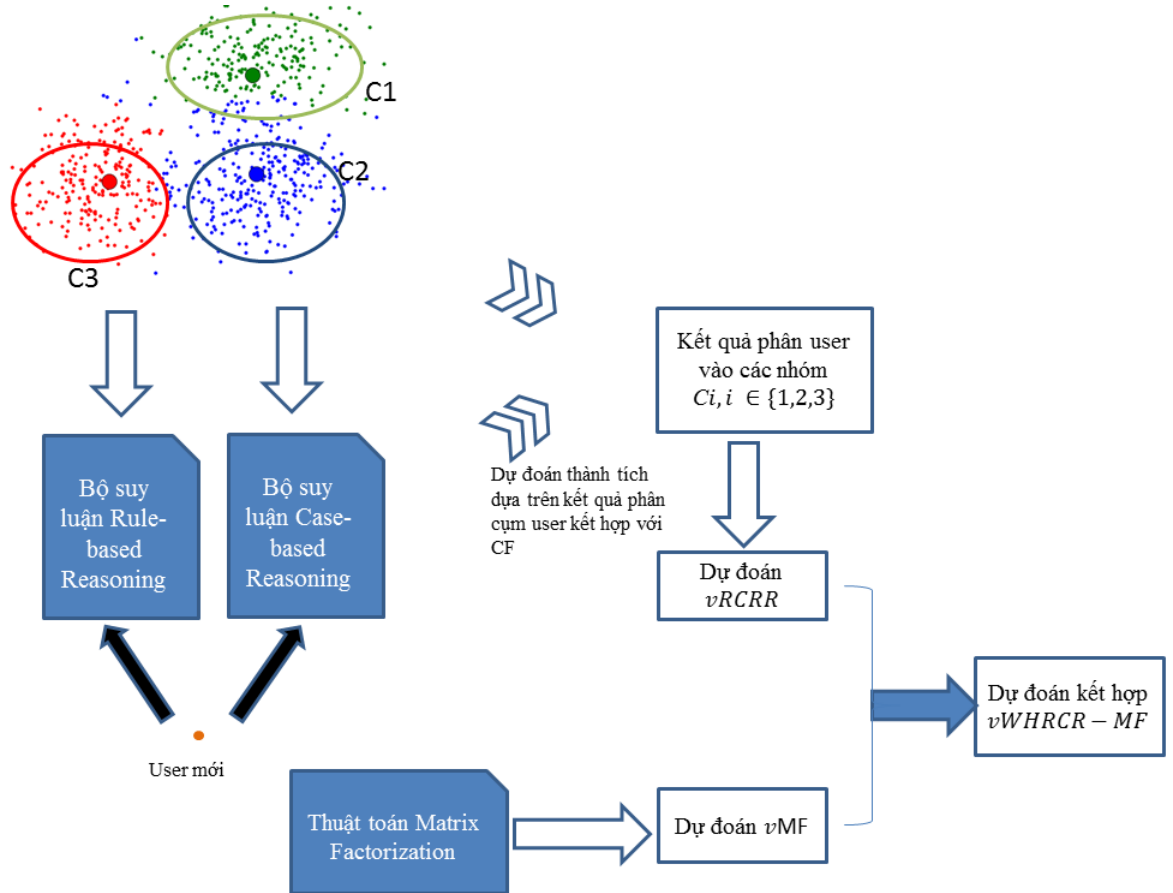
Như phần 1.4 đã trình bày, hệ khuyến nghị lai đề xuất bao gồm ba thành phần chính:

- Thành phần đầu tiên áp dụng phương pháp rút trích đề xuất để thu được vec-tơ đặc trưng tường minh của người dùng, kết hợp với vec-tơ đặc trưng không tường minh (kết quả khi áp dụng MF lên tập dữ liệu) hình thành vec-tơ đặc trưng tổng hợp biểu diễn user trong hệ.
- Thành phần thứ hai sử dụng phương pháp gom cụm k-means để xác định các cụm dữ liệu user. Tiếp theo, sử dụng phương pháp suy luận Rule-based Reasoning để tìm ra tập các luật cho phép xác định cụm của một user cụ thể. Song song đó, một phương pháp ứng dụng Case-based Reasoning được triển khai nhằm đưa ra một giải pháp xác định cụm người dùng thứ hai.
- Thành phần thứ ba sử dụng kết hợp các kết quả xác định cụm người dùng ở bước trên để dự đoán kết quả tương tác giữa user và item. Cuối cùng, kết quả dự đoán từ các phương pháp suy luận dựa trên rule-based và case-based reasoning sẽ được kết hợp với kết quả dự đoán do thuật toán MF mang lại. Việc tổng hợp tính toán dự đoán cuối cùng được thực hiện theo phương pháp tìm và áp dụng trọng số tối ưu cho mỗi trường hợp.

Trong tài liệu này tên hệ khuyến nghị lai đề xuất được đặt tên là **WHRCBR-MF** (Weighted Hybrid system of Rule-Case Based Reasoning and Matrix Factorization). Hình 4.1 và 4.2 mô tả các thành phần cơ bản của tổng quan hệ khuyến nghị lai đề xuất.



**Hình 4.1.** Mô hình tổng quan (thành phần 1) hệ khuyến nghị lai WHRCBR-MF



**Hình 4.2.** Mô hình tổng quan (thành phần 2 và 3) hệ khuyến nghị lai WHRCBR-MF

#### 4.2. Rút trích đặc trưng tường minh của người dùng

Trong tập các item (câu hỏi, bài tập, bài kiểm tra...) của một hệ e-learning, các item là khác nhau về độ khó/độ phức tạp. Dựa trên đặc tính này, ta có thể phân các câu hỏi vào  $GrNum \in \mathbb{N}$  dựa trên độ phức tạp.

Giả sử  $GrNum = 3$ , ta có thể nhóm tập item theo quy tắc sau:

- Nhóm 1 –  $G_1$  bao gồm các item mà điểm số trung bình các user đạt được nằm trong đoạn  $[0.8, 1]$ .
- Nhóm 2 –  $G_2$  bao gồm các item mà điểm số trung bình các user đạt được nằm trong đoạn  $[0.5, 0.8)$ .

- Nhóm 3 –  $G_3$  bao gồm các item mà điểm số trung bình các user đạt được nằm trong đoạn  $[0.0, 0.5)$ .

Rõ ràng nhóm  $G_1, G_2, G_3$  lần lượt chứa các item với mức độ từ khó đến trung bình và dễ đối với user (câu hỏi càng khó sẽ có càng ít user làm được hơn và ngược lại).

Sau khi xác định giá trị cụ thể  $GrNum \in \mathbb{N}$  cho toàn tập dữ liệu, đối với một user  $u$ , ta gọi  $NumOfPassG_j^u$  và  $NumOfFailG_j^u$  lần lượt là số lần  $u$  nhận được kết quả là 1 và 0 cho một lần trả lời một item thuộc nhóm  $G_j$ . Vec-tơ đặc trưng tường minh  $efv_u$  biểu diễn  $u$  được xác định như sau:

$$efv_u = (efv_u^1, efv_u^2, \dots, efv_u^j, \dots, efv_u^{GrNum}) \quad (4.1)$$

Trong đó

$$efv_u^j = \frac{(NumOfPassG_j^u - NumOfFailG_j^u)(NumOfFailG_j^u + NumOfFailG_j^u)}{|I|} \quad (4.2)$$

hoặc

$$efv_u^j = \frac{(NumOfPassG_j^u - NumOfFailG_j^u)(NumOfFailG_j^u + NumOfFailG_j^u)}{|G_j|} \quad (4.3)$$

Để minh họa, ta xét một trường hợp cụ thể với  $GrNum = 3$  (tập item được phân thành ba nhóm  $G_1, G_2, G_3$  tương ứng với ba mức độ khó giảm dần của item). Khi đó vec-tơ đặc trưng tường minh của user  $u$  sẽ có dạng:

$$efv_u = (efv_u^1, efv_u^2, efv_u^3) \quad (4.4)$$

Xét riêng trường hợp của  $efv_u^1$ , có bốn khả năng sau:

- $NumOfPassG_1^u$  và  $NumOfFailG_1^u$  lớn, tỉ số giữa  $NumOfPassG_1^u$  và  $NumOfFailG_1^u$  lớn  $\rightarrow efv_u^1$  có giá trị tuyệt đối  $Ab_1$ .
- $NumOfPassG_1^u$  và  $NumOfFailG_1^u$  nhỏ, tỉ số giữa  $NumOfPassG_1^u$  và  $NumOfFailG_1^u$  lớn  $\rightarrow efv_u^1$  có giá trị tuyệt đối  $Ab_2$
- $NumOfPassG_1^u$  và  $NumOfFailG_1^u$  lớn, tỉ số giữa  $NumOfPassG_1^u$  và  $NumOfFailG_1^u$  lớn  $\rightarrow efv_u^1$  có giá trị tuyệt đối  $Ab_3$ .

- $NumOfPassG_1^u$  và  $NumOfFailG_1^u$  nhỏ, tỉ số giữa  $NumOfPassG_1^u$  và  $NumOfFailG_1^u$  nhỏ  $\rightarrow efv_u^1$  có giá trị tuyệt đối  $Ab_4$ .

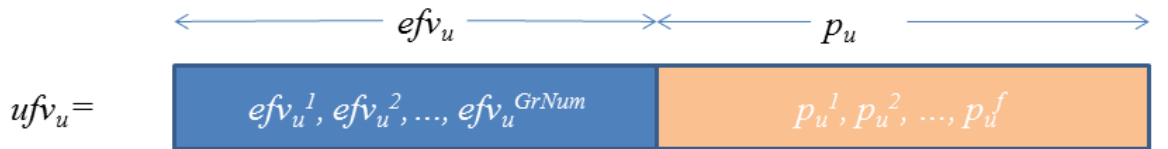
Trong đó  $Ab_1 > Ab_2 > Ab_3 > Ab_4$ . Giả sử  $|I| = 10000$  và user  $u$  có:

- $NumOfPassG_1^u = 100, NumOfFailG_1^u = 10 \rightarrow efv_u^1 = 0.99$
- $NumOfPassG_1^u = 10, NumOfFailG_1^u = 1 \rightarrow efv_u^1 = 0.0099$
- $NumOfPassG_1^u = 59, NumOfFailG_1^u = 50 \rightarrow efv_u^1 = 0.0981$
- $NumOfPassG_1^u = 1, NumOfFailG_1^u = 0 \rightarrow efv_u^1 = 0.0001$

Ta thấy mức biến thiên của  $efv_u = (efv_u^1, efv_u^2, efv_u^3)$  là khá lớn trong các trường hợp; do vậy, vector  $efv_u$  là một biểu diễn tốt các đặc trưng của một user  $u$ .

#### 4.3. Kết hợp đặc trưng tường minh và không tường minh - vector đặc trưng người dùng

Qua các phần trước, ta thấy rõ vec-tơ ẩn (không tường minh)  $p_u$  là một cách biểu diễn các đặc trưng cho một user cụ thể  $u$ . Về cơ bản, các thành phần của vec-tơ  $p_u$  là không thể xác định rõ ràng ngữ nghĩa (không thể xác định rõ các thành phần được biểu diễn bởi  $p_u$  là đặc tính nào của user  $u$ ). Trong khi đó vec-tơ  $efv_u$  lại cho phép ta nhận biết rõ các tính chất nào đang được biểu diễn, do  $efv_u$  được tính toán dựa trên khả năng hoàn thành các item với các mức độ phức tạp khác nhau. Một sự kết hợp giữa  $p_u$  và  $efv_u$  sẽ mang lại hiệu quả đáng kể trong việc mô tả đặc điểm người dùng. Ta xem xét vec-tơ đặc trưng  $ufv_u$  được tạo bởi  $p_u$  và  $efv_u$  như sau:



**Hình 4.3.** Vec-tơ đặc trưng  $ufv_u$

#### 4.4. Áp dụng thuật toán phân cụm để xác định các cụm người dùng

Để tiến hành bước này, thuật toán gom cụm k-means (thuộc nhóm thuật toán gom cụm phân hoạch) được sử dụng để phân chia tập user  $U$  vào  $k$  cụm khác nhau (với  $k$  được xác định từ đầu). Về cơ bản, bài toán phân cụm k-means là bài toán tối ưu hóa cực trị cục bộ, trong đó mỗi cụm được đặc trưng hóa bởi trung tâm của cụm (đối tượng trung bình (mean)). Độ phức tạp của thuật toán là  $O(nkt)$  với  $n = |U|$  là số lượng phần tử cần phân cụm,  $k$  là số cụm xác định trước,  $t$  là số lần lặp.

Ngoài ra các thuật toán gom cụm thuộc các nhóm khác như gom cụm phân cấp, gom cụm dựa trên mật độ, gom cụm dựa trên mô hình cũng có thể được xem xét trong các nghiên cứu về sau.

#### 4.5. Rút trích các quy tắc suy luận ra cụm người dùng dựa trên Rule-based Reasoning

Về cơ bản việc áp dụng rule-based reasoning là để rút ra các tập luật có dạng:

$$IF \langle \text{điều kiện} \rangle THEN \langle \text{ứng xử} \rangle$$

Trong bối cảnh bài toán, tập các  $\langle \text{điều kiện} \rangle$  bao gồm các trường hợp có thể được dùng để xác định cụm của một user  $u'$  cụ thể dựa trên vec-tơ đặc trưng người dùng  $ufv_{u'}$ ; trong khi đó  $\langle \text{ứng xử} \rangle$  là thao tác phân nhóm người dùng vào một cụm cụ thể. Hệ khuyến nghị đề xuất sử dụng thuật toán sinh luật C5 [15] là một cải tiến gần đây của thuật toán ID3.

#### 4.6. Suy luận cụm người dùng dựa trên Case-based Reasoning

Để áp dụng phương pháp Case-based Reasoning, ta xem mỗi user  $u$  với vec-tơ đặc trưng  $ufv_u$  đi kèm với tập các rating của  $u$  như một trường hợp (case) đã biết (mỗi user là một case).

Đối với một user  $u'$  cần xác định cụm, việc xác định cụm được thực hiện thông qua so sánh điểm Score được tính như sau [12]:

$$score(c_j) = \sum_{u \in U} \begin{cases} similarity(u, u'), & \text{nếu } cluster(u) = c_j \\ 0, & \text{các trường hợp khác} \end{cases} \quad (4.5)$$

Có nhiều thuật toán xét độ tương đồng giữa hai user. Trong tài liệu [16], công thức sau được sử dụng

$$similarity(u, u') = \frac{1}{\sqrt{\sum_{h \in H} fd(u_h, u'_h)}} \quad (4.6)$$

Trong đó  $H$  là tập các đặc trưng trong một  $ufv$  và  $fd(x, y)$  là hàm tính sự tương đồng giữa hai user được xác định như sau:

$$fd(x, y) = (x - y)^2 \quad (4.7)$$

Ngoài ra, theo [17] phương pháp giảm trọng số của các item không được tương tác nhiều sẽ mang lại kết quả tốt hơn. Công thức dưới đây được áp dụng:

$$similarity'(u, u') = \frac{Min(|V_u \cap V_{u'}|, \delta)}{\delta} similarity(u, u') \quad (4.8)$$

Với  $\delta$  là hệ số trọng số được xác định tối ưu cho mỗi bài toán cụ thể.

#### 4.7. Kết hợp logic cộng tác và các phương pháp suy diễn dựa trên tri thức

Hệ khuyến nghị WHRCBR-MF được xây dựng dựa trên việc lần lượt kết hợp kết quả dự đoán của mỗi phương pháp riêng lẻ. Phương pháp kết hợp được thực hiện theo hướng tìm ra trọng số thích hợp cho mỗi kỹ thuật sao cho trọng số của mỗi kỹ thuật tương ứng với độ chính xác trong dự đoán khi áp dụng từng kỹ thuật một cách riêng lẻ. Trước tiên, ta xem xét cách tận dụng kết quả xét cụm do hai phương pháp Rule-based Reasoning và Case-based Reasoning thu được ở các bước trước.

Lần lượt gọi các kỹ thuật dự đoán điểm số người dùng đạt được dựa trên Rule-based Reasoning và Case-based Reasoning là RBRR và CBRR (Rule-based Reasoning Recommendation và Case-based Reasoning Recommendation), công thức sau được áp dụng để dự đoán điểm số của user  $u'$  khi tương tác với item  $i$ :

$$\hat{v}_{u'i} = \overline{v_{u'}} + z \sum_{u \in c_{u'}} similarity(u, u') \times (v_{ui} - \overline{v_u}) \quad (4.9)$$



Trong công thức trên việc xác định cụm  $c_{u'}$  của user  $u'$  được thực hiện lần lượt bởi RBRR và CBRR. Đến đây ta có được hai dự đoán  $\hat{v}_{RBRR_{u'i}}$  và  $\hat{v}_{CBRR_{u'i}}$  dựa trên RBRR và CBRR.

Tiếp theo, xác định kết quả dự đoán tổng hợp dựa vào công thức:

$$\hat{v}_{RCBRR_{u'i}} = f_{u'}^{RBRR} \times \hat{v}_{RBRR_{u'i}} + f_{u'}^{CBRR} \times \hat{v}_{CBRR_{u'i}} \quad (4.10)$$

Trong các công thức (4.9) và (4.10):

- $\overline{v_u}$ : điểm trung bình của user  $u'$  được tính trên các tương tác đã có trong bộ dữ liệu huấn luyện.
- $\hat{v}_{RCBRR_{u'i}}$ : kết quả dự đoán kết hợp của RBRR và CBRR
- $z = \frac{1}{\sum_{u \in c_{u'}, similarity(u, u')}$  (4.11): hệ số chuẩn hóa. Bất kỳ hàm tính toán độ tương đồng  $similarity(u, u')$  nào cũng có thể được sử dụng. Để đơn giản, dạng hàm trong công thức (4.8) được sử dụng.
- $f_{u'}^{RBRR}, f_{u'}^{CBRR}$ : trọng số của các kỹ thuật RBRR và CBRR với điều kiện  $f_{u'}^{RBRR} + f_{u'}^{CBRR} = 1$  (4.12).

Với phương pháp trên, việc xác định  $f_{u'}^{RBRR}$  và  $f_{u'}^{CBRR}$  đóng một vai trò rất quan trọng nhằm đưa ra kết quả tối ưu nhất. Phương pháp xác định  $f_{u'}^{RBRR}$  và  $f_{u'}^{CBRR}$  sau được áp dụng:

- Xét qua tất cả các trường hợp đã biết trong đó user  $u'$  tương tác với một item  $i$  và nhận được kết quả  $v_{u'i'}$ . Sử dụng công thức (4.9) để dự đoán  $\hat{v}_{RBRR_{u'i}}$  và  $\hat{v}_{CBRR_{u'i}}$  cho mỗi trường hợp.
- Đối với mỗi trường hợp, giải hệ hai phương trình hai ẩn  $f_{u'}^{RBRR}$  và  $f_{u'}^{CBRR}$  sau để xác định  $f_{u'}^{RBRR}$  và  $f_{u'}^{CBRR}$ 

$$\begin{cases} v_{u'i'} = f_{u'}^{RBRR} \times \hat{v}_{RBRR_{u'i'}} + f_{u'}^{CBRR} \times \hat{v}_{CBRR_{u'i'}} \\ 1 = f_{u'}^{RBRR} + f_{u'}^{CBRR} \end{cases} \quad (4.13)$$
- Tính ra  $f_{u'}^{RBRR}$  và  $f_{u'}^{CBRR}$  cuối cùng bằng phương pháp lấy trung bình cộng.

Gọi kỹ thuật kết hợp RBRR và CBRR để đưa ra dự đoán là RCBRR (Rule- and Case- Based Reasoning Recommendation), với cách tiếp cận như trên, việc tích hợp kỹ thuật MF vào WHRCBR để hoàn thiện hệ WHRCBR-MF được thực hiện qua công thức:

$$\hat{v}_{u'i} = f_{u'}^{RCBRR} \times \hat{v}_{RCBRR_{u'i}} + f_{u'}^{MF} \times \hat{v}_{MF_{u'i}} \quad (4.14)$$

Trong đó,  $\hat{v}_{MF_{u'i}}$  là kết quả dự đoán của kỹ thuật MF,  $\hat{v}_{u'i}$  là kết quả dự đoán cuối cùng của hệ khuyến nghị đề xuất WHRCBR-MF,  $f_{u'}^{WHRCBR}$  và  $f_{u'}^{MF}$  lần lượt là trọng số của hai kỹ thuật WHRCBR và MF với

$$f_{u'}^{RCBRR} + f_{u'}^{MF} = 1 \quad (4.15)$$

## Chương 5. DỮ LIỆU KIỂM THỬ VÀ TIỀN XỬ LÝ DỮ LIỆU

Để đánh giá hiệu năng của hệ khuyến nghị lai đề nghị, việc lựa chọn bộ dữ liệu kiểm thử thích hợp đóng một vai trò quan trọng. Trong nghiên cứu của mình, học viên đã tìm hiểu và sử dụng bộ dữ liệu Cognitive Tutor [36]. Phần này của tài liệu tìm hiểu về cấu trúc của bộ dữ liệu gốc, các đánh giá, phân tích về bộ dữ liệu. Đồng thời các vấn đề về tiền xử lý dữ liệu cũng được đề cập đến.

### 5.1. Bộ dữ liệu kiểm thử

Bộ dữ liệu Cognitive Tutor [36] được công bố vào năm 2010 với mục đích phục vụ cho việc đánh giá các giải thuật khuyến nghị do các đội tham gia cuộc thi KDD Cup 2010 đăng tải. Bộ Cognitive Tutor bao gồm bốn tập dữ liệu con: “algebra 2005 2006”, “algebra 2006 2007”, “algebra 2008 2009”, “bridge to algebra 2006 2007”, “bridge to algebra 2008 2009”. Mỗi tập dữ liệu con chứa bốn file txt có tên dưới dạng: *<sub-dataset-name>\_train.txt*, *<sub-dataset-name>\_test.txt*, *<sub-dataset-name>\_master.txt* and finally *<sub-dataset-name>.txt*. Trong đó:

- *<sub-dataset-name>* : tên tập dữ liệu con
- *\*\_train.txt*: file dữ liệu huấn luyện, là file dữ liệu với format đầy đủ và số lượng các record nhiều nhất. Quá trình huấn luyện các thuật toán máy học, dự đoán sẽ được thực hiện trên file này.
- *\*\_test.txt*: file dữ liệu kiểm thử, là file dữ liệu để chạy thử các thuật toán. Có cấu trúc các record tương tự file *\*\_train.txt* nhưng các trường dữ liệu liên quan đến thành tích của người học trên mỗi record được giấu đi.
- *\*\_master.txt*: file dữ liệu để đánh giá thuật toán sau quá trình kiểm thử. Các record có đầy đủ các mục với cấu trúc giống như cấu trúc trên hai file *\*\_train.txt* và *\*\_test.txt*. Các record trong hai file *\*\_test.txt* và *\*\_master.txt* là giống nhau (nhưng file *\*\_master.txt* cung cấp cả các trường dữ liệu về thành tích người học)
- *\*\_.txt*: file chỉ cung cấp trường dữ liệu ROW (Id không trùng lặp của các record) tương ứng với các ROW trong hai file *\*\_test.txt* và *\*\_master.txt*.

Về cơ bản, các file dữ liệu vừa đề cập có cấu trúc và nội dung tương tự nhau (ngoại trừ file \*\_*txt* như đã nói). Tài liệu sẽ tiến hành mô tả cấu trúc file \*\_*train.txt*, cấu trúc của các file còn lại có thể được suy ra tương tự. Các file \*\_*train.txt* trong bộ dữ liệu Cognitive Tutor ghi lại các record dữ liệu, mỗi record được ghi trên một dòng riêng biệt, các trường dữ liệu của một record được ngăn cách bằng ký tự *tab*. Mỗi record chứa các thông tin về quá trình và kết quả tương tác của một học viên cụ thể với một bước trong việc giải một vấn đề do hệ thống Cognitive Tutor đưa ra. Dòng đầu tiên của một file \*\_*train.txt* cho biết các trường dữ liệu được lưu giữ trong mỗi record. Bắt đầu từ dòng thứ hai trở đi là các record. Dưới đây là hình mô tả một file \*\_*train.txt*.

Row	Anon Student Id	Problem Hierarchy	Problem Name	Problem View	Step Name	Step Start Time	Step End Time	Step Duration (sec)	Correct Step Duration (sec)	Error Step Duration (sec)	Correct First Attempt	Incorrects
1	0BrbPbwCMz	Unit ES_04, Section ES_04-1 EG4-FIXED	1	3(x+2) = 15	2005-09-09 12:24:35.0	2005-09-09 12:25:15.0	40	40	0	2	3	
2	0BrbPbwCMz	Unit ES_04, Section ES_04-1 EG4-FIXED	1	x+2 = 5	2005-09-09 12:25:15.0	2005-09-09 12:25:31.0	16	16	1	0	0	
3	0BrbPbwCMz	Unit ES_04, Section ES_04-1 EG40	1	2-8y = -4	2005-09-09 12:25:36.0	2005-09-09 12:26:12.0	36	36	0	2	3	

**Hình 5.1.** Một mẫu file dữ liệu \*\_*train.txt*

Bộ dữ liệu Cognitive Tutor được xây dựng dựa trên bốn khái niệm sau:

- **Problem – Vấn đề:** mỗi problem là một tài nguyên học tập (bài tập, bài kiểm tra). Ví dụ: Tìm diện tích vòng tròn ngoại tiếp một tam giác. Giải pháp cho một problem thông thường bao gồm nhiều step – bước.
- **Step – Bước:** Một step là một phần có thể xác định được trong một giải pháp cho một vấn đề cụ thể. Các step thuộc giải pháp cho một vấn đề có tính tuần tự (các step phải được thực hiện lần lượt mới có thể giải được vấn đề).
- **Knowledge Component (KC):** Một KC là một tri thức (công thức toán học, các định lý, tiên đề toán học...) có thể được sử dụng để giải quyết một step cụ

thể. Đối với mỗi step, số lượng KC cần có để giải step đó là một số nguyên và lớn hơn hoặc bằng không.

- Opportunity: trường dữ liệu được dùng để ghi lại số lần một người học cụ thể tiếp cận với một KC (tính đến thời điểm step hiện tại đang được thực hiện).

Bộ dữ liệu Cognitive Tutor được tạo thành dựa trên bốn thành phần căn bản Problem, Step, KC và Opportunity. Cụ thể, tên và ý nghĩa của các trường dữ liệu được trình bày trong bảng 4.1 sau đây:

STT	Tên trường dữ liệu	Ý nghĩa
01	Row	Id của từng lượt tương tác giữa người học và item (problem)
02	Anon Student Id	Id người học
03	Problem Hierarchy	Các trường biểu thị cấu trúc phân cấp của một vấn đề (problem)
04	Problem Name	
05	Problem View	
06	Step Name	Một phần có thể xác định được trong một giải pháp cho một vấn đề cụ thể
07	Step Start Time	Thời điểm bắt đầu tương tác với step
08	First Transaction Time	Thời điểm phiên tương tác đầu tiên với step
09	Correct Transaction Time	Thời điểm phiên tương tác với step có kết quả chính xác đầu tiên (người học trả lời đúng vấn đề ngay lần đầu)
10	Step End Time	Thời điểm kết thúc tương tác với step
11	Step Duration (sec)	Tổng thời gian tương tác với step
12	Correct Step Duration (sec)	Tổng thời gian tương tác với step cho kết quả chính xác
13	Error Step Duration (sec)	Tổng thời gian tương tác với step cho kết quả sai
14	Correct First Attempt	Trường cho biết liệu người học đã làm chính

		xác step ngay lần đầu tiên hay không
15	Incorrects	Số lần làm sai step
16	Hints	Gợi ý để giải step
17	Corrects	Số lần làm đúng step
18	KC(SubSkills)	Các KC cần thiết để giải quyết step. KC ở dạng SubSkills
19	Opportunity(SubSkills)	Số lần người học đã tương tác với các step có chứa các KC (SubSkills) vừa nêu
20	KC (KTracedSkills)	Các KC cần thiết để giải quyết vấn đề. KC ở dạng KTracedSkills
21	Opportunity(KTracedSkills)	Số lần người học đã tương tác với các step có chứa các KC (SubSkills) vừa nêu
22	KC (Rules)	Các KC cần thiết để giải quyết vấn đề. KC ở dạng Rules
23	Opportunity(Rules)	Số lần người học đã tương tác với các step có chứa các KC (Rules) vừa nêu

**Bảng 5.1.** Mô tả các trường dữ liệu của bộ dữ liệu Cognitive Tutor

Một điểm cần lưu ý là trong hai tập dữ liệu con “*bridge to algebra 2006 2007*” và “*bridge to algebra 2008 2009*” hai trường *KC(Rules)* và *Opportunity(Rules)* không được cung cấp.

## 5.2. Thống kê đặc tính của các tập dữ liệu kiểm thử và lựa chọn các trường dữ liệu phù hợp

Để tiến hành kiểm thử hiệu năng của hệ khuyến nghị đề nghị, hai tập dữ liệu con “*algebra 2008 2009*” và “*bridge to algebra 2008 2009*” sẽ được sử dụng. Đây là hai tập dữ liệu có số lượng record nhiều nhất và cũng là các tập được sử dụng để đánh giá hiệu năng các giải thuật tham gia cuộc thi KDD Cup 2010. Dưới đây là bảng thống kê tổng quan các đặc tính của hai tập dữ liệu:

File dữ liệu	Kích thước	Số lượng các record	Số lượng các trường dữ liệu
“algebra 2008 2009 train”	2.91GB	8,918,054	23
“algebra 2008 2009 test”	121MB	508,912	23
“bridge to algebra 2008 2009 train”	5.29GB	20,012,498	21
“bridge to algebra 2008 2009 test”	131MB	756,386	21

Bảng 5.2. Thống kê các đặc tính của hai tập dữ liệu “algebra 2008 2009” và “bridge to algebra 2008 2009”

Dựa trên bảng thống kê 4.2 ta thấy file “bridge to algebra 2008 2009 train” có số lượng record gấp 2.2 lần so với tập “algebra 2008 2009 train” trong khi lượng record trong file “bridge to algebra 2008 2009 test” chỉ hơn file “algebra 2008 2009 test” 1.5 lần. Nếu cùng sử dụng một thuật toán và tính giá trị của các trường dữ liệu được sử dụng trong hai trường hợp là như nhau, việc cài đặt hệ khuyến nghị trên tập “bridge to algebra 2008 2009” sẽ cho kết quả tốt hơn. Tuy nhiên, do tập “bridge to algebra 2008 2009” không chứa trường dữ liệu “KC(Rules)” (vốn là trường dữ liệu mang nhiều thông tin hữu ích nhất) [37], tương quan về hiệu năng trên các tập dữ liệu này vẫn chưa thể dự đoán được.

Từ các phần sau của luận văn, học viên sẽ tiến hành tìm hiểu, phân tích và sử dụng các trường dữ liệu sau: *Anon Student Id*, *Problem Hierarchy*, *Problem Name*,

*Problem View, Step Name, Correct First Attempt, Incorrects, Hints, Corrects, KC(SubSkills/KTracedSkills/Rules)*. Việc sử dụng/không sử dụng một trường dữ liệu được quyết định dựa trên tính liên quan và hữu ích của trường dữ liệu với hệ khuyến nghị lai đề nghị. Mặc dù các trường dữ liệu còn lại không được sử dụng trong khuôn khổ luận văn nhưng chúng vẫn rất hữu ích trong các cách tiếp cận/thuật toán khác. Trong tương lai, những mở rộng của luận văn sẽ hướng đến sử dụng tối đa các trường dữ liệu này.

### 5.3. Các vấn đề về tiền xử lý dữ liệu – Lựa chọn biểu cách biểu diễn một item

Một đặc điểm cần quan tâm của bộ dữ liệu Cognitive Tutor đó là trong tất cả các tập dữ liệu con đều không tồn tại một cách thức phân biệt tường minh giữa các item (step) với nhau. Điều này phần nào gây khó khăn cho việc triển khai các thuật toán trên các tập này. Trong nghiên cứu của mình, học viên thực hiện khảo sát hai giải pháp như sau:

- Sử dụng phối hợp các trường dữ liệu *Problem Hierarchy, Problem Name, Problem View, Step Name* để xác định một item (step). Trong bộ dữ liệu Cognitive Tutor, mỗi step thuộc một problem và mỗi problem thuộc một problem hierarchy, dù có những step có Step Name trùng nhau nhưng nếu ba trường dữ liệu còn lại khác nhau thì hai step cũng là khác nhau. Theo cách này mỗi item sẽ được xác định bởi bộ bốn trường dữ liệu *Problem Hierarchy, Problem Name, Problem View, Step Name*.
- Sử dụng bộ các KC(*SubSkills/KTracedSkills/Rules*) để xác định một item. Như đã nói ở phần 4.2, các KC đại diện cho các tri thức người học cần có để giải quyết được một item (step). Do vậy, việc sử dụng KC để phân biệt các step có thể tận dụng tốt các đặc tính riêng của mỗi item (step), xem sự khác nhau giữa các item (step) không phải chỉ ở tên gọi hay vị trí của chúng trong bộ dữ liệu mà ở phần tri thức hàm chứa trong chúng.

Dưới đây là bảng phân tích hai giải pháp vừa nêu:

Giải pháp	Số lượng các key có thể	Số lượng các key có thể
-----------	-------------------------	-------------------------



	sử dụng trong tập “algebra 2008 2009”	sử dụng trong tập “bridge to algebra 2008 2009”
Sử dụng các trường dữ liệu “ <i>Problem Hierarchy</i> ”, “ <i>Problem Name</i> ”, “ <i>Problem View</i> ”, “ <i>Step Name</i> ”	1,416,473	887,740
Sử dụng các trường dữ liệu KC	2,979	1,458

**Bảng 5.3.** Bảng phân tích hai giải pháp xác định item trong bộ dữ liệu *Cognitive Tutor*

Qua xem xét, giải pháp thứ nhất tỏ ra có các nhược điểm sau:

- Khả năng thích nghi kém với vấn đề cold-start problem.
- Việc áp dụng dẫn đến các tập dữ liệu trở nên thừa hơn rất nhiều so với giải pháp thứ hai.

Bên cạnh đó giải pháp thứ hai lại có các nhược điểm sau:

- Khi một item (step) không chỉ đòi hỏi một KC duy nhất, việc biểu diễn các item (step) phải được cải tiến thêm để thể hiện đúng bản chất của item (step).
- Trong một số trường hợp, không có KC nào liên quan đến item (step) được ghi lại.

Đối với giải pháp thứ hai, nhược điểm thứ nhất có thể được khắc phục thông qua việc xem một nhóm các KC cũng tương đương với một item (step). Như vậy, mặc dù số lượng các item (step) có tăng thêm nhưng vẫn đảm bảo tính toàn vẹn của dữ liệu. Để giải quyết nhược điểm thứ hai, có thể xem tất cả các item (step) không có KC là một item duy nhất và có một ID chung.

Trong nghiên cứu của mình, học viên sử dụng giải pháp thứ hai; trong đó, xem các item (step) không có KC là một item duy nhất có ID là 0. Bảng 4.4 thống kê tình trạng các tập dữ liệu khi thực hiện cách giải quyết này.

<b>Đặc điểm</b>	<b>Tập “algebra 2008 2009”</b>	<b>Tập “bridge to algebra 2008 2009”</b>
<b>Số lượng các item được xác định bởi một KC duy nhất</b>	541	993
<b>Số lượng các item được xác định bởi một nhóm gồm nhiều KC</b>	441	615
<b>Tổng số lượng các item</b>	982	1608

**Bảng 5.4.** *Thống kê tình trạng các tập dữ liệu khi sử dụng KC để xác định item (step)*

## Chương 6. THỰC NGHIỆM VÀ PHÂN TÍCH KẾT QUẢ

### 6.1. Lựa chọn dữ liệu và mô hình thực nghiệm

Để đánh giá hiệu năng của hệ khuyến nghị đề xuất, học viên đã sử dụng hai tập dữ liệu “*algebra 2008 2009*” và “*bridge to algebra 2008 2009*” trong bộ dữ liệu Cognitive Tutor để tiến hành thực nghiệm và đưa ra các phân tích.

Vấn đề quan trọng nhất khi triển khai thực nghiệm là xác định các hệ số ban đầu của hệ ( $f, \delta, \lambda, \gamma, GrNum \dots$ ). Dưới đây là bảng mô tả các hệ số được sử dụng trong thực nghiệm.

Hệ số	Giá trị được chọn
$f$ (số chiều của không gian chứa các vector ẩn trong thuật toán MF)	64 đối với tập “ <i>algebra 2008 2009</i> ” 128 đối với tập “ <i>bridge to algebra 2008 2009</i> ”
$\gamma$ (tốc độ học – learning rate – trong thuật toán MF)	0.005 đối với tập “ <i>algebra 2008 2009</i> ” 0.001 đối với tập “ <i>bridge to algebra 2008 2009</i> ”
$\lambda$ (trọng số chuẩn - regularization term hay regularization weight trong thuật toán MF)	0.015 đối với tập “ <i>algebra 2008 2009</i> ” 0.0015 đối với tập “ <i>bridge to algebra 2008 2009</i> ”
$iter$ (số lần lặp khi áp dụng phương pháp leo đồi trong thuật toán MF)	120 đối với tập “ <i>algebra 2008 2009</i> ” 80 đối với tập “ <i>bridge to algebra 2008 2009</i> ”
$k$ (số lượng cụm người dùng)	40
$\delta$ (hệ số trọng số của phương pháp Case-based reasoning)	35
$GrNum$ (số lượng nhóm các độ khó)	10

của step)	
-----------	--

**Bảng 6.1.** Thiết lập hệ số cho thực nghiệm hệ WHRCBR-MF trên hai tập dữ liệu “algebra 2008 2009” và “bridge to algebra 2008 2009”

## 6.2. Kết quả thực nghiệm và đánh giá.

Việc áp dụng hệ khuyến nghị đề xuất trên hai tập dữ liệu kiểm thử cho kết quả rất khả quan, so sánh với kết quả dựa trên chỉ số RMSE của những nhóm thi vào thời điểm 2010 ta có bảng kết quả như sau

Bộ dữ liệu kiểm thử	RMSE đối với bộ dữ liệu “algebra 2008 2009”	RMSE đối với bộ dữ liệu “bridge to algebra 2008 2009”	RMSE trung bình
Nhóm hạng nhất - NTU team	0.274311	0. 271157	0. 272734
Nhóm hạng hai - National Taiwan University team	0.274293	0.271285	0.272789
Nhóm hạng ba – starfish team	0.274955	0.271806	0.273381
Thuật toán WHRCBR-MF	<b>0.273452</b>	<b>0. 273704</b>	<b>0.273578</b>
Nhóm hạng tư - Zhang and Su team	0.274916	0.272449	0.273683
Nhóm hạng năm- BigChaos	0.276473	0.272639	0.274556
Thuật toán MF	0.298983	0.294464	0.296724

Thuật toán RBRR	0.285249	0.280938	0.283094
Thuật toán CBRR	0.283347	0.279064	0.281206

**Bảng 6.2.** So sánh kết quả dự đoán của WHRCBR-MF với các thuật toán dự thi KDD Cup 2010 và các thuật toán cơ sở

Hệ khuyến nghị đề xuất mang lại kết quả khá tốt, dựa trên tương quan giữa độ phức tạp của thuật toán và hiệu năng mang lại. Mặc dù, độ chính xác không bằng được các thuật toán xếp ở ba hạng đầu nhưng so với các thuật toán cần thời gian chạy lâu và phát sinh ra các mô hình phức tạp thì thuật toán đơn giản như WHRCBR-MF mang lại kết quả khá khả quan.

So với các thuật toán cơ sở (MF, RBRR và CBRR), hệ khuyến nghị đề xuất mang lại cải thiện về hiệu năng rõ rệt. Cụ thể, mức độ cải thiện của WHRCBR-MF so với MF, RBRR và CBRR lần lượt là 7.8%, 3.36% và 2.71%.

Một điều đáng lưu ý rằng bộ dữ liệu “*bridge to algebra 2008 2009*” có kích thước lớn hơn nhưng sự cải thiện về hiệu năng dự đoán trên tập này lại kém hơn. Điều này có thể lý giải do việc sử dụng KC (KTracedSkills) thay cho KC(Rules) làm kém đi tính chính xác trong việc mô tả các item (và đồng thời cả các user). Lý do là KC(KTracedSkills) chứa ít thông tin có giá trị hơn KC(Rules) [37].

## **Chương 7. KẾT LUẬN**

### **7.1. Kết quả đạt được**

Luận văn đã tìm hiểu qua các vấn đề cơ bản của hệ khuyến nghị và hệ cơ sở tri thức. Các thuật toán tiêu biểu đã được tìm hiểu và tận dụng tối đa trong hệ khuyến nghị đề xuất.

Bên cạnh đó, luận văn đã tìm hiểu và đề xuất các bước tiền xử lý dữ liệu phù hợp cho bài toán đặt ra và bộ dữ liệu thực nghiệm Cognitive Tutor.

Đồng thời, luận văn đã đề xuất một cách biểu diễn đặc trưng người dùng thông qua cách kết hợp các đặc trưng tường minh và không tường minh với nhau.

Cuối cùng, luận văn đã xây dựng được mô hình kết hợp có trọng số giữa hai phương pháp khuyến nghị hiện nay đó là dựa trên cơ sở tri thức và dựa trên lọc cộng tác.

Dựa trên kết quả nghiên cứu, học viên đã viết được một bài báo, hiện đang chờ xét duyệt.

### **7.2. Hướng phát triển**

Nhằm nâng cao hơn nữa hiệu năng dự đoán của hệ khuyến nghị đề xuất, các hướng sau sẽ được xem xét trong tương lai:

- Cải thiện mô hình bằng cách sử dụng các thuật toán dựa trên cơ sở tri thức, các kỹ thuật phân cụm phức tạp hơn.
- Sử dụng thuật toán bias-MF là một cải tiến của MF dựa trên ý tưởng đánh giá/kết quả của user cụ thể khi tương tác với một item nhất định sẽ phụ thuộc vào đặc tính riêng của cả user và item đó.
- Nghiên cứu áp dụng các phương pháp chèn thông tin về ngữ cảnh (ví dụ: thời gian) vào hệ khuyến nghị. Trong thực tế, nhu cầu, sở thích hoặc năng lực của user có thể thay đổi theo ngữ cảnh, do đó việc các thông tin về ngữ cảnh được thêm vào trong quá trình khuyến nghị sẽ mang lại sự cải thiện đáng kể.

- Tích hợp hệ khuyến nghị đề xuất vào các hệ thống học tập trực tuyến thực tế, tổng hợp, đánh giá việc áp dụng hệ khuyến nghị đề xuất trong thực tế.

## **TÀI LIỆU THAM KHẢO**

- [1] Mehta, B., Hofmann, T., "A survey of attack-resistant collaborative filtering algorithms," in *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, vol. 31, 2008, p. 14–22.
- [2] Michael P. O'Mahony, Neil J. Hurley, and Guenole C.M. Silvestre, "Promoting recommendations: An attack on collaborative filtering," in *Database and Expert Systems Applications*, vol. 2453, Springer-Verlag Berlin Heidelberg, 2002, pp. 213-241.
- [3] Chirita, P.A., Nejdl, W., Zamfir, C., "Preventing shilling attacks in online recommender systems," in *WIDM '05: Proceedings of the ACM Workshop on Web Information and Data Management*, 2005.
- [4] Yehuda Koren, Robert Bell and Chris Volinsky, "Matrix Factorization for Recommendation Systems," 2009.
- [5] Bobadilla, J., Serradilla, F., Hernando, A., "Collaborative filtering adapted to recommender systems of e-learning," in *Knowledge-Based Systems*, vol. 22, 2009, pp. 261-265.
- [6] Soonthornphisaj, N., Rojsattarat, E., Yim-Ngam, S., "Smart e-learning using recommender system," in *Proceedings of the 2006 international conference on Intelligent computing*, Berlin, Heidelberg, 2006.
- [7] Tan, H., Guo, J., Li, Y., "E-learning Recommendation System," in *Proceedings of the 2008 International Conference on Computer Science and*

*Software Engineering*, Washington, DC, USA, 2008.

[8] Tang, T., McCalla, G., "Evaluating a Smart Recommender for an Evolving E-learning System: A Simulation-Based Study," in *Advances in Artificial Intelligence*, Berlin Heidelberg, Springer, 2004, p. 439–443.

[9] Thai-Nghe, N., Drumond, L., Krohn-Grimberghe, A., Schmidt-Thieme, L., "Recommender system for predicting student performance," *Procedia Computer Science*, vol. 1, no. 2, pp. 2811-2819, 2010.

[10] Salehi, M., Kmalabadi, I.N., Ghouschi, M.B.G., "A New Recommendation Approach Based on Implicit Attributes of Learning Material," *IERI Procedia*, vol. 2, p. 571–576, 2012.

[11] Dunlavy, D. M., Kolda, T. G., and Acar, E., "Temporal link prediction using matrix and tensor factorizations," *ACM Transactions on Knowledge Discovery from Data*, vol. 5, no. 2, 2011.

[12] Shweta Tyagi, Kamal K. Bharadwaj, "A Hybrid Recommender System Using Rule-Based and Case-Based Reasoning," *International Journal of Information and Electronics Engineering*, vol. 2, no. 4, 2012.

[13] Robin Burke, "Hybrid Web Recommender Systems," in *The Adaptive Web, Lecture Notes in Computer Science*, vol. 4321, Berlin, Germany, Springer-Verlag, pp. 377-408.

[14] Alexander Felfernig and Robin Burke, "Constraint-based Recommender Systems: Technologies and Research Issues," in *Proceedings of the ACM International Conference on Electronic Commerce (ICEC'08)*, Innsbruck, Austria, 2008.

[15] R. Quinlan, "C5.0. Release 2.02," September 2005. [Online]. Available: <http://www.rulequest.com/see5-info.html>.



- [16] D. W. Aha, "Case-Based Learning Algorithm," in *ARPA Case-Based Reasoning Workshop*, Morgan Kaufmann, 1991, pp. 147--158.
- [17] H. Ma, I. King and M. R. Lyu, "Effective missing data prediction for collaborative filtering," in *30th annual international ACM SIGIR conference on research and development in information retrieval*, 2007.
- [18] "Wikipedia," [Online]. Available:  
[https://en.wikipedia.org/wiki/Matrix\\_decomposition](https://en.wikipedia.org/wiki/Matrix_decomposition).
- [19] Inderjit S. Dhillon, Suvrit Sra, "Generalized nonnegative matrix approximations with Bregman divergences," in *Neural Information Proc. Systems*, 2005.
- [20] Rashish Tandon and Suvrit Sra, "Sparse nonnegative matrix approximation: new formulations and algorithms," 2010.
- [21] S. Funk, "The Evolution of Cybernetics," 11 December 2006. [Online]. Available: <http://sifter.org/~simon/journal/20061211.html>.
- [22] R. Quinlan, "Induction of Decision Trees," in *Machine Learning*, vol. 1, 1986, pp. 81-106.
- [23] G. V. Kass, "An exploratory technique for investigating large quantities of categorical data," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 29, no. 2, pp. 119-127, 1980.
- [24] Hothorn, T., Hornik, K., Zeileis, A., "Unbiased Recursive Partitioning: A Conditional Inference Framework," *Journal of Computational and Graphical Statistics*, vol. 15, no. 3, p. 651--674, 2006.
- [25] Strobl, C., Malley, J., Tutz, G., "An Introduction to Recursive Partitioning: Rationale, Application and Characteristics of Classification and Regression Trees, Bagging and Random Forests," *Psychological methods*, vol. 14, no. 4, pp.

323-348, December 2009.

[26] R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993.

[27] R. Quinlan, "Improved use of continuous attributes in c4.5," *Journal of Artificial Intelligence Research*, vol. 4, pp. 77-90, 1996.

[28] S. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques," in *Proceedings of the 2007 conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, 2007.

[29] R. Quinlan, "Is See5/C5.0 Better Than C4.5?," February 2012. [Online]. Available: <http://www.rulequest.com/see5-comparison.html>.

[30] Kuhn, M., Johnson K., Applied Predictive Modeling, Springer, 2013.

[31] L. Breiman, "Bias, Variance, and Arcing Classifiers," 1996.

[32] "Wikipedia," [Online]. Available: [https://vi.wikipedia.org/wiki/L%E1%BA%ADp\\_lu%E1%BA%ADn\\_theo\\_t%C3%ACnh\\_hu%E1%BB%91ng](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_lu%E1%BA%ADn_theo_t%C3%ACnh_hu%E1%BB%91ng).

[33] D. SJ, "Using Case-Based Reasoning for Spam Filtering, PhD Thesis," 2006.

[34] Aamodt, A., Plaza, E., "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches," *Artificial Intelligence Communications*, vol. 7, no. 1, pp. 39-59, 1994.

[35] Watson, I., Marir, F., "Case-Based Reasoning: A Review," *The Knowledge Engineering Review*, vol. 9, no. 4, pp. 355-381, 1994.

[36] "KDD Cup 2010 Educational Data Mining Challenge," 2010. [Online]. Available: [https://pslcdatashop.web.cmu.edu/KDDCup/rules\\_data\\_format.jsp](https://pslcdatashop.web.cmu.edu/KDDCup/rules_data_format.jsp).

[37] Koedinger, K., Baker, R., Cunningham, K., Skogsholm, A., Leber, B. & Stamper, J., "A data repository for the edm community: The pslc datashop," in *Handbook of Educational Data Mining, Lecture Notes in Computer Science*, CRC, pp. 5, 13, 16, 18, 20, 26, 54, 153.