So sánh một số thuật toán cơ bản cho hệ thống khuyến nghị việc làm

Huỳnh Ngọc Tín, Nguyễn Thanh Anh Tuyên, Trần Minh Luận

*Trường đại học Công Nghệ Thông Tin – ĐHQGTP-HCM*

*Email:* tinhn@uit.edu.vn*,* tuyense07@gmail.com*, xcrossworker@gmail.com*

**Tóm tắt:** Khuyến nghị việc làm là bài toán đưa một danh sách được sắp sếp các việc làm được máy tính dự đoán là phù hợp đến với người tìm việc. Có rất nhiều phương pháp khuyến nghị khác nhau như phương pháp lọc nội dung phương pháp lọc cộng tác, phương pháp lai,... Vậy phương pháp nào là tốt nhất cho hệ thống khuyến nghị việc làm? Trong bài báo này chúng tôi trình bày một số đánh giá, so sánh một số phương pháp khuyến nghị cơ bản và đưa ra một vài nhận định.

1 Giới thiệu

Tìm kiếm việc làm là một nhu cầu của tất cả mọi người. Với sự trợ giúp của các công cụ đăng tuyển trực tuyến nên các ứng viên có thể tìm kiếm các thông tin ứng tuyển nhanh và dễ dàng hơn. Tuy vậy, các công cụ này thường chứa rất nhiều thông tin việc dẫn đến quá tải. Theo thống kê ngày 22/11/2016 trang tuyển dụng Vietnamworks hiện có 8,116 đăng tuyển [1], trang careerbuider hiện có 8,895 đăng tuyển [2],... với đăng tuyển nhiều như vậy các hệ thống tìm kiếm thông thường sẽ cho ra rất nhiều kết quả khác nhau và gây ra nhiễu cho người dùng. Vì vậy xây dựng một hệ thống để thay thế hoặc hỗ trợ các hệ thống tìm kiếm thông thường sẽ mang lại nhiều lợi ích hơn cho ứng viên và nhà tuyển dụng.

Việc xây dựng hệ thống khuyến nghị [3] việc làm có thể giúp cho người tìm việc giảm được thời gian tìm kiếm và đem lại hiệu quả cao hơn các hệ thống lọc tìm kiếm thông thường. Đây là đề tài được nhiều khoa học gia trên thế giới quan tâm và có một số website đăng tuyển đã đưa hệ thống khuyến nghị vào sử dụng như Linkedin, Indeed,… và ở Việt Nam cũng đã có một số hệ thống khuyến nghị tương tự như Vietnamworks, ITViec. Tuy nhiên các hệ thống trên vẫn còn rất đơn giản. Để có thể tìm ra được phương pháp nào là tốt cho hệ thống khuyến nghị việc làm, chúng tôi đã tiến hành một số thực nghiệm trên vài thuật toán khuyến nghị đơn giản và có một số đánh giá, so sánh các thuật toán này với nhau. Phần tiếp theo của chương này chúng tôi sẽ đi qua một vài khái niệm về hệ thống khuyến nghị và khuyến nghị việc làm.

2 Phát biểu bài toán

Hệ khuyến nghị tiếng Anh là Recommender System hay Recommendation System, là một hệ thống con của hệ thống lọc thông tin dùng để tiên đoán mức độ yêu thích mà một người dùng dành cho một đối tượng nào đó [4].

Hệ khuyến nghị việc làm là hệ thống được thiết kế nhằm mục đích đưa đến cho người tìm việc những công việc được dự đoán là phù hợp với họ. Để từ đó người tìm việc có thể dễ dàng chọn lựa công việc phù hợp với bản thân

3 Một số nghiên cứu liên quan

4 Hướng tiếp cận

1. **Các thuật toán khuyến nghị**
2. **Phương pháp tiếp cận lọc cộng tác**

Tiếp cận lọc cộng tác (Collaborative filtering approach) hay viết tắt là CF, là phương pháp tiếp cận dựa trên dữ liệu hành vi, sở thích trong quá khứ của người dùng (ma trận đánh giá), ý tưởng cơ bản của CF là nếu người dùng có cùng sở thích ở quá khứ thì cũng sẽ có cùng sở thích trong tương lai.

1. Ma trận đánh giá [4]

Cho không gian người dùng và không gian các đối tượng khuyến nghị . Ma trận kích thước chứa các giá trị đánh giá với . Những giá trị đánh giá thể hiện mức độ hữu ích của đối tượng với người dùng (hay ). Giá trị có thể là số nguyên hay số thực trong một khoảng cố định nào đó tùy thuộc vào bài toán cụ thể. Thông thường giá trị đánh giá trong các hệ thống ứng dụng phổ biến thường từ 1 (không hữu ích) đến 5 (rất hữu ích). Nếu một người dùng chưa thể hiện đánh giá với một đối tượng thì và cần được tính toán (dấu ? trong hình 2.1)



Hình 1. Dấu ? là các giá trị cần tiên đoán trong ma trận đánh giá.

Tiếp cận lọc cộng tác được xem là hướng tiếp cận phổ biến nhất và thành công nhất để xây dựng hệ khuyến nghị cho các hệ thống thương mại điện tử [4] [5]. Với rất nhiều các nghiên cứu, cài đặt và thực nghiệm đã thực hiện trước đó, CF được chia thành 2 nhóm chính gồm: CF dựa trên bộ nhớ và CF dựa trên mô hình. Trong phạm vi của đề tài, chúng tôi tiến hành tìm hiểu cài đặt và thực nghiệm với các thuật toán CF dựa trên bộ nhớ.

CF dựa trên bộ nhớ dùng các kỹ thuật của thống kê để tính toán tìm người dùng hoặc đối tượng khuyến nghị tương tự từ thông tin trong ma trận đánh giá. Tiếp cận CF dựa trên bộ nhớ tìm cách ước lượng giá trị của hàm hữu ích , thể hiện cho mức độ hữu ích của đối tượng khuyến nghị đối với người dùng . Dựa trên ma trận đánh giá, CF sẽ khuyến nghị cho người dùng các đối tượng tương tự với đối tượng mà người dùng đã đánh giá trước đó (lọc dựa trên đối tượng khuyến nghị - Item based) hoặc khuyến nghị các đối tượng mà những người dùng có đồng sở thích với người dùng đã đánh giá (lọc dựa trên người dùng - User based).

Tiếp theo sẽ trình bày chi tiết về 2 phương pháp lọc dựa trên người dùng (User based) và lọc dựa trên đối tượng khuyến nghị (Item based).

**(\*) Lọc dựa trên người dùng**

1. Những người dùng đồng sở thích [4]

Những người dùng đồng sở thích là những người có sở thích hay những đánh giá trong quá khứ tương tự với người dùng trên cùng những đối tượng khuyến nghị từ ma trận đánh giá. Ký hiệu là những người dùng đồng sở thích với , trong đó , , là không gian người dùng của hệ thống.

**Ý tưởng chính:** ý tưởng thực hiện của phương pháp lọc dựa trên người dùng gồm 3 bước chính: đầu tiên xác định danh sách những người dùng có cùng sở thích với người dùng hay ; Bước 2 tiến hành ước lượng giá trị hàm hữu ích của đối tượng khuyến nghị với người dùng bằng cách tổng hợp giá trị đánh giá của đối với . Bước 3 là thực hiện khuyến nghị dựa trên giá trị hàm hữu ích ước lượng được [4] [5].

**Các bước thực hiện:** các tác giả Owen và cộng sự [6] đã tiến hành cài đặt thuật toán với các bước có thể tóm gọn theo mã giả như sau:

for(every other user ui)

{

compute Sim(u, ui) //a similarity between u and ui

retain the top users, ranked by similarity as neighborhood N

}

for(every item p that some user in N has rated but user u has no rated)

{

for(every other user ui in N that has rated for p)

{

compute similarity S between u and ui

incorporate ui rating, weighted by S into a running average

}

}

Trong đó các phương pháp để tính độ tương tự () giữa người dùng hoặc giữa các việc làm được tính theo các độ đo tương tự phổ biến như Euclidean Distance, Pearson Correlation Coefficient, Tanimoto Coefficient hay LogLikilihood Similarity.

Việc xác định danh sách những người dùng đồng sở thích sẽ ảnh hưởng trực tiếp đến kết quả của hàm hữu ích, vì vậy ta cần cân nhắc để chọn danh sách TopN tốt nhất. Nếu chúng ta xem xét tất cả các người dùng đồng sở thích sẽ ảnh hưởng đến hiệu suất của việc tính toán và cũng sẽ ảnh hưởng tiêu cực đến kết quả tính toán bởi những người dùng có mức độ tương tự thấp.

Trong thực tiễn có 2 phương pháp được sử dụng phổ biến đó là lấy danh sách tất cả người dùng có độ tương tự lớn hơn một ngưỡng (threshold) nhất định hoặc lấy danh sách TopN cố định n người dùng có độ tương tự cao nhất.

Với cách thứ nhất ta có phải xem xét ngưỡng (threshold) ở mức nào là hợp lý, vì nếu chọn quá cao thì danh sách người dùng sẽ rất hạn chế (thậm chí không có), còn nếu chọn thấp quá thì danh sách này dường như không bị cắt giảm là bao so với ban đầu.

Với cách thứ 2 kích thước của tập người dùng đồng sở thích sẽ không thể bao phủ, giá trị n được chọn nếu quá cao sẽ không giới hạn được danh sách người dùng (tương tự khi chọn threshold thấp) điều này sẽ dẫn đến nhiễu dữ liệu tiên đoán, nhưng nếu chọn n quá thấp sẽ ảnh hưởng tiêu cực đến khả năng tiên đoán của hệ thống.

Tổng hợp đánh giá

Sau khi đã tìm được danh sách người dùng có cùng sở thích và mức độ tương tự (giá trị ) ta cần tổng hợp lại để được kết quả cuối cùng của hàm hữu ích để thực hiện khuyến nghị.

Owen và cộng sự [6] đã nghiên cứu và cài đặt với công thức sau cho thư viện Mahout [7]:

Trong đó:

* + N là số người dùng có sở thích tương tự với u
  + M là số người dùng có đánh giá cho đối tượng p
  + là giá trị đánh giá của u với p trong ma trận đánh giá

**(\*\*) Lọc dựa trên đối tượng khuyến nghị**

Tương tự như lọc dựa trên người dùng, lọc dựa trên đối tượng khuyến nghị cũng với ý tưởng và các bước thực hiện khá giống nhau gồm 3 bước chính: đầu tiên xác định danh sách những đối tượng khuyến nghị tương tự nhất với đối tượng khuyến nghị ; Bước 2 tiến hành ước lượng giá trị hàm hữu ích của đối tượng khuyến nghị với người dùng bằng cách tổng hợp giá trị đánh giá của cho đối với . Bước 3 là thực hiện khuyến nghị dựa trên giá trị hàm hữu ích ước lượng được [4] [5].

1. **Phương pháp tiếp cận nội dung**

Hướng tiếp cận dựa trên nội dung (Content based approach) [17] là một dạng đặc biệt của bài toán truy vấn và lọc thông tin [18]. Hướng tiếp cận này tìm cách khuyến nghị cho người dùng những đối tượng có nội dung tương tự với những đối tượng mà người dùng quan tâm trong quá khứ [18]. Hay nói cách khác tiếp cận theo hướng nội dung trả lời cho câu hỏi sau của người sử dụng: “Hãy cho tôi biết những đối tượng tương tự với những đối tượng mà tôi thích” [18]. Các đối tượng ở đây có thể là các video ca nhạc, game, bài báo, việc làm, sản phẩm,… Ví dụ, một người thường xem những đăng tuyển liên quan đến những từ khóa như java, mysql, lập trình,… thì hệ thống đưa ra cho người đó những việc làm chứa những từ khóa liên quan mà người dùng chưa xem.

Để xây dựng một hệ thống khuyến nghị tiếp cận theo hướng nội dung chúng ta cần cung cấp cho hệ thống tập các thông tin sẵn có về đối tượng được khuyến nghị như thể loại, mô tả, tính chất, đặc điểm… và một tập dữ liệu mô tả sở thích của người dùng. Nhiệm vụ của hệ thống là “học” sở thích của người dùng và tìm ra những đối tượng có đặc điểm tương tự với sở thích của người dùng.

**(\*) Phương pháp tính độ tương tự**

Độ tương tự của người dùng u và đối tượng i có thể tính bằng độ tương tự Cosine [19], khoảng cách Euclide [13], hoặc sử dụng Tanimoto [20] để tính toán. Thông thường các nghiên cứu sử dụng độ tương tự Cosine để tính toán. Công thức tính Cosine được biểu diễn như sau:

Trong đó A và B là các vector đặc trưng. Có nhiều cách để biểu diễn nội dung của đối tượng được khuyến nghị như biểu diễn ở dạng cây, biểu diễn ở dạng văn có cấu trúc và biểu diễn văn bản phi cấu trúc. Thông thường người ta chọn phương pháp biểu diễn nội dung của đối tượng dưới dạng một vector đặc trưng [18]. Phương pháp xây dựng vector đặc trưng sẽ được trình bày chi tiết ở mục tiếp theo.

**(\*\*) Phương pháp xây dựng vector đặc trưng**

Các vector đặc trưng có thể được xây dựng bằng nhiều cách khác nhau. Thông thường để xây dựng vector đặc trưng, các văn bản phi cấu trúc sẽ được tách thành *n* cụm từ không trùng lắp. Sau đó biểu diễn các cụm từ thành một vector với số chiều là *n.* [18] Giá trị của mỗi chiều sẽ bằng số lần mà cụm từ đó xuất hiện trong văn bản gốc. Tuy nhiên theo Dietmar Jannach thì việc biểu diễn bằng cách trên có một vài vấn đề cần lưu ý[18]:

* + Không phải tất cả các cụm từ có độ quan trọng giống như nhau.
  + Các văn bản dài dẫn đến tăng cơ hội trùng lắp với hồ sơ của người dùng.

Để khắc phục tình trạng trên, theo ông Dietmar Jannach [18] và một số tài liệu khác [21] [5] thì không nên dùng trực tiếp tần xuất mà cụm từ xuất hiện trong tài liệu được mô hình hóa mà nên dùng giá trị TF-IDF thay thế. Việc tính giá trị TF-IDF có thể sử dụng công thức sau:

**-**

Trong đó TF(i,j) chính là số lần cụm từ *i* xuất hiện trong văn bản *j,* IDF(i) chính là giá trị đo độ phổ biến của cụm từ *i* trong tất cả các văn bản. IDF(i) được tính theo công thức:

Trong đó *N* là số văn bản thô cần mô hình hóa. *n(i)* là số văn bản chứ cụm từ *i*.

**(\*\*\*) Phương pháp khuyến nghị**

Các bước để xây dựng hệ thống khuyến nghị dựa trên nội dung như sau:

* Bước 1: Biểu diễn nội dung của những đối tượng được khuyến nghị *i* dưới dạng một vector đặc trưng(feature vector) [22].
* Bước 2: Xây dựng và biểu diễn sở thích của người dùng *u* dưới dạng một vector đặc trưng.
* Bước 3: Tính toán độ tương tự cosine [19] của các đối tượng *i* thuộc tập đối đượng được khuyến nghị đối với người dùng *u*.
* Bước 4: Đưa ra *top N* những đối tượng có độ tương tự cao nhất.

Mã giả cho thuật toán:

Input: UserSet, JobSet, RatingSet

foreach(User in UserSet){

User = ModelingUserProfile(User)

}

foreach(Job in JobSet){

Job = ModelingItem(Job)

}

foreach(Job in JobSet){

JobVector = GetFeaturedVector(Job);

foreach(User in UserSet){

UserVector=GetFeaturedVector(User,GetUserRating(User, RatingSet));

Weight = Cosine(JobVector, UserVector);

If Weight is in TopN Then

AddToTopN(User,Job, Weight)

}

}

1. **Phương pháp tiếp cận lai**

Có rất nhiều phương pháp khuyến nghị khác nhau. Mỗi phương pháp có một điểm mạnh và điểm yếu riêng. Phương pháp lai ra đời nhằm tận dụng các điểm mạnh của các phương pháp khuyến nghị khác nhau và hạn chế các điểm yếu [9]. Phương pháp lai kết hợp kết quả khuyến nghị của các phương pháp khuyến nghị khác theo một cách nào đó nhằm đưa ra một hệ thống khuyến nghị tốt hơn [18]. Có nhiều các lai khác nhau. Cách tiếp cận lai đơn giản nhất là sử dụng phương pháp lai có trọng số (Weighted) [18] [17]. Mỗi kết quả đầu ra của các phương pháp *i* sẽ được kết hợp tuyến tính với nhau theo một hệ số *βi*. Các hệ số *βi* này sẽ dần được hiệu chỉnh cho đến khi thu được kết quả tốt nhất [9] [23].

Công thức dự đoán cho phương pháp lai có trọng số [18]



Mã giả thuật toán khuyến nghị dựa trên phương pháp lai đơn giản

Input: UserSet, JobSet, RatingSet, Alpha

CFResult = RunCF(RatingSet);

CBResult = RunCB(UserSet,JobSet, RatingSet);

foreach(Job in JobSet)

{

foreach(User in UserSet)

{

CFScore = GetScore(CFResult, User, Job);

CBScore = GetScore(CBResult, User, Job);

HBScore = CFScore \* Alpha + (1 - Alpha) \* CBScore;

If(HBScore in topN) Then

AddToTopN(User,Job, HBScore)

}

}

1. **Các phương pháp đánh giá**
2. **Phương pháp đánh giá offline**

Mục tiêu của đánh giá offline là để chọn ra được những thuật toán tốt nhất cho một hệ thống khuyến nghị. Khác với đánh giá online, đánh giá offline không cần một hệ thống chạy thực, đánh giá offline đưa ra giả thuyết “kết quả đánh giá offline tương đương với kết quả khi triển khai thực tế” tức là chúng ta mong đợi nếu kết quả đánh giá offline tốt thì kết quả thực tế cũng sẽ tốt. Để làm được điều này cần phải mô phỏng tốt quá trình đánh giá online – hệ thống khuyến nghị và người dùng đánh giá kết quả khuyến nghị đó. Thông thường để làm được điều này ta dựa vào dữ liệu đã đánh giá trong quá khứ của người dùng (ma trận đánh giá), ẩn giấu đi một phần dữ liệu và phần còn lại được sử dụng để huấn luyện hệ thống để hệ thống tiên đoán ra những dữ liệu đã bị giấu. Có nhiều cách để chia tập dữ liệu phần nào sẽ dùng để đánh giá, phần nào sẽ dùng để huấn luyện, tuy nhiên phổ biến nhất là phương pháp K-fold Cross-validation (tạm dịch: đánh giá chéo với k lần lặp).

**K-fold Cross-validation** [8]**:** Cross-validation là phương pháp thiết lập đánh giá kết quả thực nghiệm của các bài toán phân tích thống kê. Mục tiêu của cross-validation là để xác định cách lấy tập dữ liệu kiểm tra (GroundTruth) và tập dữ liệu huấn luyện (TrainingSet) để tránh vấn đề về “Overfiting” [9], giúp người dùng có thể nhận thấy sự độc lập giữa dữ liệu và mô hình huấn luyện.

K-fold Cross-validation được thiết lập như sau: với một tập dữ liệu đầu vào, hệ thống sẽ chia điều ngẫu nhiên tập dữ liệu đầu vào thành **k** phần với kích thước bằng nhau, lấy một phần làm dữ liệu kiểm tra (tập GroundTruth) và **k-1** phần còn lại được sử dụng để huấn luyện mô hình. Quá trình này lặp lại **k** lần với các tập GroundTruth khác nhau.



Hình 2. K-fold Cross-validation với k=4

1. Phương pháp đánh giá online

Với thiết lập đánh giá online hệ thống sẽ đo lường được sự thay đổi hành vi của người dùng khi họ tương tác với hệ thống khuyến nghị. Ưu điểm của đánh giá online là có thể phản ánh đúng được hiệu quả của hệ thống khuyến nghị. Tuy nhiên tồn tại song song với ưu điểm này là nhiều nhược điểm như: để đánh giá online phải có một hệ thống được thiết lập chạy thật sự; phải xem xét nhiều yếu tố để có thể thay đổi các thuật toán để đánh giá; với hệ thống chạy thật sự thì có khả năng sẽ mang lại cho người dùng những trải nghiệm tiêu cực (vì hệ thống chưa qua đánh giá có thể sẽ không phải là hệ thống tốt); và một điều quan trọng nữa là phải tốn kém nhiều chi phí để thiết lập cho một hệ thống đánh giá online. Chính vì những nhược điểm trên, trong thực tế thường sử dụng các phương pháp đánh giá offline và khóa luận cũng tập trung vào các lý thuyết và thực nghiệm theo phương pháp đánh giá offline.

5 Thực nghiệm và đánh giá

1. **Dataset**

Để tiến hành thực nghiệm so sánh đánh giá các thuật toán, chúng tôi đã tiến hành xây dựng bộ dữ liệu dataset cho thực nghiệm này. Việc xây dựng dataset bắt đầu từ việc thu thập dữ liệu việc làm, chúng tôi xây dựng tool crawler để lấy dữ liệu việc làm được đăng tuyển tại các website: vietnamworks.com, itviec.com và careerlink.vn. Số dữ liệu việc làm thu thập được gồm có 4475 việc làm đăng tuyển bằng ngôn ngữ tiếng Việt, trên tổng số 20 ngành nghề khác nhau. Dữ liệu việc làm thu thập được đưa lên website <http://sdlab.uit.edu.vn/jrs> để tiến hành khảo sát người dùng, số người dùng tham gia khảo sát là 59 người, là các bạn sinh viên từ các trường Đại học Phú Yên, Đại học Công Nghệ Thông Tin. Người dùng tham gia vào khảo sát sẽ tiến hành đánh giá các việc làm về mức độ phù hợp của từng việc làm đối với họ, mức độ phù hợp sẽ được chia theo thang điểm từ 1 đến 5 tương ứng tăng dần theo mức độ phù hợp. Tổng số các đánh giá của người dùng là 8576. Ngoài việc đánh giá các việc làm, người dùng còn cung cấp 1 bản hồ sơ việc làm (CV) cho hệ thống. Bộ dataset này được chúng tôi cung cấp tại: <http://sdlab.uit.edu.vn/jrs/dataset?dsname=jrs_dataset.zip>.

1. **Kết quả thực nghiệm**

Để tiến hành đánh giá so sánh các thuật toán, chúng tôi đã tiến hành 2 phương pháp đánh giá gồm đánh giá offline và đánh giá online. Chúng tối tiến hành đánh giá trên các thuật toán CF với các độ tương tự khác nhau, thuật toán CB và Hybrid.

Với đánh giá offline chúng tối đã cài đặt và sử dụng các độ đo phổ biến sau: độ đo về chất lượng của tiên đoán như: RMSE (Root Mean Square Error); nhóm các độ đo về chất lượng danh sách khuyến nghị: Precision, Recall, F-Measure; nhóm các độ đo về chất lượng danh sách xếp hạng các kết quả khuyến nghị: NDCG (Normalize Discounted Cumulative Gain), MAP (Mean Average Precision) hay MRR (Mean Reciprocal Rank).

Dưới đây là kết quả tốt nhất cho các thuật toán, CF với phương pháp User-base với độ tương tự Euclidance Distance là tốt nhất trong các phương pháp tiếp cận CF, Hybrid lai giữa CB và CF theo hệ số alpha = 0.8 là tốt nhất trong số các kết quả lai.

Bảng 1. Kết quả đánh giá các thuật toán với độ đo Precision, Recall và F-Measure

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Độ  Đo  Thuật  toán | Precision | | | Recall | | | F-Measure | | |
| Top5 | Top10 | Top15 | Top5 | Top10 | Top15 | Top5 | Top10 | Top15 |
| CF | 0.013 | 0.013 | 0.008 | 0.021 | 0.044 | 0.047 | 0.014 | 0.018 | 0.012 |
| CB | 0.008 | 0.007 | 0.005 | 0.017 | 0.03 | 0.034 | 0.01 | 0.01 | 0.008 |
| Hybrid | 0.011 | 0.007 | 0.005 | 0.026 | 0.038 | 0.037 | 0.013 | 0.011 | 0.009 |

Bảng 2. Kết quả đánh giá các thuật toán với độ đo RMSE

|  |  |  |  |
| --- | --- | --- | --- |
| Độ đo  Thuật toán | RMSE | | |
| Top5 | Top10 | Top15 |
| CF - User based - Euclidean Distance | 0.031 | 0.047 | 0.067 |
| CB | 0.025 | 0.04 | 0.046 |
| Hybrid | 0.189 | 0.248 | 0.262 |

Bảng 3. Kết quả đánh giá các thuật toán với độ đo NDCG, MRR và MAP

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Đô  đo  Thuật  toán | NDCG | | | MRR | | | MAP | | |
| Top5 | Top10 | Top15 | Top5 | Top10 | Top15 | Top5 | Top10 | Top15 |
| CF | 0.014 | 0.013 | 0.008 | 0.033 | 0.036 | 0.022 | 0.032 | 0.036 | 0.022 |
| CB | 0.008 | 0.007 | 0.005 | 0.015 | 0.019 | 0.017 | 0.016 | 0.019 | 0.016 |
| Hybrid | 0.011 | 0.007 | 0.006 | 0.026 | 0.020 | 0.023 | 0.026 | 0.020 | 0.023 |

Với đánh giá online chúng tôi đã sử dụng các độ đo: Precision để đo tỉ lệ chính xác của các khuyến nghị dựa trên phản hồi của người dùng; sừ dụng độ đo NDCG, MAP, và MRR để đo chất lượng ranklist (danh sách khuyến nghị). Chúng tối tiến hành đánh giá online bằng cách chọn ra 3 thuật toán CF, CB và Hybird có kết quả tốt nhất từ kết quả đánh giá offline, tiến hành chạy khuyến nghị ra danh sách Top15 các việc làm và cho người dùng đánh giá các kết quả khuyến nghị. Kết quả đạt được của đánh giá online như sau:

Bảng 4. Kết quả đánh giá online

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Độ đo  Thuật toán | Precision | NDCG | RMSE | MRR | MAP |
| CF | 0.081 | 0.148 | 0.687 | 0.593 | 0.589 |
| Hybrid | 0.261 | 0.373 | 2.445 | 0.876 | 0.773 |
| CB | 0.372 | 0.471 | 1.39 | 0.844 | 0.773 |

1. **Đánh giá kết quả thực nghiệm**

* Với kết quả thực nghiệm đánh giá offline ở trên có thể dễ thấy rằng kết quả thực nghiệm đạt kết quả quá thấp cho tất cả các thuật toán. Nguyên nhân dẫn đến kết quả này có thể lý giải như sau:
  + Bộ dữ liệu gán nhãn quá thưa.
  + Bộ dữ liệu gán nhãn thưa có nghĩa dữ liệu thực nghiệm không thể bao phủ hết các trường hợp (tức bộ dữ liệu không đặc tả được sở thích của người dùng trên tất cả các việc làm có trong bộ dữ liệu), như vậy khi hệ thống khuyến nghị khuyến nghị một việc làm chưa được đánh giá bởi người dùng thì hệ thống đánh giá sẽ không thể xác định được đó là một khuyến nghị tốt hay không (ở đây mặc định là không tốt vì vậy kết quả đánh giá hệ khuyến nghị là rất thấp).
* Đánh giá online cho kết quả phản ánh chính xác được chất lượng của các phương pháp khuyến nghị bởi vì đó là kết quả do chính người dùng đánh giá. Với kết quả trên cho thấy trong số 3 phương pháp thì tiếp cận nội dung (content-based) là phương pháp hiệu quả nhất, tiếp cận lọc cộng tác (collaborative filtering) là phương pháp kém hiệu quả nhất. Nguyên nhân:
  + Tiếp cận lọc cộng tác dựa trên dữ liệu gán nhãn để khuyến nghị, nhưng bộ dữ liệu gán nhãn quá thưa dẫn đến không thể xác định chính xác người dùng có đồng sở thích hay những việc làm giống nhau để khuyến nghị.
  + Kết quả của tiếp cận nội dung phụ thuộc vào mức độ khớp giữa hồ sơ xin việc của người dùng và nội dung đăng tuyển của việc làm, không bị ảnh hưởng bởi ma trận đánh giá nên kết quả tốt hơn so với lọc cộng tác. Tuy nhiên kết quả này vẫn chưa thực sự cao, nguyên nhân bởi phương pháp tiếp cận nội dung được cài đặt theo hương tiếp cận đơn giản, không tiến hành các xử lý phức tạp.
  + Phương pháp lai là sự kết hợp giữa lọc cộng tác và tiếp cận nội dung, vì vậy kết quả của nó nằm giữa 2 phương pháp trên là hoàn toàn đúng.
* Từ những kết quả thực nghiệm trên chúng tôi rút ra được những nhận định sau để cho một thực nghiệm đạt kết quả tốt:
  + Chuẩn bị một dataset tốt, dataset tốt sẽ là một dataset đủ lớn về số lượng việc làm cũng như số lượng người dùng.
  + Bộ dữ liệu sử dụng để thực nghiệm đánh giá cần phải phủ hết tất cả các trường hợp (tức mỗi người dùng phải đánh giá hết tất cả các việc làm)
  + Nghiên cứu khắc phục vấn đề về ma trận đánh giá thưa
  + Cài đặt nhiều thuật toán khuyến nghị khác nhau, nâng cao các thuật toán xử lý để có kết quả tốt hơn.

6 Kết luận

Dựa vào các kết quả thực nghiệm đã thực hiện, chúng tôi có nhận định rằng thuật toán Content-based là thuật toán tốt nhất, tuy nhiên điều này sẽ còn phụ nhiều vào tập dữ liệu thực nghiệm, cần tiến hành nhiều thực nghiệm khác với những dữ liệu khác để có một kết quả đánh giá chính xác nhất.

Tài liệu tham khảo

|  |  |
| --- | --- |
| [1] | “Vietnamworks,” [Trực tuyến]. Available: http://www.vietnamworks.com/tim-viec-lam/tat-ca-viec-lam. [Đã truy cập 22 10 2016]. |
| [2] | “Career Builder,” [Trực tuyến]. Available: http://careerbuilder.vn/viec-lam/tat-ca-viec-lam-vi.html. [Đã truy cập 22 10 2016]. |
| [3] | M. G. A. M. T. M. P. N. D. a. S. M. Claypool, "Combining content-based and collaborative filters in an online newspaper," in *Combining content-based and collaborative filters in an online newspaper*, 1999. |
| [4] | H. N. Tín, “Phát triển một số phương pháp khuyến nghị hỗ trợ tìm kiếm thông tin học thuật dựa trên tiếp cận phân tích mạng xã hội,” Hồ Chí Minh, 2016. |
| [5] | Dietmar Jannach, Markus Zanker, Alexander Felfernig, Gerhard Friedrich, Recommender Systems – An Introduction, Cambridge University Press, 2011. |
| [6] | Sean Owen, Robin Anil, Ted Dunning, Ellen Friedman, Mahout In Action, United States of America, 2012. |
| [7] | "Mahout," , . [Online]. Available: http://en.wikipedia.org/wiki/Mahout. [Accessed 22 10 2016]. |
| [8] | "Cross-validation (statistics)," , . [Online]. Available: http://en.wikipedia.org/wiki/Cross-validation\_(statistics). [Accessed 25 10 2016]. |
| [9] | "Wikipedia - Overfiting," [Online]. Available: https://en.wikipedia.org/wiki/Overfitting. [Accessed 13 11 2016]. |
| [10] | "Wikipedia - Recommender system," [Online]. Available: https://en.wikipedia.org/wiki/Recommender\_system. [Accessed 22 10 2016]. |
| [11] | M. Z. A. F. G. F. Dietmar Jannach, "Recommender Systems – An Introduction," London, England, 2014. |
| [12] | D. B. Michael J. Pazzani, "Content-Based Recommendation Systems," Spinger, Palo Alto, 2007. |
| [13] | "Cosine similarity - Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Cosine\_similarity. [Accessed 22 10 2016]. |
| [14] | "Euclidean distance wikipedia," [Online]. Available: http://en.wikipedia.org/wiki/Euclidean\_distance. [Accessed 21 10 2016]. |
| [15] | "Tanimoto Similarity - Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Jaccard\_index. [Accessed 22 10 2016]. |
| [16] | "Content-based Recommender Systems: State of the arts and trends," Pasquale Lops, Marco de Gemmis and Giovanni Semeraro, [Online]. Available: http://facweb.cs.depaul.edu/mobasher/classes/ect584/Papers/ContentBasedRS.pdf. [Accessed 25 10 2016]. |
| [17] | Francesco Ricci, Lior Rokach, Bracha Shapira, Paul B. Kantor, Recommender Systems Handbook, Springer New York Dordrecht Heidelberg London, 2010. |
| [18] | "Feature vector - wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Feature\_vector. [Accessed 22 10 2016]. |
| [19] | "Pearson product-moment correlation coefficient," , . [Online]. Available: http://en.wikipedia.org/wiki/Pearson\_product-moment\_correlation\_coefficient. [Accessed 21 10 2016]. |
| [20] | "Jaccard index," , . [Online]. Available: http://en.wikipedia.org/wiki/Jaccard\_index. [Accessed 21 10 2016]. |
| [21] | "Euclidean distance Cloudera," 2013. [Online]. Available: http://archive-primary.cloudera.com/cdh4/cdh/4/mahout-0.7-cdh4.3.2/mahout-core/org/apache/mahout/cf/taste/impl/similarity/EuclideanDistanceSimilarity.html. [Accessed 21 10 2016]. |
| [22] | "LogLikelihood similariy - tdunning.blogspot," [Online]. Available: http://tdunning.blogspot.com/2008/03/surprise-and-coincidence.html. [Accessed 21 10 2016]. |