BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

ISO 9001:2008

TRỊNH KHẮC DŨNG

LUẬN VĂN THẠC SĨ

NGÀNH HỆ THỐNG THÔNG TIN

Hải Phòng - 2016

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

TRỊNH KHẮC DŨNG

HỌC BÁN GIÁM SÁT TRÊN ĐỒ THỊ VỚI ỨNG DỤNG TRA CỨU ẢNH

LUẬN VĂN THẠC SĨ
NGÀNH CÔNG NGHỆ THÔNG TIN

CHUYÊN NGÀNH: HỆ THỐNG THÔNG TIN
MÃ SỐ: 60 48 01 04

NGƯỜI HƯỚNG DẪN KHOA HỌC:
PGS.TS. Ngô Quốc Tạo

MỤC LỤC

LỜI CẢM ƠN

Trước tiên, em xin bày tỏ lòng biết ơn sâu sắc tới PGS.TS Ngô Quốc Tạo, Viện Công nghệ Thông tin thuộc Viện Khoa học và Công nghệ Việt Nam là cán bộ trực tiếp hướng dẫn khoa học cho em trong quá trình thực hiện luận văn này.

Em xin chân thành cảm ơn Th.S Ngô Trường Giang, Khoa Công nghệ thông tin trường Đại học Dân lập Hải Phòng đã tận tình giúp đỡ và đóng góp những ý kiến quý báu trong suốt quá trình hoàn thành luận văn.

Xin trân trọng cảm ơn các Thầy cô Trường Đại Học Dân Lập Hải Phòng, đặc biệt là các Thầy Cô trong khoa Công Nghệ Thông Tin của nhà trường đã tận tình chỉ bảo, giúp đỡ em trong suốt thời gian học tập tại nhà trường cũng như trong quá trình làm luận văn.

Bên cạnh đó, để hoàn thành luận văn này, em cũng đã nhận được rất nhiều sự giúp đỡ, những lời động viên quý báu của các bạn bè, gia đình và đồng nghiệp. Em xin chân thành cảm ơn.

Tuy nhiên, do thời gian hạn hẹp, mặc dù đã nỗ lực hết sức mình, nhưng chắc rằng luận văn khó tránh khỏi thiếu sót. Em rất mong nhận được sự thông cảm và chỉ bảo tận tình của quý thầy cô và các bạn.

Hải Phòng, ngày tháng năm 2016

HỌC VIÊN

Trịnh Khắc Dũng

LỜI CAM ĐOAN

Tôi xin cam đoan kết quả đạt được trong luận văn là sản phẩm nghiên cứu của bản thân được thực hiện dưới sự hướng dẫn của Giáo viên hướng dẫn khoa học.

Trong toàn bộ nội dung của luận văn, những điều được trình bày hoặc là của cá nhân tôi, hoặc là được tổng hợp từ nhiều nguồn tài liệu. Tất cả các tài liệu tham khảo đều được trích dẫn hợp pháp và xuất xứ rõ ràng.

Tôi xin chịu hoàn toàn trách nhiệm với những nội dung viết trong luận văn này.

Hải Phòng, ngày tháng năm 2016

HỌC VIÊN

Trịnh Khắc Dũng

DANH MỤC CHỮ VIẾT TẮT

Stt	Từ viết tắt	Mô tả
1	ARE	Augmented Relation Embedding
2	CBIR	Content-Based Image Retrieval
3	CRF	Conditional Random Field
4	EMR	Efficient Manifold Ranking
5	*k*-NN	k-Nearest Neighbor
6	LGRM	Local and Global Regressive Mapping
7	MDP	Markov Decision Process
8	MR	Manifold Ranking
9	MRBIR	Manifold Ranking Based Image Retrieval
10	MRF	Markov Random Field
11	RF	Relevance Feedback
12	SVM	Support Vector Machine
13	TSVM	Transductive Support Vector Machine

DANH MỤC HÌNH VẼ

DANH MỤC BẢNG BIỂU

MỞ ĐẦU

Trong tra cứu ảnh dựa trên nội dung, các đặc trưng được trích chọn một cách tự động bằng cách sử dụng kỹ thuật của thị giác máy chủ yếu là các đặc trưng mức thấp thấp (màu, kết cấu, hình dạng, vị trí không gian…)[4]. Mặc dù nhiều thuật toán phức tạp đã được thiết kế để mô tả màu sắc, hình dáng và đặc trưng kết cấu, nhưng các thuật toán này vẫn không thể phản ánh thỏa đáng ngữ nghĩa ảnh. Do vậy, khoảng cách ngữ nghĩa giữa các đặc trưng mức thấp và các khái niệm mức cao vẫn còn lớn nên hiệu suất của CBIR là vẫn còn xa với mong đợi của người dùng [9].

Để thu hẹp khoảng cách ngữ nghĩa, phản hồi liên quan (Relevance Feedback - RF) được xem như là một công cụ hiệu quả để cải thiện hiệu năng của hệ thống CBIR [8], [1]. Gần đây, rất nhiều nhà nghiên cứu bắt đầu xem phản hồi liên quan như là bài toán phân lớp hoặc bài toán học. Người sử dụng sẽ cung cấp các mẫu dương hoặc mẫu âm và hệ thống sẽ học từ các mẫu này để phân tách tất cả dữ liệu thành nhóm liên quan và không liên quan. Do vậy, rất nhiều phương pháp học máy có thể được áp dụng. Những phương pháp học có thể được phân thành hai lớp: Quy nạp và Truyền dẫn tùy theo dữ liệu không được gán nhãn có được dùng trong chiến lược huấn luyện hay không.

Những phương pháp quy nạp chủ yếu dựa trên Support Vector Machines [10], [7], boosting [6] và mạng neuron [11]. Chúng được xem như là giải quyết bài toán phân lớp nhị phân (liên quan và không liên quan) và xếp hạng ảnh theo kết quả phân lớp.

Trong các phương pháp truyền dẫn, các ảnh trong cơ sở dữ liệu được biểu diễn như là các đỉnh của đồ thị có trọng số. Phản hồi liên quan của người dùng được sử dụng để tạo ra các mẫu được gán nhãn. Những mẫu này sẽ được sử dụng để làm cơ sở tính toán khả năng truyền dẫn cho mỗi ảnh [5],[12]. Các phương pháp này không chỉ sử dụng mối quan hệ từng cặp giữa ảnh truy vấn

với các ảnh trong cơ sở dữ liệu mà nó còn khai thác cả mối quan hệ giữa tất cả các ảnh với nhau, nhờ vậy, hiệu quả tra cứu của chúng được cải thiện đáng kể.

Trong giai đoạn đầu của quá trình tra cứu ảnh với phản hồi liên quan, số ảnh được gán nhãn thường rất ít trong khi số lượng ảnh chưa được gán nhãn rất nhiều. Do vậy lựa chọn phương pháp học hiệu quả để tận dụng được lợi thế của thông tin đầu vào là vấn đề quan trọng. Đó cũng là lý do mà tôi chọn đề tài: ***"Học bán giám sát trên đồ thị với ứng dụng tra cứu ảnh".***

Nội dung luận văn gồm 3 chương:

Chương 1: Khái quát về CBIR và học trên đồ thị

Chương này trình bày tổng quan tra cứu ảnh dựa trên nội dung; tra cứu ảnh dựa trên nội dung với phản hồi liên quan; các phương pháp học máy và học trên đồ thị gồm có các mô hình Học có giám sát (Supervised learning), Học không giám sát (Unsupervised learning), Học bán giám sát (Semi-Supervised learning).

Chương 2: Tra cứu ảnh dựa trên xếp hạng đa tạp

Tập trung tìm hiểu phương pháp học bán giám sát trên đồ thị qua thuật toán lan truyền nhãn. Đồng thời tập trung nghiên cứu phương pháp tra cứu ảnh dựa trên xếp hạng đa tạp và cải tiến phương pháp này khi áp dụng vào tra cứu dữ liệu ảnh có số lượng lớn.

Chương 3: Thực nghiệm

Cài đặt thử nghiệm chương trình tra cứu ảnh dựa trên nội dung theo mô hình học bán giám sát trên đồ thị qua thuật toán xếp hạng đa tạp (MR) và thuật toán xếp hạng đa tạp cải tiến (EMR). So sánh hiệu năng của hai thuật toán này.

CHƯƠNG 1: KHÁI QUÁT VỀ CBIR VÀ HỌC TRÊN ĐỒ THỊ

1.1 Tra cứu ảnh dựa trên nội dung với phản hồi liên quan

1.1.1 Giới thiệu

Hệ thống tra cứu ảnh dựa trên nội dung (Content Based Image Retrieval - CBIR) là một công cụ mạnh vì nó tra cứu ảnh trong cơ sở dữ liệu ảnh bằng việc sử dụng dấu hiệu trực quan. Các hệ thống tra cứu ảnh dựa trên nội dung trích rút các đặc trưng từ bản thân các ảnh và tính toán độ đo tương tự giữa ảnh truy vấn và các ảnh cơ sở dữ liệu dựa trên các đặc trưng này. Tra cứu ảnh dựa trên nội dung trở nên rất phổ biến do nhu cầu tra cứu ảnh trong các cơ sở dữ liệu lớn tăng nhanh. Bởi vì tốc độ và độ chính xác là quan trọng, việc tiếp tục phát triển các hệ thống tra cứu ảnh đảm bảo độ chính xác và có tốc độ nhanh là cần thiết. Tra cứu ảnh dựa trên nội dung ứng dụng vào vào rất nhiều công việc hữu ích như: tìm các ảnh phong cảnh trên Internet, điều tra hình sự dựa vào vân tay và dấu chân, chuẩn đoán bệnh trong y tế, sử dụng trong các hệ thống thông tin địa lý và viễn thám, sử dụng cho tra cứu các phần video như phim và trò chơi, các ứng dụng khác bao gồm bảo tàng trực tuyến, quảng cáo...

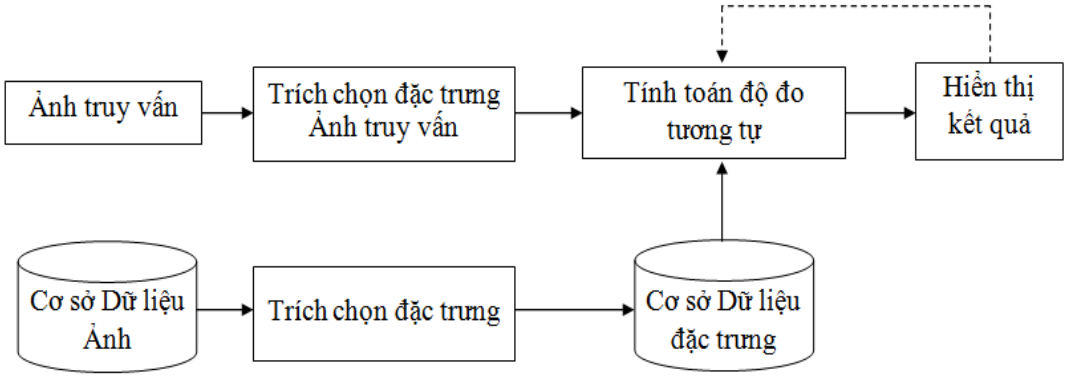
Những thành phần của một hệ thống tra cứu ảnh dựa trên nội dung:

Một hệ thống tra cứu ảnh đòi hỏi các thành phần như trong Hình 1-1. Trong đó có ba thành phần quan trọng nhất trong tra cứu ảnh dựa trên nội dung: trích chọn đặc trưng, đánh chỉ số và giao diện truy vấn cho người dùng.

Các bước tra cứu ảnh trong CBIR thường bao gồm :

- Tiếp nhận truy vấn của người dùng (dưới dạng ảnh hoặc phác thảo).
- Trích chọn đặc trưng của truy vấn và lưu trữ vào cơ sở dữ liệu đặc trưng như là một vector hoặc không gian đặc trưng.
- So sánh độ tương tự giữa các đặc trưng trong cơ sở dữ liệu với nhau từng đôi một.

- Lập chỉ mục cho các vector để nâng hiệu quả tra cứu.
- Trả lại kết quả tra cứu cho người dùng.



Hình 1-1: *Kiến trúc tổng quan về hệ thống tra cứu ảnh dựa trên nội dung*

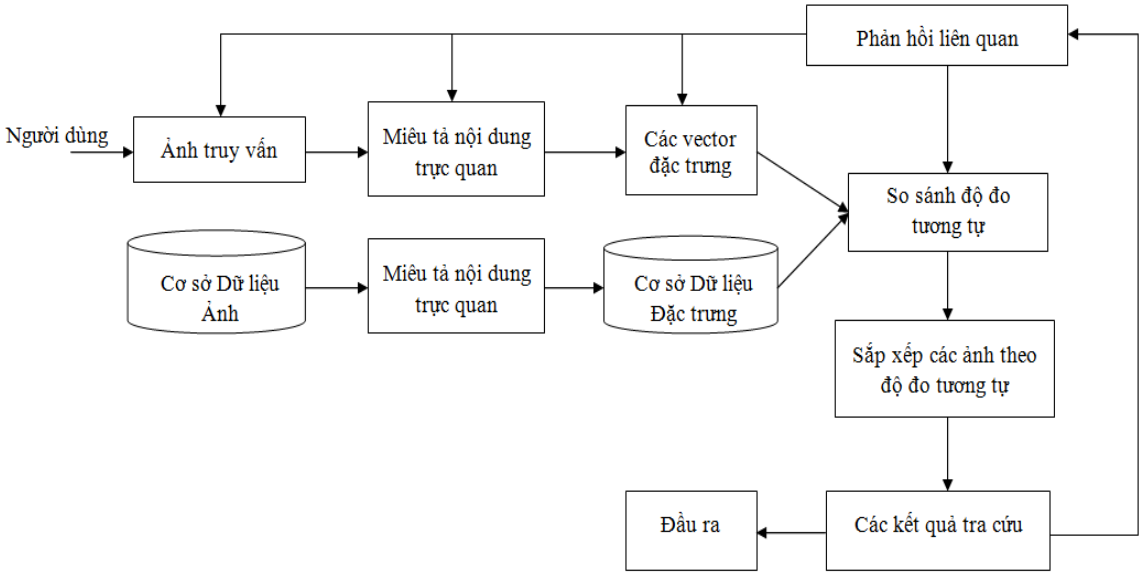
1.1.2 Kiến trúc tổng quan của hệ thống CBIR với phản hồi liên quan

1.1.2.1 Khái niệm phản hồi liên quan

Phản hồi liên quan là một kỹ thuật sửa đổi truy vấn, nó bắt nguồn trong thông tin tra cứu và qua đó sẽ tập hợp lại những đặc trưng tra cứu chính xác từ phía người dùng bằng việc lặp đi lặp lại việc phản hồi, sau đó hệ thống sẽ lọc ra thông tin chính xác. Phản hồi liên quan có thể được coi là một mô hình tìm kiếm thay thế, bổ sung cho những mô hình khác như tìm kiếm dựa trên từ khóa. Trong trường hợp không có một khuôn khổ đáng tin cậy để mô hình hóa ngữ nghĩa ảnh mức cao và nhận thức chủ quan thì phản hồi liên quan sẽ là một phương thức để tìm hiểu các trường hợp cụ thể của ngữ nghĩa truy vấn. Để giải quyết những vấn đề này, tương tác phản hồi liên quan, một kỹ thuật trong hệ thống tìm kiếm thông tin dựa trên văn bản truyền thống, đã được giới thiệu. Với phản hồi liên quan, có thể thiết lập liên kết giữa các khái niệm mức cao và đặc trưng mức thấp. Ý tưởng chính là sử dụng các mẫu dương và mẫu âm từ người sử dụng để cải thiện hiệu suất hệ thống. Đối với một truy vấn nhất định, đầu tiên hệ thống sẽ trả về một danh sách các hình ảnh được xếp theo một độ tương tự xác định trước. Sau đó, người dùng đánh dấu những hình ảnh có liên quan đến truy vấn (mẫu dương) hoặc không có liên quan

(mẫu âm). Hệ thống sẽ chọn lọc kết quả tra cứu dựa trên những phản hồi và trình bày một danh sách mới của hình ảnh cho người dùng. Do đó, vấn đề quan trọng trong phản hồi liên quan là làm thế nào để kết hợp các mẫu dương và mẫu âm để tinh chỉnh các truy vấn và/hoặc điều chỉnh các biện pháp tương tự.

1.1.2.2 Kiến trúc tổng quan của hệ thống CBIR với RF



Hình 1-2: *Mô hình tổng quát hệ thống CBIR với phản hồi liên quan*

Ý tưởng chính của phản hồi liên quan là chuyển trách nhiệm tìm kiếm xây dựng truy vấn đúng từ người dùng sang hệ thống. Để thực hiện điều này một cách đúng đắn, người dùng phải cung cấp cho hệ thống một số thông tin, để hệ thống có thể thực hiện tốt việc trả lời truy vấn ban đầu. Việc tìm kiếm ảnh thường dựa trên sự tương tự hơn là so sánh chính xác, kết quả tra cứu sẽ được đưa ra cho người dùng. Sau đó, người dùng đưa ra các thông tin phản hồi trong một bản mẫu "Các quyết định liên quan" thể hiện thông qua kết quả tra cứu. "Quyết định liên quan" đánh giá kết quả dựa trên ba giá trị. Ba giá trị đó là: liên quan, không liên quan, và không quan tâm. "Liên quan" nghĩa là ảnh có liên quan đến truy vấn của người dùng. "Không liên quan" có nghĩa là ảnh không có liên quan đến truy vấn người dùng. Còn "không quan tâm"

nghĩa là người dùng không cho biết bất kỳ điều gì về ảnh. Nếu phản hồi của người dùng là có liên quan, thì vòng lặp phản hồi sẽ tiếp tục hoạt động cho đến khi người dùng hài lòng với kết quả tra cứu. Như Hình 1-2 mô tả cấu trúc của hệ thống phản hồi liên quan. Trong hệ thống đó có các khối chính là: cơ sở dữ liệu ảnh, trích chọn đặc trưng, đo độ tương tự, phản hồi từ người dùng và thuật toán phản hồi.

1.1.2.3 Trích chọn đặc trưng

Trích chọn đặc trưng liên quan đến việc trích chọn các thông tin có ý nghĩa từ ảnh. Vì vậy, nó làm giảm việc lưu trữ cần thiết, do đó hệ thống sẽ trở nên nhanh hơn và hiệu quả trong CBIR. Khi đặc trưng đươc trích chọn, chúng sẽ đươc lưu trữ trong cơ sở dữ liệu để sử dụng trong lần truy vấn sau này. Mức độ mà một máy tính có thể trích chọn thông tin có ích từ ảnh là vấn đề then chốt nhất cho sự tiến bộ của hệ thống diễn giải hình ảnh thông minh. Một trong những ưu điểm lớn nhất của trích chọn đặc trưng là nó làm giảm đáng kể các thông tin (so với ảnh gốc) để biểu diễn một ảnh cho việc hiểu nội dung của ảnh đó. Hiện nay đã có rất nhiều nghiên cứu lớn về các phương pháp tiếp cận khác nhau để phát hiện nhiều loại đặc trưng trong ảnh. Những đặc trưng này có thể đươc phân loại như là đặc trưng toàn cục và đặc trưng cục bộ. Các đặc trưng phổ biến nhất mà đươc sử dụng là màu sắc, kết cấu và hình dạng.

- Đặc trưng toàn cục: Đặc trưng toàn cục phải được tính toán trên toàn bộ ảnh. Ví dụ, mức độ màu xám trung bình, biểu đồ về cường độ hình dạng, … Ưu điểm của việc trích chọn toàn cục là tốc độ nhanh chóng trong cả trích chọn đặc trưng và tính toán độ tương tự. Tuy nhiên, chúng có thể quá nhạy cảm với vị trí và do đó không xác định đươc các đặc tính trực quan quan trọng. Để tăng cường sự vững mạnh trong biến đổi không gian, chúng ta có thể tìm hiểu trích chọn đặc trưng cục bộ.
- Đặc trưng cục bộ: Trong đặc trưng toàn cục, các đặc trưng đươc tính toán trên toàn bộ ảnh. Tuy nhiên, đặc trưng toàn cục không thể nắm bắt

tất cả các vùng ảnh có đặc điểm khác nhau. Do đó, việc trích chọn các đặc trưng cục bộcủa ảnh là cần thiết. Các đặc trưng đó có thể đươc tính toán trên các kết quả của phân đoạn ảnh và thuật toán phát hiện biên. Vì thế, tất cả chúng đều dựa trên một phần của ảnh với một số tính chất đặc biệt.

- Điểm nổi bật: Trong việc tính toán đặc trưng cục bộ, việc trích chọn đặc trưng ảnh bị giới hạn trong một tập nhỏ các điểm ảnh, đó là những điểm chú ý. Tập các điểm chú ý đươc gọi là những điểm nổi bật. Những điểm nổi bật là những điểm có dao động lớn trong đặc trưng của vùng lân cận điểm ảnh. Nhiều hệ thống CBIR trích chọn những điểm nổi bật. Năm 2004, Rouhollah và các cộng sự đã định nghĩa điểm nổi bật có mặt trong tra cứu ảnh dựa trên nội dung như là một nhiệm vụ của CBIR, nơi mà người dùng chỉ quan tâm đến một phần của ảnh và phần còn lại là không liên quan. Ví dụ, chúng ta có thể tham khảo một số đặc trưng cục bộ như là ảnh gốc, đường tròn, đường nét, texel (các phần tử tập trung ở một khu vực kết cấu), hoặc các đặc trưng cục bộ khác, hình dạng của đường nét…

1.1.2.4 Độ đo tương tự

Trong độ đo tương tự, vector đặc trưng của ảnh truy vấn và vector đặc trưng của ảnh trong cơ sở dữ liệu được đối sánh bằng cách sử dụng một thước đo khoảng cách. Các hình ảnh được xếp hạng dựa trên giá trị khoảng cách. Vào năm 2003, Manesh và các cộng sự đã đề xuất phương pháp đo độ tương tự cho việc đối sánh chi tiết các độ đo khác nhau như: Manhattan, weighted mean-variance, Euclidean, Chebychev, Mahanobis, v.v… cho tra cứu kết cấu ảnh với đánh giá thực nghiệm. Họ nhận thấy rằng số liệu khoảng cách Canberra and Bray-Curtis thực hiện tốt hơn các số liệu khoảng cách khác.

1.1.2.5 Phản hồi từ người dùng

Sau khi có kết quả tra cứu, người dùng cung cấp phản hồi về các kết quả liên quan hoặc không liên quan. Nếu kết quả chưa được chấp nhận thì vòng lặp phản hồi sẽ được lặp lại nhiều lần cho đến khi người dùng hài lòng.

1.1.3 Các kỹ thuật phản hồi liên quan

1.1.3.1 Kỹ thuật dựa trên "học"

Kỹ thuật này dựa trên thông tin phản hồi có liên quan đến người dùng, phương pháp này đường được sử dụng một cách thích hợp để thay đổi các đặc trưng hoặc trong kỹ thuật so sánh độ tương tự. Tuy nhiên trong thực tế, kết quả của phản hồi liên quan người dùng chỉ là một số nhỏ của những ảnh được dán nhãn có liên quan đến khái niệm mức cao. Công nghệ học máy đã được nghiên cứu để giải quyết vấn đề này cũng như những vấn đề đáng quan tâm khác của phản hồi liên quan người dùng. Như là mô hình học một lớp (one - class learning), mô hình học tích cực (Active learning), mô hình học đa dạng (manifold learning). Để giải quyết các vấn đề của việc học từ các tập hợp học như vậy, các nhà nghiên cứu đã đề xuất thuật toán phân biệt EM, thuật toán này sử dụng các hình ảnh không có nhãn trong cơ sở dữ liệu cho việc lựa chọn các tính năng phân biệt tốt hơn.

1.1.3.2 Phản hồi đặc điểm kỹ thuật tiến bộ

Theo truyền thống, phản hồi liên quan đã tiếp nhận thông tin từ phía người dùng qua nhiều vòng phản hồi, mỗi vòng gồm một tập hợp các ví dụ tích cực và tiêu cực liên quan đến truy vấn dự định. Tuy nhiên, các nghiên cứu mới đây đã giới thiệu đến các mô hình tiến bộ kĩ thuật khác trực quan hơn và hiệu quả hơn. Thông tin phản hồi trực tiếp dựa trên một ảnh đặc trưng ngữ nghĩa thích hợp được gọi là phản hồi ngữ nghĩa. Một kĩ thuật khác đó là phản hồi chào mời, vấn đề của kĩ thuật này là nó sẽ tạo ra nhiều vòng phản hồi để kiểm tra sự kiên nhẫn của người dùng. Đề giải quyết vấn đề trên, những log của người dùng đã phản hồi trước đó có thể được sử dụng trong truy vấn sàng

lọc do đó làm giảm số lần người dùng phản hồi trong phản hồi liên quan, kĩ thuật này đã được Hoi và Lyu nghiên cứu vào năm 2004.

1.1.3.3 Phản hồi dựa trên định hướng người dùng

Trước đây, phân lớp, phản hồi liên quan tập trung vào việc học máy dựa vào phản hồi liên quan người dùng, ngày nay đã có một vài nghiên cứu quan tâm đến thiết kế mô hình phản hồi liên quan nhằm hỗ trợ, định hướng người dùng. Trong một vài nghiên cứu mới đây, đã có những nỗ lực trong việc cung cấp cho người dùng những dấu hiệu và gợi ý tìm kiếm để xây dựng truy vấn cụ thể. Một mô hình tìm kiếm tương tự đã được Fang và Geman đề xuất năm 2005, mô hình phản ứng liên tiếp người dùng sử dụng Bayesian, khuôn khổ lý thuyết thông tin. Với mục đích là để "học" một phân phối trên cơ sở dữ liệu ảnh đại diện và sử dụng sự phân phối này để tra cứu.

Một vấn đề khác được quan tâm, đó là việc lặp đi lặp lại các vòng phản hồi liên quan sẽ gây khó chịu cho người dùng, vấn đề này đã được giải quyết phần nào bởi nghiên cứu của Hoi và Lyu năm 2004 bằng cách sử dụng các bản ghi chứa thông tin phản hồi trước đó của người dùng.

1.1.3.4 Phương pháp xác suất

Phương pháp xác suất đã được Cox nghiên cứu năm 2000, các hệ thống PicHunter được đề xuất, nơi mà các mục tiêu không chắc chắn của người dùng được biểu diễn bởi một phân bố trên các mục tiêu tiềm năng, sau đó hình ảnh đích sẽ được lựa chọn dựa trên luật của Bayesian. Trong nghiên cứu của Su năm 2003, phản hồi liên quan được kết hợp sử dụng một phân lớp Bayesian dựa trên xếp hạng của hình ảnh sau mỗi bước phản hồi. Giả thiết ở đây là các đặc trưng của ví dụ dương bao gồm cả khả năng cư trú trong lớp ngữ nghĩa là như nhau, tất cả đều được tạo ra bởi một mật độ Gaussian cơ bản.

1.2 Học máy thống kê

1.2.1 Một số khái niệm

Khái niệm học máy:

Học máy (machine learning) là một lĩnh vực nghiên cứu các kĩ thuật, các phương pháp cho phép các máy tính có khả năng "học" giống như con người. Hay nói một cách khác cụ thể hơn, học máy là một phương pháp để tạo ra các chương trình máy tính bằng việc phân tích các tập dữ liệu, qua đó máy tính có khả năng tích lũy được tri thức thông qua việc học được các khái niệm để có thể ra quyết định trong các trường hợp tương tự [13].

Qua đó ta thấy học máy có liên quan rất mật thiết với thống kê, vì cả hai lĩnh vực đều nghiên cứu việc phân tích dữ liệu, nhưng học máy khác với thống kê ở chỗ học máy tập trung vào sự phức tạp của các giải thuật trong việc thực thi tính toán. Nhiều bài toán suy luận được xếp vào loại bài toán NP-khó, vì thế một phần của học máy là nghiên cứu sự phát triển các giải thuật suy luận xấp xỉ mà có thể xử lí được.

Phân loại: Có hai loại phương pháp học máy chính:

- Phương pháp quy nạp: Máy học/phân biệt các khái niệm dựa trên dữ liệu đã thu thập được trước đó. Phương pháp này cho phép tận dụng được nguồn dữ liệu rất nhiều và sẵn có.
- Phương pháp truyền dẫn: Máy học/phân biệt các khái niệm dựa vào các luật. Phương pháp này cho phép tận dụng được các kiến thức chuyên ngành để hỗ trợ máy tính.

Hiện nay, các thuật toán đều cố gắng tận dụng được ưu điểm của hai phương pháp này.

- *Một số khái niệm cơ bản trong học máy*

Không gian biểu diễn của dữ liệu

Không gian biểu diễn là một tập hợp:

- Ký hiệu là X, mỗi phần tử thuộc X có thể được gọi là các dữ liệu, các thể hiện, các đối tượng hay các ví dụ.
- Mỗi phần tử $S \in X$ được biểu diễn bởi một tập gồm n thuộc tính $S=\{s1, s2, \ldots, sn\}$.

Một đối tượng S cũng có thể được biểu diễn kết hợp với lớp liên thuộc của nó hay nói cách khác có thể được biểu diễn dưới dạng nhãn: z = (s,c).

Bản chất của dữ liệu

Bản chất của các dữ liệu có thể là các giá trị số trong tập số thực, các giá trị rời rạc, các giá trị nhị phân, dãy các phần tử trong một bảng chữ cái (alphabet), ...

Không gian biểu diễn của dữ liệu có thể biểu diễn dưới dạng thuần nhất (cùng kiểu) hoặc dưới dạng trộn (không cùng kiểu).

Tiền xử lý dữ liệu

Là quá trình xử lý dữ liệu đầu vào nhằm mục đích làm giảm số chiều của dữ liệu đầu vào, giảm số chiều của vấn đề, xử lý nhiễu . . .

Ta thực hiện như sau:

- Loại bỏ các thuộc tính không phù hợp hoặc ít phù hợp với quá trình học.
- Sử dụng các phép biến đổi tuyến tính hoặc không tuyến tính trên các thuộc tính ban đầu, nhằm giảm số chiều của không gian đầu vào.
- Dùng các chuyên gia hoặc sử dụng trực quan để phát hiện các bất thường, các lỗi mô tả thuộc tính hoặc nhãn, nhằm xử lý nhiễu.

Quá trình rời rạc hóa dữ liệu

Có những thuật toán học không xử lý được các dữ liệu mang tính liên tục. Do vậy cần phải biến đổi các dữ liệu mang tính liên tục thành các giá trị rời rạc.

Có thể sử dụng các phương pháp sau:

- Phương pháp phân đoạn.
- Phương pháp độ đo entropy.

Nếu dữ liệu tuân theo một luật phân phối nào đó, ví dụ phân phố Gauss, phân phố đều … ,Thì ta có thể rời rạc thành các khoảng phân phối tương ứng.

Tập mẫu

Tập mẫu là tập hữu hạn các ví dụ. Có ba kiểu tập mẫu:

- Tập mẫu học hay tập học.
- Tập mẫu hợp thức hoá hay tập hợp thức.
- Tập mẫu thử hay tập thử.

Quá trình tìm kiếm trong không gian giả thuyết

Trong một không gian giả thiết X thì học trở thành bài toán tìm kiếm giả thiết tốt nhất trong X. Nếu ta đánh giá mỗi giả thiết bởi một hàm "mục tiêu" thì ta xét học như một bài toán tối ưu hoá. Nghĩa là bài toán tìm phần tử của X làm tối ưu hàm mục tiêu.

Trong học máy người ta thường dùng tối ưu không ràng buộc hoặc tối ưu có ràng buộc. Các phương pháp tối ưu hoá thường dùng trong học máy như Gradient, nhân tử Lagrange ...

1.2.2 Các phương pháp học máy

1.2.2.1 Học có giám sát

Khái niệm: Học có giám sát là một kỹ thuật của ngành học máy nhằm mục đích xây dựng một hàm *f* từ dữ tập dữ liệu huấn luyện. Dữ liệu huấn luyện bao gồm các cặp đối tượng đầu vào và đầu ra mong muốn. Đầu ra của hàm *f* có thể là một giá trị liên tục hoặc có thể là dự đoán một nhãn phân lớp cho một đối tượng đầu vào.

Nhiệm vụ của chương trình học có giám sát là dự đoán giá trị của hàm *f* cho một đối tượng đầu vào hợp lệ bất kì, sau khi đã xét một số mẫu dữ liệu huấn luyện (nghĩa là các cặp đầu vào và đầu ra tương ứng). Để đạt được điều này, chương trình học phải tổng quát hóa từ các dữ liệu sẵn có để dự đoán được những tình huống chưa gặp phải theo một cách hợp lý.

Phương pháp học có giám sát có thể được mô tả tóm tắt như sau: hệ thống học sẽ quan sát một tập dữ liệu huấn luyện đã được gán nhãn, bao gồm các cặp (đặc tính, nhãn), được biểu diễn bởi $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$. Mục đích nhằm dự đoán nhãn y cho bất kỳ đầu vào mới với đặc tính x. Một nhiệm vụ của học có giám sát được gọi là hồi quy khi $y \in \mathbb{R}$ và phân lớp khi y có một tập các giá trị rời rạc.

Dữ liệu được gán nhãn: Là dữ liệu bao gồm các cặp gồm đối tượng đầu vào và đầu ra mong muốn. Đầu ra của một hàm có thể là một giá trị liên tục gọi là hồi quy, hoặc có thể là dự đoán một nhãn phân loại cho một đối tượng đầu vào gọi là phân loại.

Chương trình học có giám sát có nhiệm vụ là từ một đối tượng đầu vào hợp lệ bất kỳ thì chương trình phải dự đoán được giá trị của hàm, sau khi đã xem xét một số các cặp đầu vào và đầu ra tương ứng. Chương trình học phải có khả năng tổng quát hóa từ các dữ liệu sẵn có để dự đoán được những tình huống chưa gặp phải một cách hợp lý.

Mô hình phổ biến nhất của học có giám sát là mô hình toàn cục là mô hình ánh xạ đối tượng đầu vào đến đối tượng đầu ra mong muốn. Tuy nhiên, trong một số trường hợp, việc ánh xạ được thực hiện dưới dạng một tập các mô hình cục bộ.

1.2.2.2 Học không có giám sát

Khái niệm: Học không có giám sát là một phương pháp học máy mà dữ liệu huấn luyện là dữ liệu hoàn toàn chưa được gán nhãn, nhằm tìm ra một mô hình phù hợp với các quan sát [13].

Học không có giám sát khác với học có giám sát ở chỗ, là đầu ra đúng tương ứng cho mỗi đầu vào là chưa biết trước.

Trong học không có giám sát, một tập dữ liệu đầu vào thường được thu thập một cách ngẫu nhiên, và sau đó một mô hình mật độ kết hợp sẽ được xây dựng cho tập dữ liệu đó.

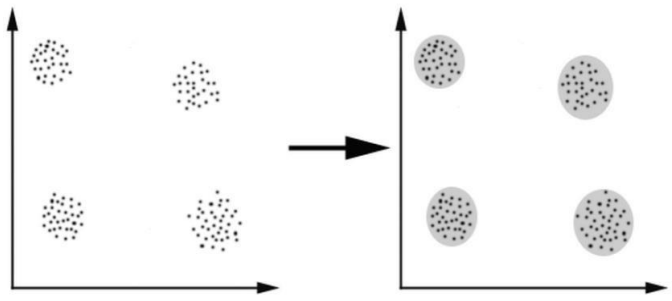
Ta có thể kết hợp học không có giám sát với suy diễn Bayes để tạo ra xác suất có điều kiện cho bất kỳ biến ngẫu nhiên nào khi biết trước các biến khác. Hay nói cách khác khi đó ta đã chuyển từ việc học không có giám sát sang học có giám sát.

* Mô hình toán học

- Cho X = {x_1, x_2, …, x_n } là tập hợp gồm n mẫu.
- Ta giả thiết rằng mẫu được tạo ra một cách độc lập và giống nhau từ một phân phối chung trên X.
- Mục tiêu là tìm ra một cấu trúc thông minh trên tập dữ liệu X.

Phân cụm dữ liệu

Một ứng dụng của học không giám sát là phân cụm dữ liệu (data clustering). Phân cụm được xem như vấn đề quan trọng nhất trong học không giám sát, vì như các vấn đề khác của phân cụm dữ liệu, nó có liên quan tới việc tìm kiếm một cấu trúc trong một tập các dữ liệu không có nhãn.



Hình 1-3: *Minh họa Phân cụm dữ liệu*

Một định nghĩa rộng hơn về phân cụm: " phân cụm là quá trình tổ chức các đối tượng dữ liệu vào trong các nhóm trong đó các đối tượng giống nhau theo một cách nào đó". Do đó, một cụm là một tập hợp các đối tượng mà giữa chúng có sự giống nhau và khác với các đối tượng thuộc về các cụm khác.

Trong ví dụ trên, chúng ta có thể dễ dàng xác định được 4 cụm mà trong đó dữ liệu được phân chia. Tiêu chí giống nhau ở đây là "khoảng cách", hai hay nhiều đối tượng thuộc về một cụm nếu chúng gần nhau hơn dựa theo khoảng cách đưa ra. Điều này gọi là phân cụm dựa trên khoảng cách.

Một dạng khác của phân cụm là Phân cụm khái niệm, hai hay nhiều đối tượng thuộc về một cụm nếu ta định nghĩa một khái niệm phổ biến cho tất cả các đối tượng. Tóm lại, các đối tượng được nhóm lại theo điều kiện mô tả chúng.

Mục đích của phân cụm dữ liệu

Mục đích của việc phân cụm dữ liệu là để xác định các nhóm trong một tập các dữ liệu không có nhãn. Nhưng làm thế nào để quyết định được điều gì tạo nên việc phân cụm tốt. Có thể nói rằng, không có một tiêu chuẩn tuyệt đối nào là tốt nhất, do đó người sử dụng phải đưa ra các tiêu chuẩn này để các dữ liệu sau khi được phân cụm sẽ phù hợp với yêu cầu của người sử dụng.

1.2.2.3 Học tăng cường

Học tăng cường là một lĩnh vực con của ngành học máy, nghiên cứu cách thức để một Agent nên chọn thực hiện các hành động nào trong một "môi trường" để cực đại hóa số "phần thưởng" nào đó. "Môi trường" trong học tăng cường được biểu diễn dưới dạng một quá trình trạng thái quyết định hữu hạn Markov (Markov decision process – MDP).

Cụ thể hơn, trong học tăng cường, các Agent tương tác với môi trường của nó bằng cách đưa ra các hành động a_1, a_2, ... Những hành động này ảnh hưởng tới trạng thái của môi trường do đó kết quả nhận được trong Agent lần lượt là số phần thưởng hoặc hình phạt r_1, r_2,... Mục đích của Agent là học một

hành động trong một cách nào đó để cực đại hóa thuộc tính phần thưởng nó nhận được (hay cực tiểu hóa rủi ro) trên vòng đời của nó.

Khác với học có giám sát, học tăng cường không có các cặp dữ liệu vào hay kết quả đúng, các hành động gần tối ưu cũng không được đánh giá đúng sai một cách tường minh.

Học tăng cường có liên quan mật thiết với lý thuyết quyết định và lý thuyết điều khiển, do đó được áp dụng trong các bài toán như: điều khiển rô bốt, điều vận thang máy, trò chơi cờ vua, ...

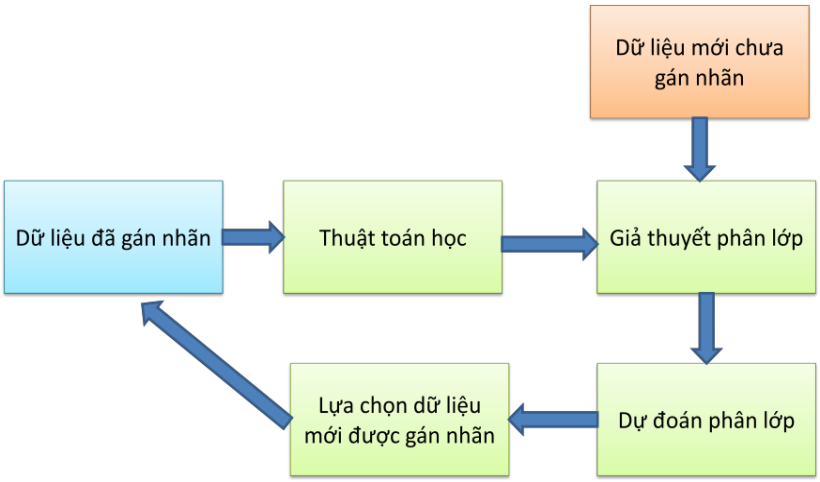
1.2.2.4 Học bán giám sát

Như chúng ta đã biết khi áp dụng học có giám thì các dữ liệu huấn luyện đã được gán nhãn. Do đó sẽ thu được kết quả có độ chính xác rất cao. Tuy nhiên, khi đó ta sẽ gặp một vấn đề rất khó khăn là khi lượng dữ liệu lớn, thì công việc gán nhãn cho dữ liệu sẽ tốn rất nhiều thời gian và công sức. Hay nói cách khác những dữ liệu được gán nhãn là rất đắt và việc tạo ra nhãn cho những dữ liệu đòi hỏi những nỗ lực rất lớn của con người.

Đối với mô hình học không có giám sát thì ngược lại, các dữ liệu huấn luyện không được gán nhãn. Do đó kết quả thu được có độ chính xác không cao. Tuy nhiên dữ liệu chưa được gắn nhãn, có thể dễ dàng thu thập được rất nhiều. Hay nói cách khác là dữ liệu chưa gắn nhãn có chi phí rất rẻ.

Học bán giám sát đã khắc phục được các nhược điểm, và phát huy được ưu điểm của học có giám sát và học không có giám sát. Bằng cách kết hợp giữa học có giám sát và học không có giám sát, với một lượng lớn dữ liệu chưa gán nhãn và một lượng nhỏ những dữ liệu đã được gán nhãn, bằng các giải thuật học bán giám sát sẽ thu được kết quả vừa có độ chính xác cao vừa mất ít thời gian công sức. Do đó, học bán giám sát là một phương pháp học đạt được hiệu quả rất tốt trong lĩnh vực học máy.

Tóm lại học bán giám sát là một phương pháp học máy mà dữ liệu huấn luyện là sự kết hợp của dữ liệu được gán nhãn và dữ liệu chưa được gán nhãn[13].



Hình 1-4: *Sơ đồ quá trình thực hiện Học bán giám sát*

1.3 Học trên đồ thị

1.3.1 Giới thiệu

Lý thuyết đồ thị đều được nghiên cứu trong Học máy và Phản hồi thông tin. Trước đây học máy đã nghiên cứu một số mô hình học dựa trên đồ thị, từ đó việc nghiên cứu tìm kiếm thông tin cũng được hưởng lợi nhiều từ các mô hình này và được xác minh cho các nhiệm vụ tìm kiếm thông tin khác nhau. Vì vậy cần nghiên cứu những mô hình học dựa trên đồ thị và các ứng dụng của nó trong tìm kiếm thông tin.

Trong việc nghiên cứu mô hình học trên trên đồ thị, ta tham khảo các phương pháp học máy theo mô hình cấu trúc đồ thị cơ bản. Lưu ý rằng các mô hình dựa trên đồ thị được nghiên cứu ở đây mang một ý nghĩa tổng quát hơn so với mô hình đồ họa, xuất hiện thường xuyên nhất trong Bayesian để phân tích văn bản. Ở mô hình đồ họa hay xác suất trong tự nhiên, cần tham khảo cấu trúc suy luận trong các dạng đồ thị. Đặc biệt, các nút trong mô hình đồ thị đại diện cho các biến ngẫu nhiên và các cạnh đại diện cho các giả định

sự phụ thuộc điều kiện. Trong nghiên cứu này, ta sẽ không chú trọng đến mô hình xác suất hoặc phi xác suất.

Trên phương diện toán học, một đồ thị là tập hợp các điểm và một số đường kết nối (có thể rỗng) tập hợp con trong chúng. Các điểm của đồ thị được gọi phổ biến nhất là đỉnh đồ thị, nhưng cũng có thể được gọi là "nút" hoặc đơn giản là "điểm". Tương tự như vậy, các đường nối các đỉnh của đồ thị được gọi phổ biến nhất là các cạnh đồ thị, nhưng cũng có thể được gọi là "liên kết", "vòng cung" hoặc đơn giản là "dòng". Trong đồ thị vô hướng thì cạnh không định hướng, tức là không phân biệt một đường từ điểm *A* đến điểm *B* với một đường từ điểm *B* đến điểm *A*. Tuy nhiên, đồ thị có hướng thì có hai hướng khác biệt. Trong nhiều trường hợp, trọng số (thường là số dương) sẽ được liên kết với mỗi cạnh để cho thấy sức mạnh của các mối quan hệ trong các cặp đỉnh tương ứng.

1.3.2 Xây dựng đồ thị

Cho đồ thị G = (V, E) với n đỉnh, V: tập đỉnh, E: tập cạnh. Đồ thị G được thể hiện bởi ma trận liền kề W trong đó $w_{ij} > 0$ nếu có cạnh nối giữa đỉnh i và đỉnh j và $w_{ij} = 0$ trong các trường hợp còn lại.

Đặt $w_{ij} = 0$ nếu đồ thị có chứa chu trình thì không tính cạnh đó. Với đồ thị có hướng, đỉnh i có tổng bậc đầu ra $w_{i+} = \sum_{j=1}^{n} w_{ij}$ và tổng bậc đầu vào $w_{+i} = \sum_{j=1}^{n} w_{ij}$ .Tổng trọng số của đồ thị ký hiệu $vol(G) = \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}$. Giả sử không có đỉnh nào bị cô lập, do đó $w_{+i} > 0$ và $w_{i+} > 0$. Trong trường hợp đặc biệt của đồ thị không có trọng số, ta có $w_{ij} \in \{0, 1\}$. Với đồ thị vô hướng, tổng bậc đầu vào và đầu ra tương ứng với $d_i (= w_{i+} = w_{+i})$ đại diện cho cả hai để nhấn mạnh như một thuộc tính.

Trong thực tế, w_{ij} thường có thể được giải thích một cách tự nhiên. Nó có thể là số lượng các siêu liên kết từ một trang web tới các trang khác hay là một giá trị nhị phân chỉ ra protein i tương tác với protein j.

Tuy nhiên, khi các trọng số không sẵn có từ các dữ liệu, thường có hai bước xử lý để xây dựng chúng. Bước đầu tiên là sử dụng một hàm không âm và đối xứng để định lượng các mối quan hệ giữa một cặp đỉnh. Ví dụ, nếu mỗi đỉnh được xét trong không gian Euclidean ^{d}, một sự lựa chọn phổ biến là sử dụng hàm mật độ Gaussian $a(i, j) = \exp(-\|x_i - x_j\|^2 / 2\sigma^2)$, trong đó $x_i \in E^d$ thể hiện vị trí đỉnh i. Sau đó, chúng ta cần xây dựng trọng số w_{ij} dựa trên mối quan hệ giữa các cặp a(i, j). Phương pháp tiếp cận ε-láng giềng, đặt $w_{ij} = a(i, j)$ nếu $a(i,j) > \varepsilon$ và $w_{ij} = 0$ trong các trường hợp còn lại. Mặt khác, phương pháp k-láng giềng gần nhất đặt $w_{ij} = a(i, j)$ nếu j là một trong số các láng giềng gần i nhất và $w_{ij} = 0$ trong các trường hợp còn lại. Từ đây, ta giả sử rằng đồ thị được xây dựng với các trọng số cạnh được đưa ra dựa trên phương pháp xác định trọng số trong không gian Euclidian.

1.3.3 Phân tích đồ thị

Khi một đồ thị được xây dựng, chúng ta có thể giả định rằng tất cả các cấu trúc thông tin đã được nhúng. Tuy nhiên, hầu hết các thông tin có thể không được sử dụng trực tiếp, vì vậy nó cần được định nghĩa lại. Nói chung, cần phân tích sâu hơn về các đồ thị nhằm mục đích sau đây:

- Xác định các tính chất hoặc tìm những thông tin bất biến
- Phù hợp với một số giả định được thực hiện trên các dữ liệu
- Gần đúng với cấu trúc đơn giản

Dưới đây, sẽ trình bày một số phương pháp phân tích đồ thị tương ứng với các mục đích nêu trên.

Phân tích dựa trên Lý thuyết phổ đồ thị

Một cách tiếp cận chủ đạo để phân tích đồ thị dựa trên Lý thuyết phổ đồ thị, đó là tập trung vào nghiên cứu các thuộc tính dựa trên việc đo những giá trị riêng của các vector riêng trong ma trận kề.

Một khái niệm quan trọng tập trung trong Lý thuyết phổ đồ thị là đồ thị Laplacian. Với một ma trận kề W đối xứng, đồ thị Laplace được định nghĩa như sau: L = D – W

Trong đó D là đường chéo ma trận D = $diag(d_1, d_2, \ldots, d_n)$ với $di = \sum j \in v^{w_{i,j}}$.

Đồ thị Laplace L có các thuộc tính sau:

- Tất cả các giá trị riêng của nó là không âm.
- giá trị riêng nhỏ nhất của nó luôn luôn bằng 0. Đối với đồ thị Laplace L chưa chuẩn hóa, vector riêng tương ứng là $e=(1/\sqrt{n},1/\sqrt{n},...,1/\sqrt{n})^T$;

Tập hợp các giá trị riêng của đồ thị Laplace L có thể được ký hiệu bởi $0=\lambda_0 \le \lambda_1 ... \le \lambda_{n-1}$, gọi là phổ của L (hoặc đồ thị chính nó). Lý thuyết phổ đồ thị cho chúng ta biết rằng cấu trúc của một đồ thị và thuộc tính có thể được hình thành từ phổ của nó. Đặc biệt, nó đã được chứng minh rằng những giá trị riêng có liên quan chặt chẽ với hầu hết các bất biến chính của đồ thị và liên kết các thuộc tính bên ngoài với nhau.

Phân tích dựa trên Lý thuyết trường ngẫu nhiên

Láng giềng của các nút ở trong đồ thị duy trì cấu trúc cục bộ của dữ liệu và có thể được giải thích như là ràng buộc về ngữ cảnh trên dữ liệu. Điều này thúc đẩy việc phân tích đồ thị bởi mô hình thuộc tính không gian của nó. Lý thuyết trường ngẫu nhiên là một cách tổng quát và hữu hiệu để giải quyết các thuộc tính không gian. Nó đặc trưng bởi mối liên hệ láng giềng dựa trên ngôn ngữ xác suất [13].

Có hai cách điển hình để mô tả sự phụ thuộc, đó là: Markov và Gaussian. Những giả thiết thuộc tính của Markov trên sự phụ thuộc trường ngẫu nhiên Markov và các trường ngẫu nhiên có điều kiện, trong khi giả thiết thuộc tính Gaussian dự sự phụ thuộc trường Gaussian ngẫu nhiên. Mặc dù định nghĩa khác nhau, ba trường ngẫu nhiên chia sẻ hai đặc điểm chung có liên quan họ chặt chẽ với nhau.

- Tất cả các mô hình đồ thị đều là vô hướng.
- Thông thường, nó rơi vào cùng một nhóm với hàm mật độ xác suất theo cấp số nhân.

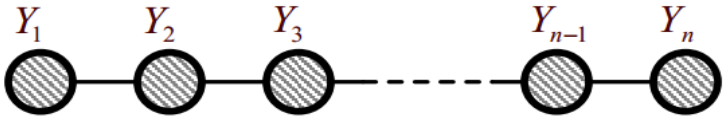
Để bắt đầu, chúng ta hãy giải thích các ký hiệu sẽ được sử dụng. Đối với đồ thị G=(V,E) như được định nghĩa trước kia, hai bộ biến số ngẫu nhiên được giới thiệu: một tập hợp các biến số đầu vào X=$\{X_i\}$ và một tập hợp các biến đầu ra Y=$\{Y_i\}$. Các biến đầu vào dao động trên tất cả các dữ liệu quan sát và các biến đầu ra phạm vi trên tập hữu hạn các nhãn hoặc có giá trị liên tục thực gắn liền với mỗi dữ liệu.

Trường ngẫu nhiên Markov

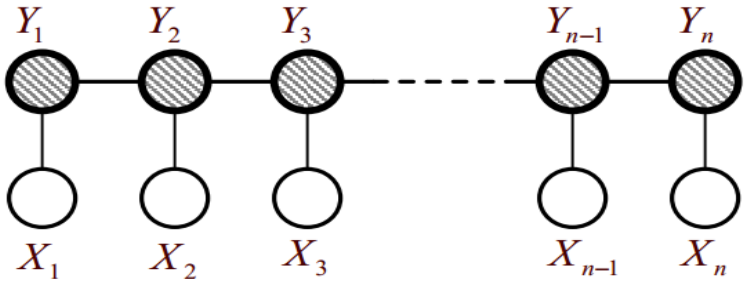
Một trường ngâu nhiên Markov - Markov Random Field (MRF) được xác định liên quan đến đồ thị G, nếu $\Pr(Y_i \mid \{Y_j \mid i \neq j\}) = \Pr(Y_i \mid \{Y_j \mid i \sim j\})$ trong đó *i~j* đại diện cho nút thứ i và nút thứ j là láng giềng của nhau.

Các cấu trúc đồ thị của trường ngẫu nhiên tạo nhân tố giúp phân bố chung trên biến đầu ra trong tạo các Hàm số tiềm năng thực sự có giá trị. Mỗi Hàm tiềm năng hoạt động trên một tập hợp con của các biến ngẫu nhiên. Để đảm bảo (có điều kiện) biến độc lập sẽ không xuất hiện trong các hàm tiềm năng tương tự, cách dễ nhất là để xác định một Hàm tiềm năng Φ_k của mỗi tập hợp C_k trong đồ thị. Sau đó, sự phân bố chung trên các biến đầu ra trở thành

$$Pr(Y = y) = \frac{1}{Z}\prod_k \Phi_k(y_{(k)}) \quad (1.1)$$



Hình 1-5: *chuỗi cấu trúc MRF*



Hình 1-6: *chuỗi cấu trúc CRF*

Trường ngẫu nhiên có điều kiện

Một trường ngẫu nhiên có điều kiện Conditional Random Field (CRF) được xác định khi số lượng các biến ngẫu nhiên Y, có điều kiện bởi các biến đầu vào X, tuân theo tính chất Markov $\Pr(Y_i \left| \{X, Y_j \right| i \neq j\}) = \Pr(Y_i \left| \{X, Y_j \right| i \neq j\})$. Minh họa so sánh các cấu trúc đồ thị của chuỗi cấu trúc MRF và CRF ở Hình 1-6 và Hình 1-7.

Tương tự như MRF, với sự giúp đỡ của xác suất chung có điều kiện của các biến đầu ra, từ đó ta có thể định nghĩa một hàm tiềm năng cho mỗi nhóm trong đồ thị. Cũng như MRF, CRF cũng bị mất thời gian tính toán và do đó chủ yếu được áp dụng cho các cấu trúc đồ thị đơn giản.

Trường ngẫu nhiên Gaussian

Thay vì phụ thuộc trực tiếp bằng cách tính Markov, Trường ngẫu nhiên Gaussian giả định một phân bổ Gaussian qua xác suất chung của bất kỳ chuỗi nhãn

$$p(y \left| X \right.) = \frac{1}{(2\pi)^{n/2} \left| \sum (X)^{1/2} \right|} \exp(-\frac{1}{2}(y-\mu)^T \sum (X)^{-1}(y-\mu)) \quad (1.2)$$

Với ma trận hiệp phương sai Σ mã hóa các thông tin cấu trúc thể hiện trong đồ thị. Các ma trận hiệp phương sai Σ được tính toán bằng cách áp dụng một hàm K (.,.) với các mẫu đầu vào của các ví dụ.

Trường ngẫu nhiên Gaussian có ưu điểm ở chỗ nó chắc chắn dự đoán được các giá trị đầu ra có thể xảy ra nhất của các biến không quan sát được. Điều này thường được thể hiện qua giá trị trung bình và phương sai của các biến để dự đoán. Cụ thể, giả sử ta có *l+1* ví dụ $X \cup \{x'\}$. nơi các ví dụ *l* đầu tiên được quan sát với giá trị *y* và một *x'* để tạo ra một dữ liệu mới có giá trị đầu ra có thể tiên đoán được.

Các ma trận hiệp phương sai Σ * có thể được phân rã như

$$\Sigma^* = \begin{bmatrix} \Sigma & c \\ c^T & K(x', x') \end{bmatrix}$$

Sau đó, sự phân bố của các ví dụ mới x cho các ví dụ quan sát được ta có một Gaussian trung bình có điều kiện và phương sai

$$\mu_{y'} \mid X = c^T \Sigma^{-1} (y - \bar{y})$$

$$\sigma^2_{y'} \mid X = K(x', x') - c^T \Sigma^{-1} c$$

Phân tích dựa trên ma trận xấp xỉ & nhân tử

Khi ta sử dụng một ma trận đại diện cho một đồ thị, chẳng hạn như ma trận kề hoặc các biến thể khác, đó là một khó khăn tính toán tiềm năng khi các đồ thị có nhiều nút và cạnh. Một kỹ thuật đưa ra để tính được gần đúng các ma trận trong khi vẫn giữ càng nhiều thông tin càng tốt. Điều này đặc biệt hữu ích khi ta quan tâm nhiều hơn trong việc tìm hiểu cấu trúc toàn cục của dữ liệu. Lấy quan điểm này, một vài mô hình học dựa trên đồ thị có thể được xem như là một xấp xỉ ma trận hay một ma trận nhân tử.

Tìm kiếm thứ hạng k tối ưu xấp xỉ cho một thứ hạng r . Ma trận A(k<r) có thể được xây dựng như sau:

$$B = \arg\min_{Rank(B)=k} \|A - B\|_F \tag{1.3}$$

Áp dụng phân rã số ít giá trị cho ma trận A, chúng ta có

$$A = \mathrm{USV}^T \tag{1.4}$$

nơi U và V là ma trận trực giao và S = diag(s_1 , s_2, …, s_r ,0,…,0) với $s_1 \geq s_2 \geq ... \geq s_r \geq 0$, các giải pháp cho các vấn đề xấp xỉ bậc thấp hơn sẽ là

$$B = U_k diag(s_1, s_1, ..., s_1) V_k^T \tag{1.5}$$

Vấn đề liên quan cao đến ma trận xấp xỉ, ma trận nhân tử không âm cố gắng để gần đúng ma trận A với hai ma trận không âm U và V

$$A \approx UV \tag{1.6}$$

Để đo lường chất lượng xấp xỉ, việc tính toán hai hàm được sử dụng. Phép đo đầu tiên là bình phương của khoảng cách Euclide

$$\|A - B\| = \sum_{i,j} (A_{i,j} - B_{i,j})^2 \tag{1.7}$$

và phép đo thứ hai là sự phân kỳ

$$D(A\|B) = \sum_{i,j} A_{i,j} \log \frac{A_{i,j}}{B_{i,j}} - A_{i,j} + B_{i,j}) \tag{1.8}$$

Lưu ý rằng ở phép đo thứ hai D(A||B) là luôn luôn không âm và đạt số không chỉ khi $A_{i,j} = B_{i,j}$ giữ cho tất cả cặp (i, j).

Một thuật toán lặp đã được đề xuất để giải quyết hiệu quả các vấn đề bằng cách lặp đi lặp lại việc giảm thiểu việc tính toán hai Hàm trên. Đặc biệt, các quy tắc cập nhật tối thiểu sau khoảng cách Euclide ||A-UV||

$$\begin{aligned} U_{i,a} &\leftarrow U_{i,a} \sum_k \frac{A_{i,k}}{(UV)_{i,k}} V_{a,k} \\ U_{i,a} &\leftarrow \frac{U_{i,a}}{\Sigma_j U_{j,a}} \end{aligned} \tag{1.9}$$

và quy tắc cập nhật sau giảm thiểu sự phân kỳ D(A||UV)

$$V_{a,k} \leftarrow V_{a,k} \sum_i U_{i,a} \frac{A_{i,k}}{(UV)_{i,k}} \quad (1.10)$$

1.3.4 Các mô hình học dựa trên đồ thị

1.3.4.1 Học có giám sát

k- Láng giềng gần nhất (k-NN)

Các phương pháp *k*-NN sử dụng trọng số như những quan sát trong việc thiết lập đào tạo gần gũi với một ví dụ thử nghiệm để tạo thành một dự đoán.

$$y_i = \sum_{x_j \in NN_i} \mathrm{w}_j y_j \quad (1.11)$$

trong đó NN_i đại diện cho tập láng giềng gần nhất của ví dụ dữ liệu thử nghiệm, x_i và w_j là một trọng số đáp ứng $\Sigma_j \mathrm{w}_j = 1$ và luôn luôn liên quan đến sự giống nhau giữa x_i và x_j

Phương pháp k-NN giải thích xác suất của một thử nghiệm chẳng hạn x_i được phân loại vào thứ j lớp C_j . Có thể được viết như sau

$$\Pr(x_i \in C_j) = \sum_{x' \in NN_t} \Pr(x' \in C_j) \Pr(x_i \rightarrow x') \Pr(x' \in C_j) \quad (1.12)$$

Trong đó $\Pr(x' \epsilon C_j)$ là một yếu tố bình thường hóa $\Pr(x_i \rightarrow x')$ có liên quan đến sự tương đồng giữa x_i và x' , và $\Pr(x' \epsilon C_j) = 1$ nếu x' thuộc về lớp C_j và bằng 0 nếu ngược lại.

Quá trình Gaussian

Quá trình Gaussian được định nghĩa như là một hàm phân bố xác suất $y(\mathrm{x})$, có thuộc tính lựa chọn bất kỳ hữu hạn các điểm $\mathrm{x}^{(1)}$, $\mathrm{x}^{(2)}$,..., $\mathrm{x}^{(k)}$, có mật độ biên $\Pr(y(\mathrm{x}^{(1)}), y(\mathrm{x}^{(2)}),..., y(\mathrm{x}^{(k)})$ như một Gaussian. Các giả định về một trường ngẫu nhiên Gaussian (như đã nêu ở phần trước) rơi vào các thể loại Quá trình Gaussian, đó là lý do mà Quá trình Gaussian như một mô hình dựa trên đồ thị.

1.3.4.2 Học không giám sát

Phân cụm dựa trên quang phổ

Phương pháp phân cụm dựa trên quang phổ xem các vấn đề về phân cụm dữ liệu là một vấn đề của phân vùng đồ thị. Cần 2 chiều phân vùng đồ thị như là một ví dụ để tạo thành hai dữ liệu tách rời bộ A và B từ một đồ thị G = (V, E), cạnh nối hai phần này cần được loại bỏ. Các mức độ không giống nhau giữa các bộ phận phân vùng được thể hiện bởi các khái niệm về *cut*, được định nghĩa như sau:

$Cut(A,B) = \Sigma_{v_i \in A, v_j \in B} w_{i,j}$. Nói chung, một phân vùng tốt sẽ dẫn đến một giá trị *cut* nhỏ. Để giải quyết vấn đề cân bằng khác nhau, có một số biến thể của định nghĩa *cut* dẫn đến việc phân vùng tối ưu trong các giác quan khác nhau. Đầu tiên, ta xác định S: $S(A,B) = \Sigma_{i \in A} \Sigma_{j \in B} w_{i,j}$ và $d_A = \Sigma_{i \in A} d_i$. Tỷ lệ *Cut* giải quyết số dư trên các kích thước của đồ thị phân vùng dẫn đến giảm thiểu của hàm mục tiêu sau đây

$$\Im_{RCut} = \frac{S(A,B)}{|A|} + \frac{S(A,B)}{|B|} \tag{1.13}$$

Thông thường, *Cut* giải quyết các mối quan tâm về cân bằng trọng lượng của đồ thị phân vùng dẫn đến giảm thiểu

$$\Im_{NCut} = \frac{S(A,B)}{d_A} + \frac{S(A,B)}{d_B} \tag{1.14}$$

Min-Max Cut giải quyết các mối quan tâm cân bằng giữa trọng số ở một cụm và trọng số liên cụm trong một phân vùng dẫn đến giảm thiểu

$$\Im_{MCut} = \frac{S(A,B)}{S(A,A)} + \frac{S(A,B)}{S(B,A)} \tag{1.15}$$

Bằng cách làm dãn các cụm để giá trị thực giảm thiểu trên tất cả các vấn đề có thể được xây dựng như vector eigen, việc này liên quan đến đồ thị Laplace như Phân tích dựa trên Lý thuyết quang phổ đồ thị

Hạt nhân k-means

K-means nhằm giảm thiểu tổng khoảng cách trong cụm. Mặc dù đồ thị không được xây dựng một cách rõ ràng, một độ đo khoảng cách (một hàm hằng số) là không thể thiếu cho các thuật toán. Độ đo khoảng cách được sử dụng để xác định số lượng các mối quan hệ của các ví dụ dữ liệu đến tâm cụm.... Vấn đề này đã được chứng minh rằng *k*-means phân cụm có mối quan hệ chặt chẽ với các phân cụm phổ.

Cụ thể, giả sử một hạt nhân được đưa vào cho khoảng cách số liệu với hàm ánh xạ Φ, ta biểu diễn các cụm dự kiến *k* như $\{C_j\}_{j=1}^{k}$. Hàm mục tiêu tổng quát thuật toán hạt nhân *k*-means được xác định là giảm thiểu như sau

$$D\left(\{C_j\}_{j=1}^{k}\right)=\sum_{j=1}^{k}\sum_{x_i\in C_j}\left\|\Phi(x_i)-m_j\right\|^2 \tag{1.16}$$

nơi m_j là trọng tâm của cụm C_j

$$m_j=\frac{\Sigma_{x_l\in C_j}\Phi(x_l)}{\Sigma_{x_l\in C_j}} \tag{1.17}$$

Xếp hạng

Trong nhiều trường hợp, Các dữ liệu cần sắp xếp theo một thứ tự được mong đợi, vì vậy vấn đề xếp hạng được đặt ra. Thông thường, tạo ra một chức năng được huấn luyện (được học) để ghi lại điểm số bảng xếp hạng có thể được tạo ra. Mô hình điển hình cho mục đích này là mô hình hồi quy. Tuy nhiên, nhiệm vụ này trở nên khó khăn hơn khi các thông tin dữ liệu khó mô tả nhưng các mối quan hệ giữa các dữ liệu là tương đối dễ dàng hơn để nắm bắt. Một số mô hình dựa trên đồ thị đã được đề xuất để tạo ra bảng xếp hạng dựa trên các phân tích về mối quan hệ giữa các cặp dữ liệu. Như một mối quan hệ tuyến tính, một thứ hạng có thể được xem như là một xấp xỉ bậc nhất đối với cấu trúc của dữ liệu. Theo đó, ta sẽ tìm thấy mô hình xếp hạng thường dựa trên đồ thị.

1.3.4.3 Học bán giám sát

Thuật toán Mincuts

Thuật toán Mincuts là một thuật toán dựa trên việc tìm kiếm một lát cắt nhỏ nhất trên đồ thị, chúng sử dụng các cặp quan hệ giữa các điểm dữ liệu để học từ cả dữ liệu đã gán nhãn và chưa gán nhãn. Thuật toán sử dụng một độ đo tương tự giữa các điểm dữ liệu để xây dựng đồ thị và sau đó đưa ra kết quả phân loại dữ liệu tương ứng với việc phân chia đồ thị bằng cách cực tiểu hóa số lượng các cặp dữ liệu tương tự nhau mà có nhãn khác nhau [13].

Bài toán cho trước một tập dữ liệu đã gán nhãn và chưa gán nhãn, chúng ta xây dựng một đồ thị trên các mẫu dữ liệu này, do đó lát cắt nhỏ nhất trên đồ thị này sẽ cho ta việc gán nhãn nhị phân tối ưu trên các dữ liệu chưa gán nhãn theo một hàm tối ưu nhất định.

Giống như hầu hết các phương pháp tiếp cận khác dùng để kết hợp dữ liệu đã gán nhãn và chưa gán nhãn, ý tưởng chính của thuật toán này là gán giá trị cho các dữ liệu chưa có nhãn để tối ưu hóa một hàm mục tiêu có liên quan. Với phương pháp tiếp cận Mincuts, loại hàm được tối ưu được giới hạn để chỉ phụ thuộc vào mối quan hệ giữa các cặp dữ liệu. Điều gì làm cho phương pháp này thực sự được quan tâm? Đó là các hàm chúng ta có thể xử lý, các lát cắt đồ thị đưa ra một thuật toán có thời gian đa thức để tìm ra sự tối ưu hóa toàn cục.

Thuật toán

Giả sử, xét bài toán phân lớp nhị phân (dữ liệu đã gán nhãn được ký hiệu bởi +, dữ liệu chưa gán nhãn được ký hiệu bởi -), L: là tập các dữ liệu đã gán nhãn, U: là tập các dữ liệu chưa gán nhãn, L_+: tập các dữ liệu có nhãn dương trong tập L, L_- : tập các dữ liệu có nhãn âm trong tập L.

Thuật toán như sau:

Bước 1 . Xây dựng một đồ thị trọng số G=(V, E) trong đó V=L ∪ U ∪ {v_+, v_-}, và E ⊆ V×V. Với mỗi cạnh e ∈ E có trọng số w(e). Gọi các đỉnh v_+, v_- là các đỉnh phân lớp (Classication vertices) và tất cả các đỉnh khác gọi là các đỉnh mẫu (Example vertices).

Bước 2. Các đỉnh phân lớp có nhãn giống nhau được nối với nhau bởi các cạnh có trọng số ∞. Ví dụ, w(v, v_+)=∞ với tất cả các đỉnh v ∈ L_+ và w(v, v_+)=∞ với tất cả các đỉnh v ∈ L

Bước 3. Các cạnh nối giữa Các Đỉnh mẫu được gán trọng số dựa trên mối quan hệ giữa các mẫu dữ liệu đó, ví dụ như sự giống nhau hoặc khoảng cách giữa chúng. Việc lựa chọn cụ thể các trọng số cạnh này sẽ được đề cập ở phần sau. Hàm gán trọng số cho các cạnh được kết nối bởi Các đỉnh dữ liệu sẽ được gọi là Hàm gán trọng số w.

Bước 4. Bây giờ ta xác định một lát cắt nhỏ nhất (v_+, v_-) cho đồ thị; đó là phải tìm ra tổng trọng số nhỏ nhất của các cạnh mà nếu loại bỏ chúng đi thì sẽ làm mất kết nối giữa các đỉnh v_+ và v_- . Điều này có thể tìm được bằng cách sử dụng thuật toán Luồng cực đại mà trong đó các đỉnh v_+ là các đỉnh đầu, v_- là các đỉnh được ẩn đi và các trọng số cạnh được xem như capacities. Việc loại bỏ các cạnh theo lát cắt sẽ chia đồ thị thành hai tập đỉnh được gọi là V_+ và V_- , với v_+ ∈ V_+ và v_-∈V_- Nếu có nhiều lát cắt, ta có thể đặt cho thuật toán chọn một tập đỉnh V_+ là nhỏ nhất.

Bước 5. Ta gán nhãn dương (+) cho tất cả các mẫu dữ liệu chưa có nhãn trong tập V_+ và gán nhãn âm (-) cho tất cả các mẫu dữ liệu chưa có nhãn trong tập V_- .

Trong thuật toán này, nếu các cạnh nối giữa các điểm dữ liệu có trọng số cao tương tự nhau thì hai điểm dữ liệu tương tự nhau sẽ được đặt trong cùng một tập đỉnh thu được từ lát cắt nhỏ nhất. Điều này phù hợp với giả thiết cơ bản của nhiều thuật toán học (giống như láng giềng gần nhất) rằng

các điểm dữ liệu tương tự nhau thì có khuynh hướng được phân lớp giống nhau.

Như vậy, thực tế đặt ra là chúng ta có thể đánh trọng số các cạnh như thế nào? Nếu chúng ta có một khái niệm gọi là "khoảng cách" giữa các điểm dữ liệu thì các mẫu dữ liệu này sẽ có nhãn giống nhau, sau đó hàm tính khoảng cách sẽ đặt các cạnh nối các điểm dữ liệu gần nhau một trọng số cao và các cạnh có nối giữa các điểm dữ liệu xa nhau (hoặc không có cạnh nối) một trọng số thấp. Nếu chúng ta không khởi tạo ban đầu một hàm tính khoảng cách thì chúng ta có thể giữ các điểm dữ liệu đã gán nhãn trong một số thuật toán học trợ giúp để học một hàm khoảng cách. Ví dụ, chúng ta có thể sử dụng các dữ liệu đã gán nhãn để đánh trọng số cho các thuộc tính dựa trên các thông tin có được. Ta cũng có thể đánh trọng số cho một cạnh (x, y) với $x \in U$ dựa trên $y \in L$ hoặc không. Việc lựa chọn hàm đánh trọng số cạnh có thể ảnh hưởng đến chất lượng đầu ra của thuật toán.

Các trường Gaussian ngẫu nhiên và các hàm điều hòa

Các trường Gaussian ngẫu nhiên

Một cách tiếp cận khác của học bán giám sát là đề xuất dựa trên mô hình Gaussian ngẫu nhiên (GRF). Dữ liệu đã gán nhãn và chưa gán nhãn được đưa ra như là các đỉnh trong một đồ thị có trọng số, với việc mã hóa trọng số các cạnh giữa các mẫu dữ liệu giống nhau. Bài toán học sau đó được xây dựng trong các trường Gaussian ngẫu nhiên trên đồ thị này, tại đó ý nghĩa của các trường được đặc trưng bởi các hàm điều hòa và hiệu quả thu được bằng cách sử dụng các phương pháp ma trận hay lan truyền tin cậy.

Không giống như các phương pháp khác hiện nay, dựa trên hàm năng lượng cực tiểu và các trường ngẫu nhiên trong học máy, ta xem xét các trường Gaussian ngẫu nhiên trên không gian trạng thái liên tục thay vì các trường ngẫu nhiên trên các tập dữ liệu rời rạc. Đặc biệt, dạng phổ biến nhất của các trường (field) có thể xảy ra là duy nhất, được đặc trưng bởi các hàm điều hòa

và có thể tính toán dựa vào các phương pháp ma trận hay lan truyền. Ngược lại, với các trường ngẫu nhiên đa nhãn, việc tính toán cấu hình năng lượng thấp nhất thường là NP-khó và các thuật toán xấp xỉ hay ước lượng được sử dụng. Kết quả của thuật toán phân lớp với các trường Gaussian có thể được xem như một dạng của phương pháp tiếp cận láng giềng gần nhất, tại đó các mẫu dữ liệu láng giềng đã được gán nhãn bởi phương pháp Bước di chuyển trên đồ thị.

Ta giả sử có ℓ điểm đã được gán nhãn $(x_1, y_1),..., (x_\ell, y_\ell)$ và u điểm chưa gán nhãn $x_{\ell+1}, ..., x_{\ell+u}$; $\ell << u$.

Cho $n=\ell+u$ là tổng số các điểm dữ liệu. Để bắt đầu, ta giả sử các nhãn mang giá trị nhị phân: $y \in \{0, 1\}$. Xét một đồ thị liên thông G=(V, E) với tập các đỉnh V tương ứng với n điểm dữ liệu, với tập L các đỉnh ứng với các điểm dữ liệu đã gán nhãn, với các nhãn $y_1,..., y_\ell$ và tập U các đỉnh tương ứng với các điểm dữ liệu chưa được gán nhãn. Nhiệm vụ của chúng ta là dự đoán các nhãn cho các điểm dữ liệu trong tập U.

Giả sử một ma trận trọng số đối xứng $W_{n\times n}$ trên các cạnh của đồ thị được đưa ra. Ví dụ khi $x \in \sigma^m$ thì ma trận trọng số được biểu diễn theo công thức sau:

$$w_{ij} = \exp\left(-\sum_{d=1}^{m} \frac{\left(x_{id} - x_{jd}\right)^2}{\sigma_d^2} \right) \qquad (1.18)$$

Trong đó, x_{id} là thành phần thứ d của mẫu dữ liệu x_i biểu diễn như một véc tơ $x_i \in \sigma^m$ và $\sigma_1, ..., \sigma_m$ là siêu tham số cho mỗi chiều.

Chiến lược của chúng ta đầu tiên là tính toán một hàm giá trị thực $f: V \rightarrow R$ trên đồ thị G với các thuộc tính nhất định và sau đó gán các nhãn dựa trên f. Chúng ta hạn chế f để $f(i) = f_l(i) \equiv y_i$ trên dữ liệu đã gán nhãn $i=1,..., \ell$. Bằng trực giác chúng ta muốn các điểm dữ liệu chưa gán nhãn gần

nhau trong đồ thị sẽ có nhãn tương tự nhau. Điều này dẫn tới sự lựa chọn hàm bậc hai:

$$E(f)=\frac{1}{2}\sum_{i,j} w_{ij}\left(f(i)-f(j)\right)^2 \tag{1.19}$$

Rõ ràng, E được cực tiểu hóa bởi hàm hằng số. Nhưng vì ta đã có một số dữ liệu đã gán nhãn, chúng ta gán cho *f* một giá trị $f(i) = y_i$, (i ∈ L trên tập dữ liệu đã gán nhãn). Ta chỉ định một phân bổ xác suất tới hàm f bởi một trường Gausian ngẫu nhiên (GRF).

$$p(f)=\frac{1}{Z}e^{-\beta E(f)} \tag{1.20}$$

Trong đó β là một tham số "nghịch đảo" và Z là một hàm phân hoạch.

$$Z=\int_{f_L=Y_L}\exp(-\beta E(f))\,df \tag{1.21}$$

được chuẩn hóa trên các hàm ràng buộc với Y_L trên dữ liệu đã gán nhãn. Ta đang quan tâm đến vấn đề suy luận $p(f_i|Y_L)$, i ∈ U hay trung bình $\int_{-\infty}^{\infty} f_i\,p(f_i|Y_L)\,df_i$

Sự phân bố *p(f)* giống các trường Markov ngẫu nhiên với các trạng thái rời rạc. Thực tế thì sự khác biệt duy nhất là trạng thái giá trị số thực. Tuy nhiên, điều này làm cho vấn đề suy luận đơn giản hơn rất nhiều. Bởi vì hàm năng lượng bậc 2 *p(f)* và $p(f_U|Y_U)$ đều là phân phối Gaussian đa biến. Do đó *p* được gọi là GRF. Biên $p(f_i|Y_L)$ là một biến đơn Gaussian và gần với giải pháp đưa ra.

Đồ thị Laplace

Bây giờ ta xem xét tổ hợp Laplace, ký hiệu: Δ

Cho D là ma trận đường chéo bậc của các đỉnh, có $D_{ii} = \Sigma_j W_{ij}$ là bậc của đỉnh *i*.

Laplace được định nghĩa như sau: $\Delta \equiv D - W$.

Laplace là viết tắt cho hàm năng lượng:

$$E(f) = \frac{1}{2}\sum_{i,j} w_{ij}(f(i) - f(j))^2 = f^T \Delta f \quad (1.22)$$

Hàm Gaussian ngẫu nhiên có thể được viết như sau:

$$p(f) = \frac{1}{Z} e^{-\beta f^T \Delta f} \quad (1.23)$$

Các hàm điều hòa

Không khó để chỉ ra rằng hàm năng lượng cực tiểu $f = \arg\min_{f_L = Y_L} E(f)$ là hàm điều hòa, cụ thể là $\Delta_f = 0$ trên các điểm dữ liệu chưa gán nhãn trong tập U, và bằng $\Delta_f = Y_L$ trên các điểm dữ liệu đã gán nhãn L.

Ký hiệu hàm điều hòa là: *h*

Thuộc tính điều hòa ở đây có nghĩa là giá trị của *h(i)* tại mỗi điểm dữ liệu chưa gán nhãn i là giá trị trung bình của các láng giềng của nó trong đồ thị, ta có công thức sau:

$$h(i) = \frac{1}{D_{ii}} \sum_{j \sim i} w_{ij} h(j), \text{ for } i \in U \quad (1.24)$$

Do các nguyên lý cực đại của hàm điều hòa (Doyle & Snell, 1984), *h* thỏa mãn $0 \le h(i) \le 1$ với $i \in U$ (lưu ý: $h(i)=0$ hoặc $h(i)=1$ cho mỗi $i \in L$).

Để tính toán giải pháp điều hòa, ta chia nhỏ ma trận W (tương tự D, Δ,...) thành 4 khối cho L và U:

$$W = \begin{bmatrix} W_{LL} & W_{LU} \\ W_{UL} & W_{UU} \end{bmatrix} \quad (1.25)$$

Lời giải điều hòa $\Delta_h = 0$ với $h_L = Y_L$ được đưa ra bởi

$$
\begin{aligned}
h_U &= (D_{UU} - \mathrm{W}_{UU})^{-1} \mathrm{W}_{UL} Y_L \\
&= -(\Delta_{UU})^{-1} \Delta_{UL} Y_L \\
&= (I - P_{UU})^{-1} P_{UL} Y_L
\end{aligned} \tag{1.26}
$$

Mô tả cuối cùng giống với công thức $f_U = (I - P_{UU})^{-1} P_{UL} Y_L$, mà P=D/W là ma trận lan truyền trong đồ thị. Thuật toán lan truyền nhãn (labled propagation) về sau đã tính hàm điều hòa này.

Hàm điều hòa đã cực tiểu hóa năng lượng và do đó nó là một dạng của Trường Gaussian ngẫu nhiên.

Hàm điều hòa có thể được thể hiện trong một vài cách nhìn khác nhau và những cách nhìn khác nhau này cung cấp một tập hợp các lý luận bổ sung và kỹ thuật phong phú cho lĩnh vực học bán giám sát trên đồ thị.

Kết luận Chương I:

Trong chương này, chúng ta đã nghiên cứu tổng quan tra cứu ảnh dựa trên nội dung; tra cứu ảnh dựa trên nội dung với phản hồi liên quan; các phương pháp học máy và học trên đồ thị gồm có các mô hình Học có giám sát, Học không giám sát và đặc biệt là Học bán giám sát dựa trên đồ thị.

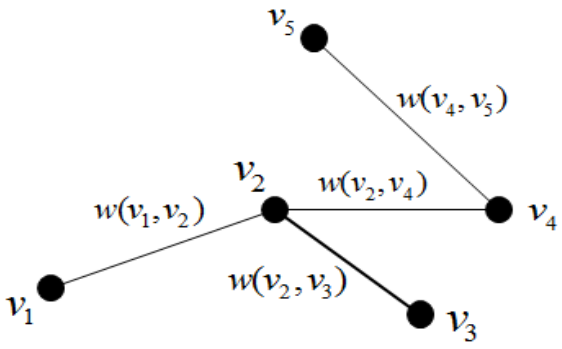
Hầu hết các thuật toán học bán giám sát dựa trên đồ thị đều dựa trên việc học lan truyền, một nhược điểm của phương pháp này là chúng ta không thể dễ dàng mở rộng thêm các điểm dữ liệu mới mà không thuộc tập $L \cup U$, hoặc nếu các điểm dữ liệu mới được thêm vào đồ thị thì sẽ làm thay đổi cấu trúc của đồ thị dẫn tới chi phí tính toán bị tăng lên. Bên cạnh đó, một lý do nữa có ảnh hưởng tới chi phí tính toán đó là phụ thuộc vào loại đồ thị sẽ xây dựng, nếu sử dụng đồ thị kết nối đầy đủ thì ta phải tính toán cho tất cả các cạnh nối giữa hai đỉnh bất kỳ.

Ở phần tiếp theo, chúng ta sẽ tập trung nghiên cứu phương pháp dựa trên xếp hạng đa tạp để khắc phục một số nhược điểm của thuật toán học lan truyền.

CHƯƠNG 2: TRA CỨU ẢNH DỰA TRÊN XẾP HẠNG ĐA TẠP

2.1 Thuật toán lan truyền nhãn

2.1.1 Ký hiệu



Hình 2-1: *Đồ thị với các trọng số cạnh*

Cho đồ thị G = (V, E, W), trong đó:

V: Tập các đỉnh, |V|=n.

E: tập các cạnh

W: Ma trận trọng số các cạnh tính theo công thức Euclid ($W_{n\times n}$)

n: Số lượng đỉnh trong G, $n=n_\ell+n_m$.

- ✓ n_ℓ**:** số đỉnh đã được gán nhãn
- ✓ n_m**:** số đỉnh chưa được gán nhãn

C: tập các nhãn, thể hiện số lượng các nhãn, với |C|=m.

P: Ma trận xác suất chuyển đổi nhãn, $P_{n\times n}$

Y: ma trận nhãn ban đầu $Y_{\ell\times C}$

f: là ma trận $f_{n\times m}$, lưu trữ thông tin của các nhãn của đỉnh đang huấn luyện. Phương pháp học nửa giám sát với đồ thị sẽ tính ma trận *f* từ các ma trận *P*, *Y*.

2.1.2 Nội dung thuật toán

Cho $\{(x_1, y_1)\ldots(x_\ell, y_\ell)\}$ là các dữ liệu đã gán nhãn, $Y_L=\{y_1, \ldots y_\ell\}$ là các nhãn của các lớp, y ={1..C} và $\{x_{\ell+1} \ldots x_{\ell+u}\}$ là các dữ liệu chưa được gán nhãn, thường $\ell \ll u$. Cho $n=\ell+u$. Chúng ta thường sử dụng L và U để thể hiện

tương ứng với tập dữ liệu đã gán nhãn và tập dữ liệu chưa gán nhãn. Giả sử rằng, số lượng các lớp C là đã biết và tất cả các lớp đã được thể hiện trong dữ liệu đã gán nhãn. Chúng ta sẽ nghiên cứu **bài toán lan truyền** cho việc tìm kiếm các nhãn cho tập U.

Bằng trực giác, chúng ta muốn các điểm dữ liệu tương tự nhau sẽ có cùng nhãn. Ta tạo ra một đồ thị đầy đủ mà các đỉnh là tất cả các điểm dữ liệu, cả dữ liệu đã gán nhãn và chưa gán nhãn. Cạnh nối bất kỳ giữa đỉnh i và đỉnh j biểu thị cho sự giống nhau của chúng. Giả sử đồ thị là đầy đủ với các trọng số sau đây, các trọng số được điều khiển bởi tham số **α**.

$$w_{ij} = \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{\alpha^2}\right) \tag{2.1}$$

hoặc cụ thể hơn

$$w_{ij} = \exp\left(-\frac{d_{ij}^2}{\alpha^2}\right) = \exp\left(-\frac{\sum_{d=1}^{D}\left(x_i^d - x_j^d\right)^2}{\alpha^2}\right) \tag{2.2}$$

Trong đó: d_{ij} là khoảng cách giữa điểm dữ liệu x_i và x_j.

Có thể lựa chọn cách tính giá trị khoảng cách khác nhau, tuy nhiên có lẽ là phù hợp nếu x là các giá trị rời rạc. Trong phạm vi thuật toán này, việc lựa chọn khoảng cách Euclid để xác định khoảng cách giữa các điểm dữ liệu và tùy theo các giá trị siêu tham số **α** cho mỗi chiều thuộc tính.

Tất cả các đỉnh có nhãn mềm có thể thay đổi nhãn trong quá trình thực hiện việc lan truyền nhãn và được hiểu là phân phối nhãn.

Chúng ta cho nhãn của một đỉnh lan truyền tới tất cả các đỉnh khác thông qua các cạnh giữa chúng. Cạnh có trọng số lớn hơn cho phép các nhãn đi qua dễ dàng hơn. Ta định nghĩa một Ma trận xác suất chuyển đổi **Pn×n**.

$$P_{ij} = P(i \to j) = \frac{w_{ij}}{\sum_{k=1}^{n} w_{ik}} \quad (2.3)$$

Trong đó $\mathbf{P_{ij}}$ là xác suất để nhảy từ đỉnh **i** tới **j**. Cũng định nghĩa một ma trận nhãn $\mathbf{Y_{L\ \ell \times C}}$ mà dòng thứ i của chúng là một véctơ chỉ số cho $\mathbf{y_i, i \in L:}$ $\mathbf{Y_{ic} = \delta(y_i, c)}$. Chúng ta sẽ tính toán nhãn mềm **f** cho các đỉnh. **f** là ma trận n × C ($\mathbf{f}_{\ n \times C}$), các hàng có thể được thể hiện như sự phân bổ xác suất trên các nhãn. Việc khởi tạo giá trị ban đầu cho f là không quan trọng. Sau đây chúng ta sẽ xem xét thuật toán:

Thuật toán:

Đầu vào: đồ thị vô hướng bao gồm các đỉnh đã gán nhãn và các đỉnh chưa gán nhãn.

Đầu ra: đồ thị vô hướng với các đỉnh đã được gán nhãn.

Thuật toán lan truyền nhãn thực hiện theo các bước sau:

- *Bước 1*. Lan truyền: $\mathbf{f \leftarrow Pf}$
- *Bước 2*. Gán (giữ lại) các dữ liệu đã gán nhãn $\mathbf{f_L = Y_L}$ **(**$\mathbf{Y_L}$ đã xây dựng ở trên**)**
- *Bước 3*. Lặp lại từ bước 1 cho tới khi **f** hội tụ.

Trong bước 1, tất cả các đỉnh lan truyền các nhãn tới các láng giềng của chúng. Bước 2 là quan trọng: chúng ta muốn giữ lại các nhãn từ dữ liệu đã gán nhãn. Vì vậy, thay vì cho việc làm các nhãn mờ đi, chúng ta giữ lại chúng ở ma trận $\mathbf{Y_L}$. Với sự hỗ trợ từ các đỉnh đã được gán nhãn, các lớp biên có thể được phân loại thông qua các vùng có tỉ trọng cao và các vùng tỉ trọng thấp.

2.1.3 Sự hội tụ của thuật toán

Bây giờ ta thể hiện sự hội tụ của thuật toán tới một lời giải đơn giản.

Cho f dưới dạng sau: $f = \begin{pmatrix} f_L \\ f_U \end{pmatrix}$

Vì f_L được gán bằng Y_L nên chúng ta chỉ quan tâm đến $\mathbf{f_U}$. (f_U chính là phần ma trận của các phần tử chưa được gán nhãn). Mục đích của chúng ta cũng là xác định ma trận $\mathbf{f_U}$ này.

Chia ma trận P thành ma trận con đã gán nhãn và ma trận con chưa gán nhãn, có dạng:

$$P = \begin{bmatrix} P_{LL} & P_{LU} \\ P_{UL} & P_{UU} \end{bmatrix} \tag{2.4}$$

Theo công thức ở trên, thuật toán có thể được mô tả như sau:

$$f_U \leftarrow P_{UU} f_U + P_{UL} Y_L \tag{2.5}$$

Dẫn đến

$$fv = \lim_{n\to\infty} (P_{UU})^n f_U^0 + \left(\sum_{i=1}^{n} (P_{UU})^{(i-1)} \right) P_{UL} Y_L \tag{2.6}$$

Trong đó, $\boldsymbol{f}^{\boldsymbol{0}}_{\boldsymbol{U}}$ là giá trị khởi tạo ban đầu của f_U. Chúng ta cần cho thấy $(P_{UU})^n f_U^0 \to 0$

Vì P có các hàng bình thường và P_{UU} là ma trận con của P, dẫn đến:

$$\exists_\gamma < 1, \sum_{j=1}^{u} (P_{UU})_{ij} \le \gamma, \forall i = 1...u \tag{2.7}$$

Do đó

$$\begin{aligned} \sum_j (P_{UU})^n_{ij} &= \sum_j \sum_k (P_{UU})^{(n-1)}_{ik} (P_{UU})_{kj} \\ &= \sum_k (P_{UU})^{(n-1)}_{ik} \sum_j (P_{UU})_{kj} \\ &= \sum_k (P_{UU})^{(n-1)}_{ik} \gamma \\ &= \gamma^n \end{aligned} \tag{2.8}$$

Do đó tổng các hàng của $(P_{UU})^n$ hội tụ về 0. Điều này có nghĩa rằng $(P_{UU})^n f_U^0 \to 0$. Do đó giá trị khởi tạo của $\boldsymbol{f}^{\boldsymbol{0}}_{\boldsymbol{U}}$ là không quan trọng. Mà $\boldsymbol{f_U = (I P_{UU})^{-1} P_{UL} Y_L}$ là cố định.

Vì vậy nó là điểm cố định duy nhất và là lời giải cho việc lặp lại thuật toán.

Điều này đưa ra cho chúng ta một cách giải quyết bài toán lan truyền nhãn một cách trực tiếp không cần lan truyền lặp lại.

Lưu ý: lời giải này khả thi khi ma trận $(1-P_{UU})$ có thể nghịch đảo được. Điều kiện này sẽ được thỏa mãn khi mọi thành phần kết nối trong đồ thị có ít nhất một điểm dữ liệu được gán nhãn.

2.1.4 Phương pháp xác định siêu tham số của đồ thị

Các thuật toán học nửa giám sát đã được áp dụng thành công trong nhiều ứng dụng với số lượng ít dữ liệu có nhãn bằng cách sử dụng các dữ liệu không có nhãn. Một vấn đề quan trọng là các thuật toán học bán giám sát dựa trên đồ thị phụ thuộc vào chất lượng của đồ thị hay siêu tham số của nó.

Phương pháp học bán giám sát dựa trên đồ thị tạo ra một đồ thị mà các đỉnh tượng trưng cho dữ liệu có nhãn và chưa có nhãn, trong khi các cạnh được thể hiện sự giống nhau giữa các cặp điểm dữ liệu. Sự phân lớp ở đây được thực hiện bằng cách sử dụng đồ thị và gán nhãn cho các dữ liệu chưa có nhãn dựa vào việc các đỉnh được kết nối bởi các cạnh có trọng số lớn hơn thì sẽ có nhãn giống nhau.

Các bộ phân lớp phụ thuộc đáng kể vào độ đo tương tự của đồ thị, thường được thực hiện theo hai bước. Bước thứ nhất, các trọng số cạnh được xác định cục bộ bằng cách sử dụng các hàm tính khoảng cách. Các hàm tính khoảng cách đóng vai trò thành phần quan trọng trong học bán giám sát dựa trên đồ thị để có được một khoảng cách tốt nhất. Bước thứ hai là bước làm mịn, được áp dụng vào toàn bộ đồ thị, điển hình là dựa trên sự lan truyền quang phổ của đồ thị Laplace.

Hiện nay, mới chỉ có một vài phương pháp tiếp cận để giải quyết vấn đề học trên đồ thị như: Học không tham số trên đồ thị Laplace, phương pháp này giả sử rằng trọng số và khoảng cách được đưa ra trước; Học có tham số

trên đồ thị sử dụng phương pháp chứng minh cực đại hóa sử dụng suy luận xấp xỉ gradient. Việc sử dụng chứng minh cực đại hóa và xấp xỉ Laplace để học các tham số đơn giản của các hàm tương tự vì chỉ học trên một đồ thị tốt, đề xuất xây dựng các đồ thị mạnh hơn bằng cách áp dụng sự xáo trộn ngẫu nhiên và bỏ đi cạnh từ một tập các cạnh trong cây bao trùm nhỏ nhất. Kết hợp đồ thị Laplace để học đồ thị. Học các băng thông khác nhau với các chiều khác nhau bằng phương pháp cực tiểu hóa Entropy trên dữ liệu chưa gán nhãn, giống như phương pháp lề cực đại trong TSVM.

Trở lại với công thức tính ma trận W trong nội dung thuật toán Lan truyền nhãn (Labeled Propagation), siêu tham số của đồ thị được ký kiệu là α. Ma trận trọng số W được đưa ra là cố định. Sau đây chúng ta nghiên cứu việc học các trọng số từ cả dữ liệu gán nhãn và dữ liệu chưa gán nhãn. Có một số phương pháp dùng để xác định siêu tham số như: Phương pháp chứng minh cực đại trong các tiến trình Gaussian (Evidence Maximization), Phương pháp Cực tiểu hóa Entropy (Entropy Minimization) và phương pháp Cây khung nhỏ nhất (Minimum spanning tree). Sau đây, chúng ta sử dụng phương pháp Cây khung nhỏ nhất, để xác định siêu tham số cho đồ thị.

Phương pháp cây khung nhỏ nhất:

Nếu các cạnh của đồ thị được đánh trọng số exp với một siêu tham số **α**, ta có thể xác định giá trị tham số **α** này theo thuật toán sau:

Ta xây dựng một cây khung nhỏ nhất trên tất cả các điểm dữ liệu với thuật toán Kruskal.

Ban đầu không có đỉnh nào được nối với nhau. Trong suốt quá trình phát triển cây, các cạnh lần lượt được xác định bởi trọng số từ ngắn đến dài.

Một cạnh được thêm vào cây nếu nó kết nối hai thành phần riêng biệt với nhau.

Quá trình lặp lại cho tới khi toàn bộ đồ thị được kết nối.

Ta tìm ra cạnh đầu tiên của cây mà kết nối hai thành phần với nhãn khác nhau. Ta coi độ dài của cạnh này là d^0 như là một giải thuật tối thiểu hóa khoảng cách giữa các vùng lớp với nhau.

Sau đó ta đặt $\boldsymbol{\alpha} = d^0/3$ theo quy tắc 3α, do đó trọng số của cạnh này sẽ gần tới 0, với hy vọng rằng việc lan truyền cục bộ chủ yếu bên trong các lớp.

2.1.5 Độ phức tạp của thuật toán

Thuật toán lan truyền nhãn được thực hiện dựa trên quá trình tính toán các ma trận và việc lặp lại để xác định sự hội tụ của thuật toán.

Đầu vào của thuật toán là một đồ thị, trong đó:

ℓ: số đỉnh đã gán nhãn.

u: số đỉnh chưa gán nhãn ($u \gg \ell$).

$n = \ell + u$: tổng số đỉnh của đồ thị.

Thuật toán thực hiện các quá trình tính toán với độ phức tạp của các thành phần như sau:

- Quá trình xác định các ma trận trọng số W, ma trận xác suất P, ma trận xác suất chuyển nhãn PUU, ma trận xác suất PUL, ma trận nhãn YL, ma trận xác suất nhãn f, có độ phức tạp:

$$O(n^2) \tag{2.9}$$

- Quá trình xác định siêu tham số α dựa trên thuật toán tìm cây khung nhỏ nhất, có độ phức tạp:

$$O(n^2 \times \log n) \tag{2.10}$$

- Quá trình lặp để thực hiện việc lan truyền nhãn được thực hiện trong m bước lặp (m khá lớn), trong đó: việc xác định sự hội tụ của thuật toán dựa trên quá trình tính toán các định thức ma trận, các phép toán nhân ma trận và tìm ma trận nghịch đảo, có độ phức tạp: $O(n^3)$.

Do đó, độ phức tạp của quá trình lặp là:

$$O(m \times n^3) \quad (2.11)$$

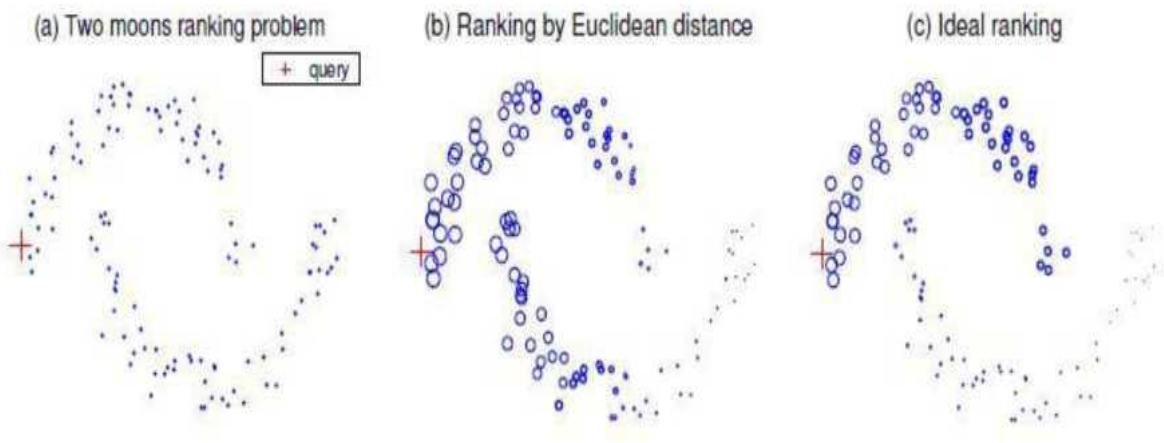
Từ (2.9), (2.10) và (2.11), suy ra độ phức tạp của thuật toán lan truyền nhãn là: $O(m \times n^3)$

2.2 CBIR dựa trên Xếp hạng đa tạp

2.2.1 Giới thiệu

Thông thường, một hệ thống CBIR sử dụng phương pháp cảm nhận từng cặp (ví dụ như xác định khoảng cách Euclide) để đo sự tương tự giữa ảnh truy vấn và mỗi ảnh cơ sở dữ liệu. Mặt khác, hệ thống dựa trên xếp hạng đa tạp CBIR dựa vào một phương pháp giữa hình ảnh truy vấn và cơ sở dữ liệu truy vấn hình ảnh để khai thác các mối quan hệ liên quan của tất cả các điểm dữ liệu trong không gian đặc trưng nhất định, đồng thời xếp hạng điểm số của các ảnh được dán nhãn để ảnh được gán nhãn lan truyền đến ảnh không có nhãn qua một đồ thị có trọng số [3].

Rất nhiều dữ liệu thực được biểu diễn một cách phù hợp trong không gian cấu trúc đa tạp cục bộ hơn là trong những không gian khác, giống như khoảng cách Minwoski dựa trên không gian cấu trúc. Hình 2.2 biểu diễn ví dụ với một vài cấu trúc phù hợp. Trong ví dụ này, một tập hợp các điểm mẫu tạo thành một mô hình 2 moon. Giả sử rằng một truy vấn trong moon trên được đưa ra, nhiệm vụ là để xếp hạng các điểm còn lại phù hợp với truy vấn. Ta có thể dễ dàng cho rằng tất cả các điểm trong moon trên là phù hợp hơn với các điểm truy vấn trong moon dưới. Tuy nhiên, nếu ta đo sự giống nhau của các điểm truy vấn trong không gian Euclide, các điểm dưới bên trái của moon dưới giống với các truy vấn hơn với mức điểm trên bên phải của moon trên. Rõ ràng, kết quả này thể hiện trong Hình 2.2 (b) là không thỏa đáng nếu dựa trên nhận thức của con người.



Hình 2-2: *Ví dụ minh họa ưu điểm của kỹ thuật đa tạp*
(a) dữ liệu được thiết lập với một truy vấn duy nhất được đánh dấu bằng một dấu cộng ; (b) khoảng cách Euclide dựa trên kết quả xếp hạng; (c) Kết quả nhận thức dựa trên xếp hạng lý tưởng. Trong cả (b) và (c), dấu chấm rỗng lớn hơn đại diện cho các kết quả xếp hạng.

2.2.2 Học truyền dẫn trong CBIR

2.2.2.1 Kỹ thuật xếp hạng đa tạp cơ bản

Xếp hạng dữ liệu trong cấu trúc đa tạp thuộc mô hình học bán giám sát. Với một giả định rằng mỗi điểm dữ liệu trong một không gian đặc trưng có một mối quan hệ với các điểm dữ liệu khác trong không gian tương tự, nên có một cạnh để kết nối mỗi cặp điểm, nơi cạnh được gán một trọng số để biểu diễn mối liên quan dữ liệu giưa hai điểm. Do đó, hệ thống trước hết xây dựng một đồ thị có trọng số cho tất cả các điểm dữ liệu trong không gian đặc trưng. Sau đó, điểm dữ liệu truy vấn ban đầu được gán một giá trị nhất định, các điểm dữ liệu còn lại có liên quan được gán giá trị 0. Thứ hai, tất cả các điểm dữ liệu lan truyền xếp hạng của chúng đến các điểm dữ liệu bên cạnh thông qua các đồ thị có trọng số. Quá trình lan truyền của các điểm số xếp hạng lặp đi lặp lại cho đến khi hội tụ tới một tình trạng ổn định toàn cục. Các điểm chính thức được xếp hạng đại diện cho việc giống nhau giữa điểm dữ liệu và điểm truy vấn. Các điểm dữ liệu tương tự như các điểm truy vấn là những điểm xếp hạng lớn nhất.

Zhou và các cộng sự giải thích thuật toán basic manifold ranking như sau. Cho 1 tập điểm: $\chi=\{x_1,\ldots,x_q, x_{q+1},\ldots,x_n\} \subset \mathbb{R}^p$, ở đây n là số điểm. q là điểm truy vấn và toàn bộ các điểm còn lại được xếp hạng theo liên quan của chúng với điểm truy vấn. Với mỗi cặp điểm x_i và x_j, thì một khoảng cách $d(x_i,x_j)$ được định nghĩa như sau d: χ x $\chi \rightarrow \mathbb{R}$, đó là độ đo metric của tập hợp điểm χ. Khoảng cách có thể là Euclidean hoặc khoảng cách Manhattan… . Tương ứng, mỗi điểm x_i có giá trị xếp hạng f_i được định nghĩa: $f_i=f(x_i)$ nơi f: $\chi \rightarrow \mathbb{R}$. Cuối cùng 1 vector $y=[y_1,\ldots,y_n]^T$ được định nghĩa trong đó $y_i=1$ nếu x_i là một truy vấn, nếu không thì $y_i=0$.

Thuật toán xếp hạng đa tạp cơ bản được tóm tắt như sau [3]:

- *Bước 1*: Sắp xếp các khoảng cách Euclidean theo thứ tự tăng dần. Nối hai điểm liên tiếp với một cạnh theo thứ tự cho đến khi một đồ thị liên thông thu được.
- *Bước 2:* Mô hình ma trận W được định nghĩa bởi

$$W_{ij} = \exp[-d^2(x_i, x_j) / 2\sigma^2] \qquad (2.12)$$

- trong đó $d(x_i, x_j)$ là khoảng cách Euclide giữa điểm x_i và điểm x_j, σ là sai số chung của χ . Lưu ý rằng $W_{ij} = 0$ vì không có vòng lặp trong đồ thị.
- *Bước 3:* Chuẩn hóa W được xác định bởi $S=D^{-1/2}WD^{-1/2}$. Trong đó trong đó D là ma trận đường chéo với với (i,i) - nguyên tố là tổng của hàng thứ i của W.
- *Bước 4:* Lặp $f(t+1)= \alpha Sf(t)+(1- \alpha)y$ cho đến khi hội tụ, trong đó α là một tham số trong [0,1).
- *Bước 5*: Để *fi** biểu thị giới hạn của dãy $\{f_i(t)\}$. Sắp xếp mỗi điểm x_i theo điểm số xếp hạng của nó với F^*_i (điểm số lớn nhất sẽ được xếp hạng đầu tiên).

Trước hết, một đồ thị được xây dựng để kết nối tất cả các điểm trong cơ sở dữ liệu. Sau đó, các cạnh trong đồ thị này được gán trọng số tương ứng của phương trình (2.12). Công việc được thực hiện trong bước 3 để đảm bảo hội tụ. Tiếp theo, các điểm được xếp hạng theo điểm số hạng cuối cùng..và các điểm đó được sắp xếp theo tỷ lệ cuối cùng. Ở đây, tham số α xác định sự đóng góp vào điểm số xếp hạng từ các đỉnh lân cận (1-α) và xác định sự đóng góp vào điểm số xếp hạng ban đầu. Điểm số xếp hạng là lan truyền đối xứng vì S là ma trận đối xứng.

Theo Cox và các cộng sự, hàm {*f(t)*} ở bước 4 hội tụ đến

$$f^* = \beta(1-\alpha S)^{-1} y \tag{2.13}$$

Trong đó β là một yếu tố phổ biến cho mọi điểm khi tính điểm xếp hạng và đặt theo công thức *1-*α. Kết quả là β có thể được bỏ qua trong tính toán số điểm xếp hạng, và phương trình (2.13) có thể được đơn giản hóa như sau:

$$f^* = (1-\alpha S)^{-1} y \tag{2.14}$$

Biến thể của kỹ thuật xếp hạng đa tạp khi sửa đổi bước 2 hoặc bước 4 hoặc cả hai bước. Trong các phần dưới đây, ta có một số biến thể đại diện của các hệ thống đa tạp mà chỉ sửa đổi bước 4.

2.2.3 Học truyền dẫn với phản hồi liên quan

2.2.3.1 Lan truyền với chỉ mẫu dương

Trong các thuật toán dựa trên xếp hạng đa tạp cơ bản, điểm số xếp hạng lan truyền được lặp đi lặp lại cho đến khi ổn định toàn cục. Trong mỗi lần lặp, hệ thống tích hợp thông tin phản hồi của người sử dụng cho lần kế tiếp của việc lan truyền điểm xếp hạng. Khi người dùng chỉ gửi phản hồi tích cực (phản hồi +) cho các mẫu thông tin trả lại, hoặc khi chỉ có những hình ảnh có liên quan đến truy vấn có liên quan, mẫu thông tin tích cực mới trở lại được thêm vào các thiết lập truy vấn và lan truyền điểm xếp hạng sẽ liên tục tinh

chỉnh các kết quả thu hồi. Để kết thúc phần này, phương trình (2.14) có thể được sửa đổi như sau:

$$f^* = (1-\alpha S)^{-1} y = (1-\alpha S)^{-1} \sum_{i-1}^{n^+ +1} y^i \quad (2.15)$$

Trong đó y^i là là một vector n chiều với thành phần thứ i bằng 1 và những thành phần khác bằng 0, n^+ là số mẫu tin về phản hồi tích cực (mẫu +). Nói cách khác, các thành phần khác không trong y tương ứng với ảnh trả lại có nhãn tích cực (mẫu +) và đóng góp vào sự lan truyền của bảng xếp hạng điểm số trong quá trình tạo ra.

2.2.3.2 Lan truyền với mẫu dương và mẫu âm

Kể từ khi thông tin phản hồi của người sử dụng có thể chứa cả kết quả mẫu dương và mẫu âm đối với các mẫu thông tin lấy ra, một số hệ thống đa tạp sử dụng cả hai mẫu thông tin lan truyền các nhãn dựa trên hai quan sát sau đây:

- hình ảnh liên quan có xu hướng hình thành các cụm nào đó trong không gian đặc trưng.
- hình ảnh không thích hợp có thể hình thành một số cụm dữ liệu khác nhau với ngữ nghĩa khác nhau.

Những hệ thống này cho rằng các kiến thức đã học từ cả hai hình ảnh có liên quan và không liên quan là hữu ích để tinh chỉnh kết quả thu hồi và đạt được một kết quả thu hồi khá chính thức. Để đáp ứng với phản hồi mẫu dương và mẫu âm, hai vector y^+ và y^- được đưa ra. Các vector đầu tiên tương tự như các vector được định nghĩa trước $y^+ = \left[y_1^+, ..., y_n^+ \right]^T$, mà các thành phần được thiết lập là 1 nếu các hình ảnh tương ứng là các truy vấn chính nó hoặc một hình ảnh trở về có gắn nhãn dương.. Các yếu tố trong vector thứ hai $y^- = \left[y_1^-, ..., y_n^- \right]^T$ được thiết lập là -1 nếu các hình ảnh tương ứng là một hình ảnh trở về có gắn nhãn âm. Tất cả các thành phần còn lại trong cả hai vectơ được thiết lập là 0. Phương trình (2.15) có thể được định nghĩa như sau:

$$f^* = f^{+*} + f^{-*} = Ay^+ + Ay^- \quad (2.16)$$

Trong đó $A = (1- \alpha S)^{-1}$, f^{+*} và f^{-*} là điểm số xếp hạng thu được từ các phản hồi dương và âm tương ứng.

2.2.3.3 Lan truyền với Phản hồi dương và phản hồi âm có trọng số

Hơn nữa, các trọng số khác nhau có thể được áp dụng cho phương trình (2.16) dựa trên hai quan sát sau đây:

- Một hình ảnh không có nhãn nằm càng xa từ dữ liệu tích cực trong không gian đặc trưng càng ít khả năng nó là mẫu dương.
- Nếu một hình ảnh không có nhãn nằm xa ví dụ tiêu cực trong không gian đặc trưng thì khả năng mẫu dương là không chắc chắn, vì nó có thể không được gần với các dữ liệu tích cực.

Do đó, dữ liệu tích cực nói chung đóng góp nhiều hơn cho các điểm số xếp hạng cuối cùng so với dữ liệu tiêu cực, và phương trình (2.16) có thể được điều chỉnh như sau:

$$f^* = f^{+*} + \gamma * f^{-*} = Ay^+ + \gamma * Ay^- \quad (2.17)$$

Ở đây, tham số $\gamma \in (0,1]$ làm suy yếu sự đóng góp của các điểm số xếp hạng tiêu cực đến f^{-*}. γ càng nhỏ thì càng tác động ít đến nhãn tiêu cực dữ liệu trong bảng xếp hạng điểm số cuối cùng. Khi $\gamma = 1$ thì sự đóng góp dữ liệu nhãn tiêu cực tương tự như dữ liệu nhãn tích cực trong việc lan truyền các điểm số xếp hạng. Hệ thống này sẽ trở thành hệ thống lan truyền với phản hồi tích cực và phản hồi tiêu cực như đã giải thích trong phần 2.2.3.2.

Từ những phân tích trên ba biến thể dự trên xếp hạng đa tạp đã rõ ràng để khẳng định rằng các biến thể Lan truyền với phản hồi mẫu dương và mẫu âm có trọng số là tốt hơn so với 2 biến thể còn lại bởi vì nó sử dụng cả thông tin phản hồi tích cực và phản hồi tiêu cực từ người dùng đồng thời xử lý với hai loại thông tin phản hồi với sự nhấn mạnh khác nhau trong việc tính điểm

xếp hạng. Do đó, tôi đề xuất hệ thống được xây dựng trên nền tảng của biến thể Lan truyền với phản hồi mẫu dương và phản hồi mẫu âm có trọng số.

2.3 Kỹ thuật xếp hạng đa tạp cải tiến

2.3.1 Giới thiệu

Mặc dù xếp hạng đa tạp (Manifold) đã được sử dụng rộng rãi trong nhiều ứng dụng, tuy nhiên đây là khó khăn cần giải quyết để xử lý cơ sở dữ liệu quy mô lớn vì vậy đã làm hạn chế đáng kể khả năng ứng dụng của nó.

Đầu tiên là phương pháp xây dựng đồ thị của xếp hạng đa tạp. Đồ thị *k*NN là khá thích hợp để xếp hạng đa tạp vì khả năng tốt để nắm bắt cấu trúc cục bộ của dữ liệu. Nhưng xây dựng đồ thị *k*NN có chi phí là $O(n^2 \log k)$ do đó rất mất thời gian xử lý trong các trường hợp quy mô lớn. Hơn thế nữa, việc xếp hạng đa tạp cũng như nhiều thuật toán dựa trên đồ thị khác trực tiếp sử dụng ma trận kề W trong việc tính toán. Chi phí lưu trữ của một ma trận thưa W là *O(kn)*. Như vậy, chúng ta cần phải tìm cách để xây dựng một đồ thị sao cho thời gian xây dựng thấp và không gian lưu trữ nhỏ cũng như khả năng tốt để nắm bắt cấu trúc cơ bản cho cơ sở dữ liệu.

Thứ hai, việc xếp hạng đa tạp rất mất thời gian tính toán vì khai thác ma trận nghịch đảo trong phương trình (2.14).

Điều này là trọng tâm chính để áp dụng xếp hạng đa tạp trong các ứng dụng quy mô lớn. Mặc dù chúng ta có thể sử dụng các thuật toán lặp trong phương trình $f(t+1)=\alpha Sf(t)+(1-\alpha)y$, tuy nhiên vẫn còn kém hiệu quả trong các trường hợp quy mô lớn để có thể đi đến một hội tụ cục bộ. Vì vậy, ban đầu việc xếp hạng đa tạp là không đủ cho một hệ thống truy cập thời gian thực.

Để khắc phục hạn chế của xếp hạng đa tạp truyền thống. Bin Xu và các cộng sự đã mở rộng và đề xuất phương pháp xếp hạng đa tạp hiệu quả (Efficient Manifold Ranking - EMR) [2]. Phương pháp này tập trung giải quyết hai vấn đề chính: thứ nhất đó là xây dựng đồ thị Anchor có chi phí tính

toán nhỏ thay vì đồ thị *k*-NN truyền thống, thứ hai là xây dựng mô hình tính toán xếp hạng đa tạp hiệu quả bằng cách thiết kế hình thức mới của ma trận kề nhằm tăng tốc độ tính toán xếp hạng. Mô hình này có hai chiến lược tách rời: thứ nhất là chiến lược off-line để xây dựng mô hình xếp hạng, thứ hai là chiến lược on-line để xử lý một truy vấn mới.

Với đề xuất này, hệ thống có thể xử lý một cơ sở dữ liệu với 1 triệu ảnh và có thể tra cứu trực tuyến trong một thời gian ngắn.

2.3.2 Xây dựng đồ thị

Để xử lý các cơ sở dữ liệu lớn thì cần thiết phải xây dựng đồ thị có chi phí tính toán là tuyến tính nhỏ (sub-linear) với kích thước đồ thị. Điều đó có nghĩa là đối với mỗi điểm thì không thể tìm kiếm toàn bộ cơ sở dữ liệu như chiến lược *k*NN đã làm. Để đạt được yêu cầu này cần xây dựng đồ thị Anchor và đề xuất một thiết kế mới cho ma trận kề W.

Các định nghĩa điểm Anchor và đồ thị Anchor đã xuất hiện trong một số nghiên cứu khác nhau. Một số tác giả đề xuất mỗi điểm dữ liệu trên đa tạp có thể xấp xỉ cục bộ bằng tổ hợp tuyến tính của các điểm Anchor láng giềng của nó và trọng số tuyến tính trở thành mã hóa phối hợp cục bộ. Liu và các cộng sự thiết kế ma trận kề trong một độ đo xác suất và sử dụng nó cho khả năng mở rộng học bán giám sát.

2.3.2.1 Xây dựng đồ thị Anchor

Giả sử ta có một tập dữ liệu $\chi=\{x_1,...,x_n\}\subset R^m$ với n mẫu trong không gian m chiều và $U=\{u_1,...,u_d\}\subset R^m$ là một tập hợp các điểm Anchor phân bố trong cùng một không gian tương tự với tập dữ liệu. Ký hiệu $f:\chi\to R$ là một hàm giá trị thực để gán mỗi điểm dữ liệu trong χ với một nhãn ngữ nghĩa . Mục đích là tìm một ma trận trọng số $Z\in R^{d\,x\,n}$ để biểu diễn mối quan hệ tiềm năng giữa các điểm dữ liệu trong χ và các điểm Anchor trong U. Sau đó giá trị *f(x)* được ước lượng cho mỗi điểm dữ liệu như là một trung bình trọng số của các nhãn trên Anchor.

$$f(x_i) = \sum_{k=1}^{d} z_{ki} f(u_k), i = 1, ..., n, \quad (2.18)$$

với các ràng buộc $\sum_{k=1}^{d} z_{ki} = 1$ và $z_{ki} \geq 0$. Phần tử z_{ki} biểu diễn trọng số giữa điểm dữ liệu x_i và điểm Anchor u_k. Điểm chính của việc xây dựng đồ thị Anchor là làm thế nào để tính toán vector trọng số z_i cho mỗi điểm dữ liệu x_i. Hai vấn đề cần được xem xét đó là số lượng của các vector trọng số và chi phí của việc tính toán. Tương tự như ý tưởng của LLE, một cách đơn giản để đo trọng số cục bộ là tối ưu hóa các vấn đề còn tồn tại sau đây:

$$\begin{aligned} &\min_{z_i} \varepsilon(z_i) = \frac{1}{2} \left\| x_i - \sum_{s=1}^{|N(x_i)|} u_{s \in N(x_i) Z_{is}} \right\|^2 \\ &s.t \quad \sum_s z_{is} = 1, \quad z_i \geq 0 , \end{aligned} \quad (2.19)$$

Ở đây $N(x_i)$ là tập chỉ số các điểm Anchor gần nhất của x_i. Bài toán trên được gọi là bài toán *ước lượng trọng số cục bộ*. Bài toán này có thể được giải bằng cách sử dụng quy hoạch toàn phương chuẩn (QP), tuy nhiên việc tính toán QP rất tốn nhiều thời gian. Do đó một số phương pháp đã được đề xuất để giải phương trình trên đó là phương pháp dựa trên phép chiếu Gradient được đề xuất để tính toán ma trận trọng số và phương pháp hồi quy hạt nhân (Kernel Regression) đã được áp dụng. Hai phương pháp này đều nhanh hơn QP. Trong nghiên cứu sau đây, ta so sánh hai phương pháp khác nhau để tìm ra trọng số vector z_i..

Giải quyết bằng phép chiếu Gradient

Phương pháp đầu tiên là phương pháp phép chiếu Gradient. Các quy luật cập nhật trong phương pháp này được giải thích theo công thức lặp như sau:

$$z_i^{(t+1)} = \Pi_s(z_i^{(t)} - \eta_t \nabla \varepsilon(z_i^t)) \quad (2.20)$$

Trong đó η_t biểu thị khoảng bước thời gian *t*, $\nabla\varepsilon(z)$ biểu thị độ chênh lệch của ε tại *z*, và $\prod_s(z)$ là toán tử biểu thị phép chiếu đơn hình bất kỳ trên $z \in \square^s$. Thuật toán như sau:

Input: A vector $z \in \square^s$

sort z into v such that $v_1 \geq v_2 \geq .... \geq v_s$

find $\rho = max\left\{ j \in [1:s] : v_j - \frac{1}{j}\left(\sum_{r=1}^{j} v_r - 1 \right) > 0 \right\}$

compute $\theta = \frac{1}{\rho}\left(\sum_{j=1}^{\rho} v_j - 1 \right)$

Output: A vector $z' = [z'_1, ..., z'_s]^T$ such that $Z'_j = max\{z_j - \theta, 0\}$

Giải quyết bằng hồi quy hạt nhân (Kernel Regression)

Trong phương pháp này, hồi quy hạt nhân Nadaraya-Watson đã được sử dụng để gán các trọng số theo công thức sau:

$$z_{ki} = \frac{K\left(\frac{|x_i - u_k|}{\lambda} \right)}{\sum_{l=1}^{d} K\left(\frac{|x_i - u_l|}{\lambda} \right)} \tag{2.21}$$

với các hạt nhân bậc hai Epanechnikov

$$K_\lambda(t) = \begin{cases} \frac{3}{4}(1-t^2) & \text{if } |t| \leq 1; \\ 0 & otherwise \end{cases} \tag{2.22}$$

Các thông số làm mịn λ xác định kích thước của vùng cục bộ, trong đó các điểm Anchor có thể phản ánh được những điểm mục tiêu. Đó là lý do để xem xét rằng những điểm dữ liệu có nhãn ngữ nghĩa tương tự với các điểm Anchor lân cận trong một xác suất cao. Có rất nhiều cách để xác định các tham số λ. Ví dụ, nó có thể là một hằng số qua kiểm chứng chéo từ một tập

hợp các dữ liệu huấn luyện. Trong nghiên cứu này, Bin Xu và cộng sự đã sử dụng một cách hiệu quả hơn để nhận được λ bằng cách sử dụng láng giềng gần nhất kích thước *s* để thay thế λ, đó là:

$$\lambda(x_i) = \left| x_i - u_{[s]} \right| \tag{2.23}$$

Ở đây, $u_{[s]}$ là điểm Anchor gần nhất thứ *s* của x_i.

Cụ thể, để xây dựng đồ thị Anchor, ta kết nối mỗi mẫu để cố định điểm gần nhất và sau đó gán các trọng số. Vì vậy việc xây dựng có độ phức tạp O(*nd* log *s*), trong đó *d* là số điểm Anchor và *s* là rất nhỏ. Như vậy, số lượng các điểm Anchor xác định hiệu quả của việc xây dựng đồ thị Anchor. Nếu $d << n$ thì việc xây dựng là tuyến tính với cơ sở dữ liệu.

Làm thế nào mà có thể có được các điểm Anchor? Học chủ động hoặc phương pháp phân cụm là các giải pháp đáng kể cần lựa chọn. Trong hướng nghiên cứu này, các tác giả đã sử dụng thuật toán *k*-means để lựa chọn tâm như các điểm Anchor. Một số các thuật toán *k*-means fast có thể tăng tốc độ tính toán. Lựa chọn ngẫu nhiên là một phương pháp cần tính tới vì có chi phí lựa chọn cực kỳ thấp và hiệu suất chấp nhận được.

Đặc điểm chính cũng là lợi thế chính của xây dựng đồ thị Anchor đó là phân tách việc xây dựng đồ thị thành hai phần: lựa chọn các điểm Anchor và xây dựng đồ thị. Mỗi mẫu dữ liệu là độc lập với các mẫu khác nhưng chỉ liên quan đến các điểm Anchor. Việc xây dựng luôn là hiệu quả vì nó có độ phức tạp tuyến tính với kích cỡ của cơ sở dữ liệu. Lưu ý rằng chúng ta không cần phải cập nhật thường xuyên các điểm Anchor vì các điểm Anchor tương đối ổn định như thông tin cho cơ sở dữ liệu lớn (ví dụ các tâm cụm) ngay cả khi một vài mẫu mới được thêm vào.

2.3.2.2 Thiết kế Ma trận kề

Trong phần này, Bin Xu và các cộng sự đề xuất một cách tiếp cận mới để thiết kế ma trận kề *W* và giải thích một cách trực quan. Các ma trận trọng

số $Z \in R^{d \times n}$ có thể được xem như là một đại diện chiều d của dữ liệu $X \in R^{m \times n}$, d là số điểm Anchor. Đó là các điểm dữ liệu có thể được biểu diễn trong không gian mới mà vẫn có các tính năng ban đầu. Đây là một lợi thế lớn để xử lý một số dữ liệu có kích thước lớn. Sau đó, với các kết quả bên trong khi số liệu để tính khối lượng giáp ranh giữa các điểm dữ liệu, ta thiết kế ma trận kề là một hình thức cấp bậc thấp

$$W = Z^T Z \tag{2.24}$$

điều đó có nghĩa rằng nếu hai điểm dữ liệu tương ứng ($W_{ij} > 0$), ta chia sẻ ít nhất một điểm Anchor thông thường, ngược lại nếu $W_{ij} = 0$. Bằng cách chia sẻ cùng các điểm Anchor, điểm dữ liệu có khái niệm ngữ nghĩa tương tự trong một xác suất cao như đã nghiên cứu. Vì vậy, thiết kế là hữu ích để khám phá các mối quan hệ ngữ nghĩa trong dữ liệu.

Công thức này tự nhiên duy trì một số đặc tính tốt của W đó là thưa thớt và không tiêu cực. Các ma trận thưa Z mức độ cao làm cho W thưa thớt, phù hợp với các quan sát rằng hầu hết các điểm trong một đồ thị chỉ có một lượng nhỏ các cạnh với các điểm khác. Các thuộc tính không âm làm cho trọng số kế cận có ý nghĩa hơn: trong dữ liệu thực tế, mối quan hệ giữa hai hạng mục luôn luôn là tích cực hay không tích cực, nhưng không tiêu cực. Hơn nữa, W không âm đảm bảo các tính xác định bán tích cực của đồ thị Laplacian trong nhiều thuật toán dựa trên đồ thị.

2.3.3 Tính toán xếp hạng

Sau khi xây dựng đồ thị, các chi phí tính toán chính để xếp hạng đa tạp là đảo ma trận trong phương trình (2.14) có độ phức tạp là $O(n^3)$. Vì vậy, kích thước dữ liệu n không thể quá lớn. Mặc dù ta có thể sử dụng các thuật toán lặp những vẫn không hiệu quả đối với trường hợp quy mô lớn.

Ta có thể lập luận rằng các ma trận đảo có thể được thực hiện off-line. Tuy nhiên, off-line tính chỉ có thể xử lý các trường hợp khi truy vấn là đã có trong đồ thị (trong mẫu). Nếu truy vấn không phải là trong đồ thị (một mẫu

ngoài), cho cấu trúc đồ thị chính xác, ta có thể cập nhật toàn bộ đồ thị để thêm các truy vấn mới và tính nghịch đảo ma trận trong phương trình (2.14) một lần nữa. Do đó, việc tính toán off-line không làm việc cho một mẫu truy vấn thiếu. Trên thực tế, đối với một hệ thống CBIR truy vấn thực của người dùng luôn luôn là một mẫu truy vấn thiếu.

Với các dạng của $W = Z^{\mathrm{T}}Z$, ta có thể viết lại phương trình (2.14), các bước chính của bảng xếp hạng đa tạp theo công thức như Woodbury sau. Hãy cho $H = ZD^{-\frac{1}{2}}$ và $S = H^{\mathrm{T}}H$, sau đó Hàm xếp hạng cuối cùng f có thể được tính trực tiếp bằng:

$$f^* = (I_n - \alpha H^T H)^{-1} y = \left(I_n - H^T \left(HH^T - \frac{1}{\alpha} I_d \right)^{-1} H \right) y \qquad (2.25)$$

Bằng phương trình (2.25), phần đảo ngược (lấy chi phí tính toán nhất) thay đổi từ một ma trận $n \times n$ với một ma trận $d \times d$. Nếu $d << n$ thì sự sự thay đổi này có thể tăng tốc độ của việc tính toán xếp hạng đa dạng. Do đó, việc áp dụng phương pháp đề xuất của ta đến một hệ thống truy cập thời gian thực là khả thi.

Trong quá trình tính toán, ta không bao giờ sử dụng ma trận kề W. Vì vậy không lưu ma trận W trong bộ nhớ nhưng lưu ma trận Z để thay thế. Trong phương trình (2.25), D là một ma trận đường chéo với $D_{ii} = \sum_{j=1}^{n} \mathrm{w_{ij}}$ khi $W = Z^{\mathrm{T}}Z$

$$Dii = \sum_{j=1}^{n} z_i^T z_j = z_i^T v \qquad (2.26)$$

trong đó z_i là cột thứ i của ma trận Z và $v = \sum_{j=1}^{n} z_j$. Như vậy chúng ta có được ma trận D mà không sử dụng ma trận W.

Một thủ thuật hữu ích cho việc tính toán f^* trong phương trình (2.25) đang chạy từ phải sang trái. Vì vậy, mỗi khi ta nhân một ma trận với một

vector (tránh ma trận nhân ma trận). Từ đó ta tính các hàm xếp hạng EMR có phức tạp $O(d_n+d^3)$.

2.3.4 Phân tích độ phức tạp

Trong phần này, ta thực hiện một phân tích phức tạp toàn diện của MR và EMR, bao gồm cả các chi phí tính toán và chi phí lưu trữ. Như trên đã đề cập, cả MR và EMR có hai giai đoạn đó là giai đoạn xây dựng đồ thị và giai đoạn tính toán xếp hạng.

Đối với mô hình MR:

- MR xây dựng một đồ thị *k*NN, tức là khi cho mỗi mẫu dữ liệu, chúng ta cần phải tính toán các mối quan hệ để *k*-láng giềng gần nhất. Vì vậy, chi phí tính toán là $O(n^2 \log k)$. Đồng thời ta lưu lại ma trận kề $W \in R^{n \times n}$ với một chi phí lưu trữ $O(kn)$ vì W là đồ thị thưa.
- Trong giai đoạn tính toán xếp hạng, các bước chính là tính nghịch đảo trong 2 ma trận, đó có khoảng $O(n^3)$.

Đối với mô hình EMR:

- EMR xây dựng một đồ thị cố định. Tức là, cho mỗi mẫu dữ liệu, ta tính toán các mối quan hệ để xác định *s*- khu vực gần điểm cố định của nó. Chi phí tính toán là $O(nd \log s)$. Ta sử dụng *k-means* để chọn điểm cố định, vì vậy cần một chi phí $O(Tdn)$, trong đó T là số lần lặp. Nhưng bước lựa chọn này có thể được thực hiện off-line và không cần thiết được cập nhật thường xuyên. Đồng thời, chúng ta lưu các ma trận thưa $Z \in R^{d \times n}$ với một chi phí lưu trữ $O(sn)$.
- Trong giai đoạn tính toán xếp hạng, các bước chính là phương trình (2.25), trong đó có một tính toán phức tạp $O(d_n + d^3)$.

Kết quả là, EMR có chi phí tính toán $O(d_n) + O(d^3)$ (bỏ qua s, T) và chi phí lưu trữ $O(sn)$, trong khi MR có chi phí tính toán $O(n^2) + O(n^3)$ và một chi

phí lưu trữ *O(kn)*. Rõ ràng, khi *d*<<*n* thì EMR có chi phí thấp hơn nhiều so với MR trong tính toán.

Kết luận chương 2

Trong chương này, chúng ta đã tìm hiểu về phương pháp học bán giám sát dựa trên đồ thị theo thuật toán lan truyền nhãn. Đặc biệt là phương pháp dựa trên xếp hạng đa tạp. Đây là phương pháp khá quan trọng trong việc xây dựng bài toán tra cứu ảnh dựa trên học bán giám sát. Đồng thời cũng đã nghiên cứu Kỹ thuật xếp hạng đa tạp cải tiến để tăng tốc độ tra cứu và giảm thời gian tính toán khi áp dụng vào việc tra cứu ảnh cho cơ sở dữ liệu lớn.

Trong chương sau, chúng ta sẽ tiến hành cải đặt thử nghiệm chương trình tra cứu ảnh dựa trên nội dung theo mô hình học bán giám sát trên đồ thị qua thuật toán xếp hạng đa tạp (MR) và thuật toán xếp hạng đa tạp cải tiến (EMR) . So sánh hiệu năng của hai thuật toán này.

CHƯƠNG 3: THỰC NGHIỆM

3.1 Môi trường thực nghiệm

3.1.1 Cơ sở dữ liệu

Chương trình được xây dựng bằng bộ công cụ Matlab 7.11.0 (R2010b). Tập dữ liệu sử dụng trong thử nghiệm là tập con của cơ sở dữ liệu Corel Photo Gallery để đánh giá hiệu năng của phương pháp đề xuất. Tập ảnh dữ liệu thử nghiệm bao gồm 10800 ảnh được tổ chức thành 80 lớp: autumn, aviation bonsai, castle, cloud, dog, elephant, iceberg, primates, ship, tiger... (như trong Hình 3-1), mỗi lớp gồm khoảng 100 ảnh. Các ảnh trong cơ sở dữ liệu có kích cỡ là 128× 85 điểm ảnh hoặc 85×128 điểm ảnh. Đây là tập dữ liệu thường được dùng để thực nghiệm tra cứu ảnh.

Hình 3-1: *Tập cơ sở dữ liệu ảnh COREL*

3.1.2 Trích chọn đặc trưng

Để trích chọn đặc trưng ảnh, chương trình thực nghiệm trích chọn 3 kiểu đặc trưng: Mầu sắc, kết cấu và hình dạng như sau:

Với đặc trưng mầu sắc, chương trình thực nghiệm lựa chọn những momen mầu. Trước hết, tiến hành chuyển đổi không gian mầu từ RGB thành HSV. Sau đó, 3 momen mầu: color mean, color variance và color skewness trong mỗi kênh màu sẽ được trích chọn. Do đó một vector 9 chiều mô tả momen màu sẽ được sử dụng.

Với đặc trưng kết cấu, biến đổi wavelet rời rạc được thực hiện trên ảnh đa mức xám. Mỗi phân tích wavelet trên ảnh 2 chiều sẽ trả về 4 ảnh con thu nhỏ. Với phân tích ba mức được thực hiện và các đặc trưng được trích chọn từ 9 ảnh con, ta thu được một vector đặc trưng 9 chiều biểu diễn cho mỗi ảnh.

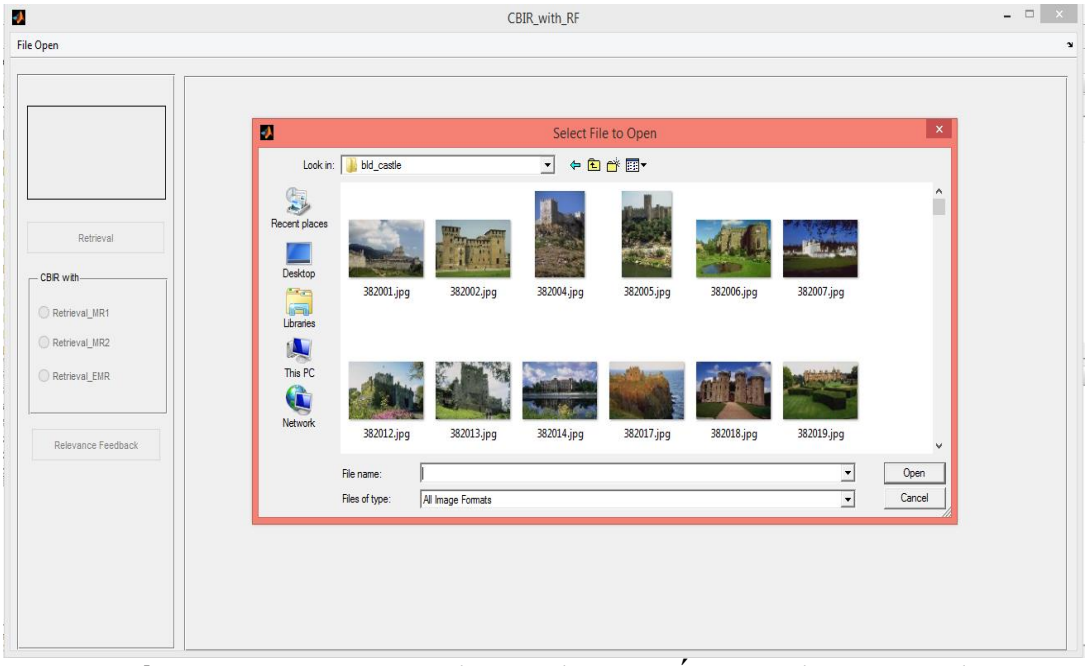
Với đặc trưng hình dạng, biểu đồ hướng cạnh được sử dụng. Thông tin cạnh chứa trong ảnh được tạo ra và xử lý. Biểu đồ hướng cạnh sau đó được lượng tử hóa thành 18 bin với 20 độ đo cho mỗi bin. Do đó tổng số 18 đặc trưng cạnh được trích chọn.

Tất cả các đặc trưng này được kết hợp thành một vector đặc trưng 36 chiều và sau đó được chuẩn hóa thành phân bố chuẩn để loại bỏ sự ảnh hưởng của co dãn.

3.2 Mô tả chương trình thực nghiệm

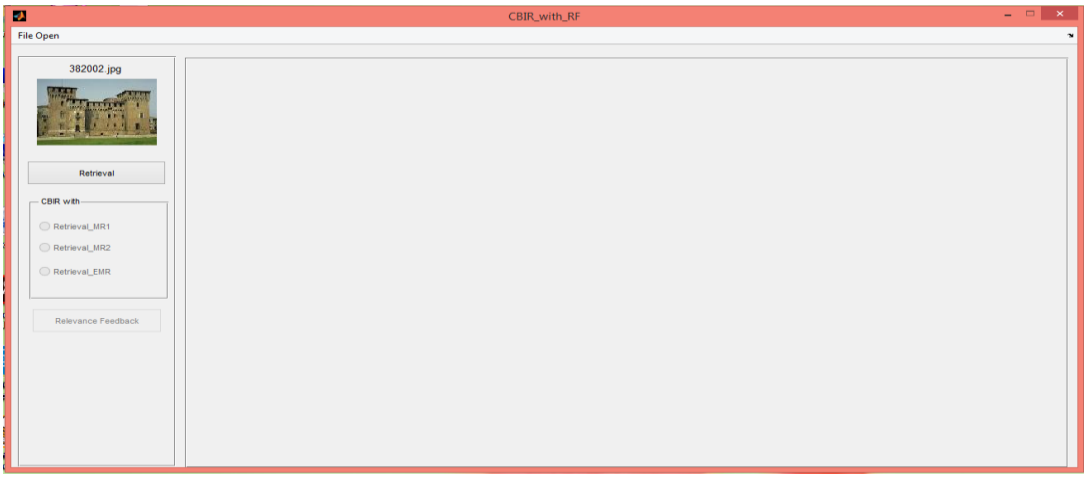
3.2.1 Mở ảnh truy vấn

- Mở ảnh bằng cách chọn File -> Open trên Menu chức năng



Hình 3-2: *Giao diện chọn ảnh truy vấn của chương trình*

- Chọn ảnh truy vấn và hiển thị ảnh truy vấn

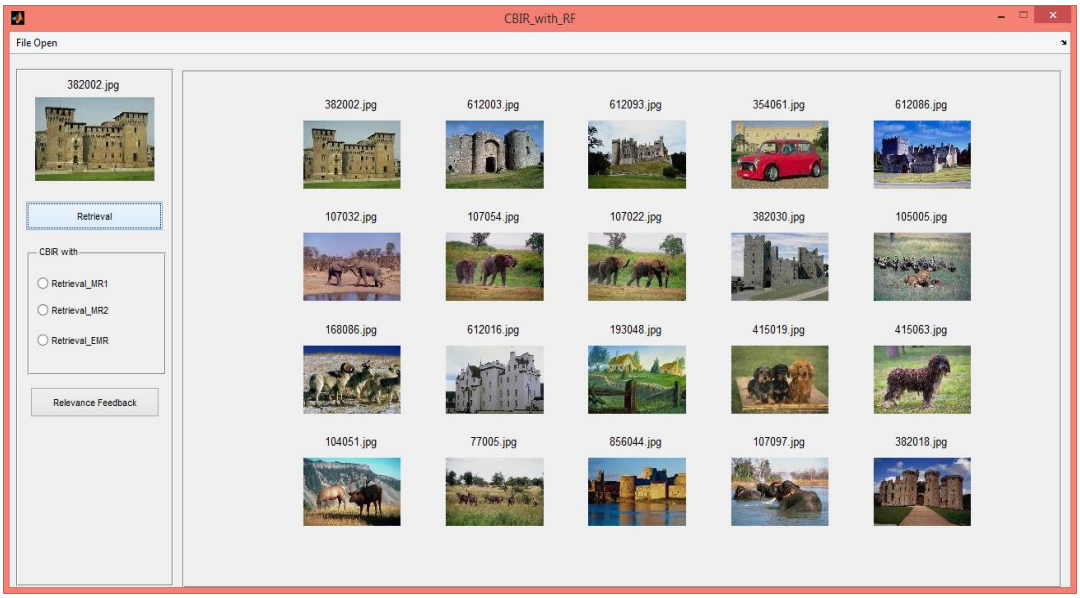


Hình 3-3: *Giao diện hiển thị ảnh truy vấn*

3.2.2 Tra cứu ảnh

Khi người sử dụng nhấn nút Retrieval. Chương trình sẽ tiến hành:

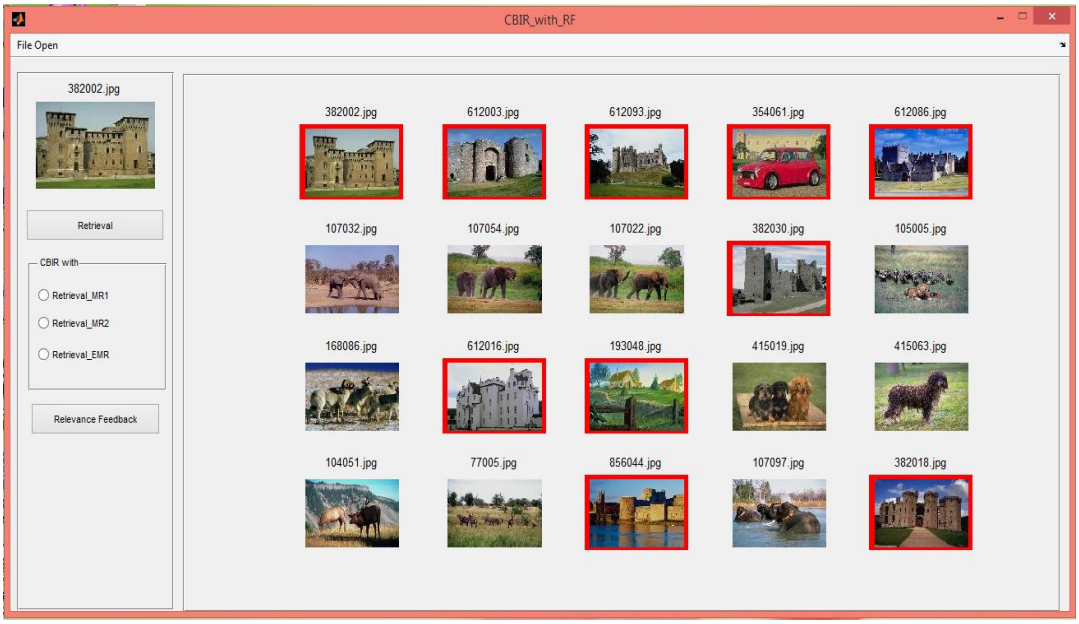
- Trích chọn đặc trưng ảnh truy vấn
- Tính toán độ tương tự giữa ảnh truy vấn với ảnh trong cơ sở dữ liệu thông qua hàm tính toán khoảng cách Euclid
- Chọn một số ảnh có độ tương tự giống nhất với ảnh truy vấn và hiển thị lên khung retrieval results để người dùng gán nhãn



Hình 3-4: *Giao diện hiển thị kết quả tra cứu ảnh ban đầu*

3.2.3 Phản hồi liên quan

Người dùng tiến hành gán nhãn có liên quan (+1) cho ảnh gần với ảnh truy vấn nhất bằng cách kích vào ảnh đó (Hình 3-5), và những ảnh còn lại không được người dùng chọn thì được gán nhãn là không liên quan (-1).



Hình 3-5: *Giao diện người dùng chọn các ảnh liên quan*

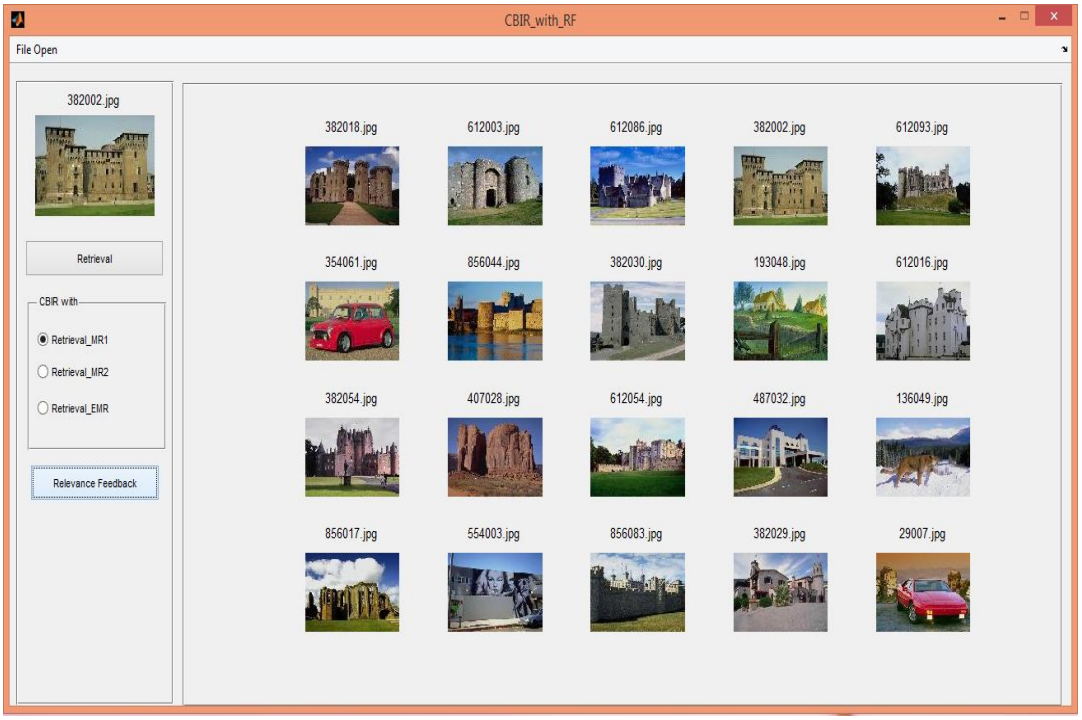
Chọn một trong các cách thức phản hồi liên quan ở khung CBIR_with, sau đó nhấn nút Relevance Feedback.

Chương trình sẽ tiến hành thực hiện các thao tác sau:

- Nếu chọn Retrieval_MR1: người dùng gửi phản hồi tích cực (phản hồi +) cho các ảnh tra cứu trả lại, hoặc khi chỉ có những ảnh có liên quan đến truy vấn, mẫu thông tin tích cực mới trở lại được thêm vào các thiết lập truy vấn và lan truyền điểm xếp hạng, đồng thời tinh chỉnh các kết quả thu hồi để đưa ra hiển thị.
- Nếu chọn Retrieval_MR2: thông tin phản hồi của người dùng có thể chứa cả kết quả mẫu dương (phản hồi +) và mẫu âm (phản hồi -) đối với các ảnh tra cứu đưa ra, hệ thống sử dụng cả hai mẫu thông tin để lan truyền các nhãn và tiến hành tính toán xếp hạng. Sau đó tinh chỉnh kết quả để hiển thị.

- Nếu chọn Retrieval_EMR: Chương trình tiến hành lấy các ảnh đã được trả về ban đầu và sử dụng chúng như tọa độ của các điểm dữ liệu trong đồ thị. Sau đó chọn điểm đại diện (điểm Anchor) và xây dựng ma trận trọng số Z với một kích thước nhỏ, cập nhật các trọng số ma trận Z và trực tiếp tính toán xếp hạng. Hình ảnh với điểm xếp hạng cao nhất được coi là phù hợp nhất và trả lại cho người dùng.

Quá trình phản hồi được lặp đi lặp lại cho đến khi thỏa mãn yêu cầu đối với người dùng.



Hình 3-6: *Giao diện hiển thị kết quả sau phản hồi*

3.3 Đánh giá hiệu năng

3.3.1 Đánh giá qua độ chính xác với các ảnh trả về khác nhau

Mỗi một lớp trong tập cơ sở dữ liệu, chọn ngẫu nhiên [20,40,60,80,100] ảnh truy vấn. Mỗi ảnh truy vấn được thực hiện thông qua 6 vòng phản hồi để đánh giá.

Bảng 1: *Độ chính xác của số ảnh lấy ngẫu nhiên trong cơ sở dữ liệu sau các lần phản hồi*

MR1

Số ảnh	Lần 1	Lần 2	Lần 3	Lần 4	Lần 5	Lần 6	TB
20	58,40%	64,02%	68,21%	71,73%	74,36%	77,18%	68,98%
40	46,89%	51,40%	54,63%	57,55%	59,88%	62,13%	55,41%
60	40,45%	44,00%	46,67%	49,19%	51,38%	53,33%	47,50%
80	35,89%	38,90%	40,76%	42,92%	45,01%	46,75%	41,70%
100	32,33%	34,85%	36,31%	37,99%	39,63%	40,97%	37,01%
TB	42,79%	46,64%	49,32%	51,87%	54,05%	56,07%	

MR2

Số ảnh	Lần 1	Lần 2	Lần 3	Lần 4	Lần 5	Lần 6	TB
20	59,50%	64,69%	67,24%	68,82%	70,79%	71,89%	67,16%
40	48,31%	53,27%	56,44%	59,31%	62,50%	64,19%	57,34%
60	41,71%	45,87%	48,82%	51,82%	55,11%	57,16%	50,08%
80	37,10%	40,51%	42,87%	45,82%	48,82%	51,12%	44,37%
100	33,45%	36,30%	38,22%	40,66%	43,05%	44,81%	39,42%
TB	44,01%	48,13%	50,72%	53,29%	56,05%	57,83%	

EMR

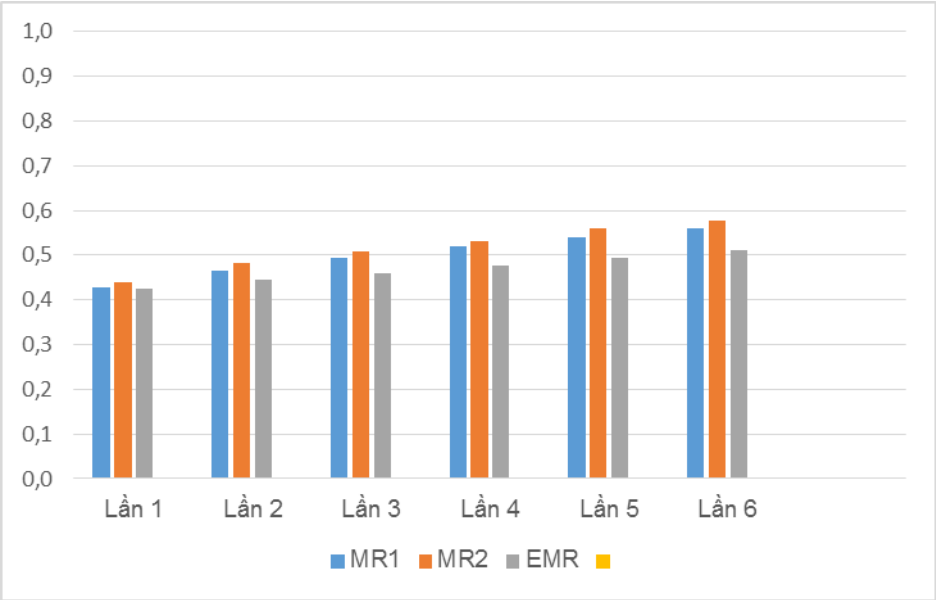
Số ảnh	Lần 1	Lần 2	Lần 3	Lần 4	Lần 5	Lần 6	TB
20	57,07%	60,25%	62,01%	63,84%	65,18%	66,55%	62,48%
40	46,25%	48,90%	50,95%	53,05%	54,99%	56,61%	51,79%
60	40,32%	42,18%	43,78%	46,12%	48,04%	49,75%	45,03%
80	36,13%	37,29%	38,48%	40,49%	42,43%	44,19%	39,84%
100	32,66%	33,48%	34,16%	35,57%	37,04%	38,60%	35,25%
TB	42,49%	44,42%	45,88%	47,81%	49,54%	51,14%	

Bảng 2: *So sánh độ chính xác trung bình của số lượng ảnh lấy ngẫu nhiên [20,40,60,80,100] ảnh trong CSDL sau các lần phản hồi*

Phương pháp	***Độ chính xác trung bình của số lượng ảnh lấy ngẫu nhiên [20,40,60,80,100] ảnh trong CSDL sau các lần phản hồi***						***TB***
	1	2	3	4	5	6	
MR1	42,79%	46,64%	49,32%	51,87%	54,05%	56,07%	50,12%
MR2	44,01%	48,13%	50,72%	53,29%	56,05%	57,83%	51,67%
EMR	42,49%	44,42%	45,88%	47,81%	49,54%	51,14%	46,88%

Ở đây, ta thấy sau mỗi vòng phản hồi thì độ chính xác của vòng phản hồi sau cao hơn vòng phản hồi trước. Trong 3 phương pháp đưa ra để so sánh,

phương pháp MR2 (phản hồi với cả mẫu dương và mẫu âm) có độ chính xác cao hơn với hai phương pháp còn lại.



Hình 3-7: *Biểu đồ so sánh độ chính xác của số ảnh lấy ngẫu nhiên sau 6 vòng phản hồi*

3.3.2 Đánh giá qua khảo sát trên tập dữ liệu khác

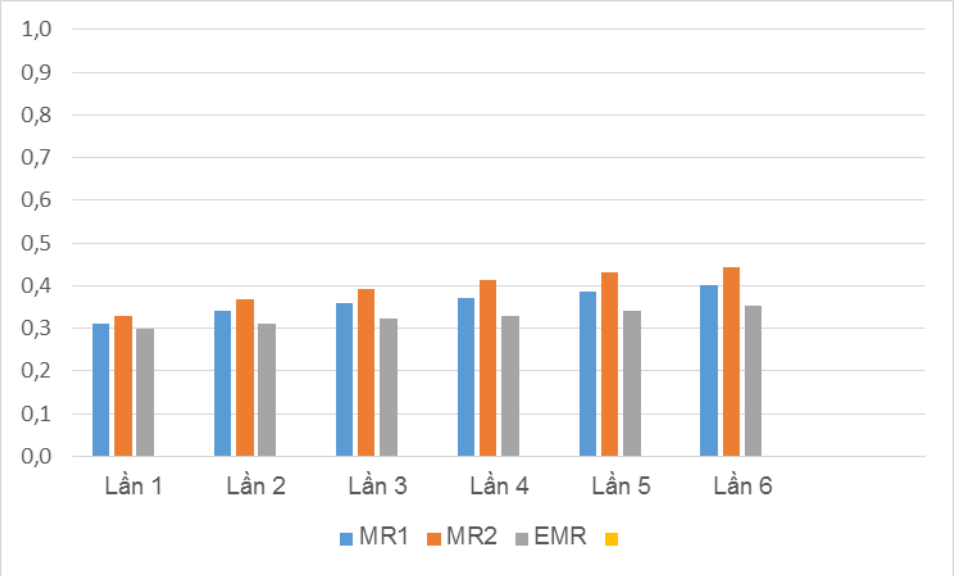
Để có sự đánh giá khách quan hơn, ta tiến hành khảo sát ba phương pháp trên qua tập cơ sở dữ liệu Caltech với 9144 ảnh được tổ chức thành 102 lớp.

Bảng 3: *Kết quả khảo sát trên tập cơ sở dữ liệu Caltech*

Phương pháp	***Độ chính xác trung bình của số lượng ảnh lấy ngẫu nhiên [20,40,60,80,100] ảnh trong CSDL sau các lần phản hồi***						***TB***
	1	2	3	4	5	6	
MR1	31,15%	33,93%	35,81%	37,20%	38,73%	40,16%	36,16%
MR2	32,83%	36,88%	39,29%	41,22%	43,07%	44,29%	39,60%
EMR	29,99%	31,15%	32,17%	32,93%	34,05%	35,14%	32,57%

Cũng giống như tập cơ sở dữ liệu Corel. Đối với tập cơ sở dữ liệu Caltech, ta thấy sau mỗi vòng phản hồi thì độ chính xác của vòng phản hồi sau cao hơn vòng phản hồi trước. Trong 3 phương pháp đưa ra để so sánh, phương pháp MR2 có độ chính xác cao hơn với hai phương pháp còn lại. Tuy

nhiên, tập cơ sở dữ liệu Caltech có dung lượng ít hơn tập cơ sờ dữ liệu Corel vì vậy độ chính xác của các ảnh tra về sau phản hồi cũng thấp hơn.



Hình 3-8: *Biểu đồ so sánh trên tập cơ sở dữ liệu Caltech*

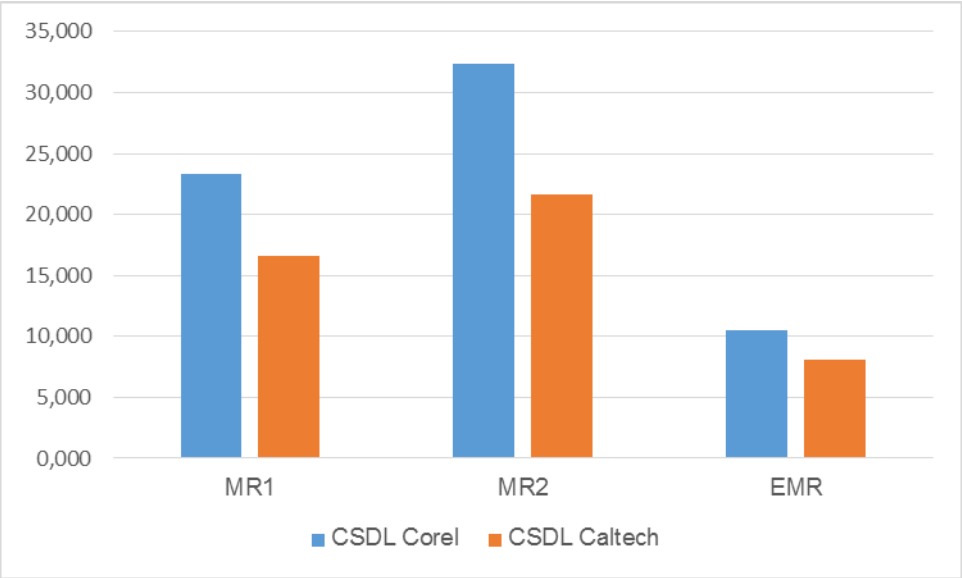
3.3.3 Đánh giá về thời gian thực hiện

Ta tiếp tục so sánh về thời gian thực hiện khi phản hồi qua tính trung bình của 6 vòng lặp cho cả ba phương pháp trên cả hai tập cơ sở dữ liệu Corel và Caltech.

Bảng 4: *So sánh về thời gian thực hiện của 3 phương pháp trên 2 tập CSDL*

Phương pháp	***CSDL Corel***	***CSDL Caltech***
MR1	23,301	16,627
MR2	32,395	21,594
EMR	10,501	8,108

Khi xét thời gian thực hiện thông qua 6 vòng phản hồi để đánh giá (tính trung bình của [20,40,60,80,100] ảnh ngẫu nhiên truy vấn) cho cả tập cơ sở dữ liệu Corel và tập cơ sở dữ liệu Caltech. Đối với phương pháp EMR, thời gian thực hiện được xử lý rất nhanh, tốc độ gấp 3 lần so với hai phương pháp còn lại.



Hình 3-9: *Biểu đồ so sánh thời gian thực hiện*

Kết luận chương 3

Hiện nay, hàng triệu các ảnh được lưu trữ trong các cơ sở dữ liệu khổng lồ và trên Internet, để tìm các ảnh quan tâm trong các tập này là vấn đề cực kỳ khó khăn. Các phương pháp CBIR hiện nay cho phép tra cứu ảnh thông qua tạo các biểu diễn của nội dung ảnh và nhóm các biểu diễn này dựa trên độ tương tự của chúng. Các phương pháp đó rất khó để có thể trả về các kết quả thỏa mãn với người dùng, bởi vì độ phức tạp và các biến đổi trong các ảnh trực quan làm cho việc tra cứu các ảnh thỏa mãn yêu cầu người dùng như mò kim đáy biển. Để giải quyết vấn đề này trong quá trình tra cứu ảnh, người dùng cần tương tác với hệ thống và đánh giá sự liên quan của các ảnh được tra cứu.

Từ những thực tế trên đặt ra một bài toán tra cứu ảnh như sau: với một ảnh truy vấn đầu vào ta thu được một tập ảnh đầu ra, người dùng đánh giá và gán nhãn cho ảnh liên quan là tích cực (mẫu dương) hay tiêu cực (mẫu âm) sau đó hệ thống sẽ tính toán và cập nhật lại trọng số của ảnh và đưa ra kết quả sau phản hồi.

KẾT LUẬN

Tra cứu ảnh dựa trên nội dung ứng dụng vào vào rất nhiều công việc hữu ích như: tìm các ảnh phong cảnh trên Internet, , điều tra hình sự dựa vào vân tay và dấu chân, chuẩn đoán bệnh trong y tế, sử dụng trong các hệ thống thông tin địa lý và viễn thám . . . . Do đó Tra cứu ảnh dựa trên nội dung trở nên rất phổ biến do nhu cầu tra cứu ảnh trong các cơ sở dữ liệu lớn tăng nhanh. Trong tra cứu ảnh thì tốc độ và độ chính xác là quan trọng, vì vậy việc tiếp tục phát triển các hệ thống tra cứu ảnh đảm bảo độ chính xác và có tốc độ nhanh là cần thiết

Nội dung Luận văn đã trình bày phương pháp tra cứu ảnh với phản hồi liên quan sử dụng mô hình Học bán giám sát trên đồ thị. Một số phương pháp học bán giám sát dựa trên đồ thị theo thuật toán lan truyền nhãn. Trước hết, tiến hành xây dựng một đồ thị có trọng số với các đỉnh là các ảnh trong cơ sở dữ liệu. Sau mỗi vòng lặp phản hồi, các ảnh được người dùng gán nhãn sẽ được xem như các đỉnh trên đồ thị, tất cả các ảnh trong cơ sở dữ liệu lan truyền xếp hạng của chúng đến các ảnh dữ liệu bên cạnh thông qua đồ thị có trọng số. Quá trình lan truyền của các điểm số xếp hạng lặp đi lặp lại cho đến khi hội tụ tới một tình trạng ổn định toàn cục để xếp hạng các ảnh liên quan đến ảnh truy vấn.

Để xử lý các cơ sở dữ liệu lớn thì cần thiết phải xây dựng đồ thị có chi phí tính toán là tuyến tính nhỏ với kích thước đồ thị. Để đạt được yêu cầu này cần xây dựng đồ thị Anchor và đề xuất một thiết kế mới cho ma trận kề W nhằm mục đính giảm thời gian đáng kể cho việc tính toán.

Về mặt thực nghiệm, chương trình tra cứu ảnh được cài đặt bằng ngôn ngữ Matlab. Kết quả thực nghiệm trên tập dữ liệu ảnh Corel và tập dữ liệu Caltech cho thấy tốc độ tra cứu ảnh đã được cải thiện đáng kể so với một số

phương pháp khác, tuy nhiên vẫn còn một vài vấn đề hạn chế, như kết quả tra cứu chưa chính xác cao

Mặc dù đã hoàn thành được mục tiêu chính của luận văn nhưng do điều kiện về thời gian có hạn mà lĩnh vực cần tìm hiểu cũng tương đối rộng nên những gì tìm hiểu được trong luận văn sẽ khó tránh khỏi những thiếu xót. Chương trình thử nghiệm cũng chưa thực sự hoàn thiện nhưng cũng đã đưa ra kết quả khả quan. Trong thời gian tới nếu có điều kiện em sẽ xây dựng chương trình tra cứu hình ảnh theo hướng học bán giám giát trên đồ thị một cách hoàn chỉnh hơn, tối ưu thuật toán nhằm tăng tốc độ tra cứu và độ chính xác của kết quả hiển thị.

TÀI LIỆU THAM KHẢO

Tiếng Anh

[1.] B. Thomee and M. Lew, "Interactive search inimage retrieval: a survey," *International Journal of Multimedia Information Retrieval*, vol. 1, no. 2, pp.71–86, 2012.

[2.] Bin Xu, Jiajun Bu, Chun Chen, Can Wang, Deng Cai, Xiaofei He, "EMR: A Scalable Graph-based Ranking Model for Content-based Image Retrieval", *IEEE Transactions on Knowledge & Data Engineering*, vol. 27, no. , pp. 102-114, Jan. 2015

[3.] Chang Ran, "Effective Graph-Based Content-Based Image Retrieval Systems for Large-Scale and Small-Scale Image Databases" (2013). *All Graduate Theses and Dissertations*. Paper 2123.

[4.] J. He, M. Li, H.-J. Zhang, H. Tong, and C. Zhang, "Manifold-ranking based image retrieval," in *Proceedings of the 12th Annual ACM International Conference on Multimedia*, 2004, pp. 9–16.

[5.] K. Tieu and P. Viola, "Boosting image retrieval," in *Proceedings of the Conference on Computer Visionand Pattern Recognition*, vol. 1, 2000, pp. 228–235 vol.1.

[6.] L. Zhang, L. Wang, and W. Lin, "Semisupervisedbiased Maximum Margin Analysis for Interactive Image Retrieval" *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2294–2308, April 2012.

[7.] M. O. Y. Rui, T. S. Huang and S. Mehrotra, "Relevance feedback: A powerful tool for interactivecontent-based image retrieval," *IEEE Transactionson Circuits and Systems for Video Technology*, vol. 8, pp. 644– 655, 1998.

[8.] Ritendra Datta, Dhiraj Joshi, Jia Li and James Z. Wang, ``Image Retrieval: Ideas, Influences, and Trends of the New Age," *ACM Computing Surveys*, vol. 40, no. 2, article 5, pp. 1-60, 2008.

[9.] S. C. H. Hoi, R. Jin, J. Zhu, and M. R. Lyu,"Semisupervised svm batch mode active learningwith applications to image retrieval," *Journal ACM Transactions on Information Systems*, vol. 27, no. 3,pp. 16:1–16:29, May 2009.

[10.] S. Nematipour, J. Shanbehzadeh, and R. A.Moghadam, "Relevance feedback optimization incontent based image retrieval via enhanced radialbasis function network," in *Proceedings of the International Multiconference of Engineers and Computer Scientists*, vol. 1, 2011.

[11.] S. Rota Bu, M. Rabbi, and M. Pelillo,"Contentbased image retrieval with relevance feedback usingrandom walks," *Pattern Recognition*, vol. 44, no. 9,pp. 2109–2122, Sep. 2011.

[12.] Xiaojin Zhu, *Semi-Supervised Learning with Graphs*, May 2005

[13.] Yi Liu, *Graph-based Learning Models for Information Retrieval: A Survey*, August 29, 2006