TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

Đào Đức Anh - Nguyễn Thành Nhân

Xây dựng hệ thống gợi ý sản phẩm dựa trên mô hình Auto-Encoder

KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN
CHƯƠNG TRÌNH CHÍNH QUY

Tp. Hồ Chí Minh, tháng MM/YYYY

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

Đào Đức Anh - 1712270
Nguyễn Thành Nhân - 1712631

Xây dựng hệ thống gợi ý sản phẩm dựa trên mô hình Auto-Encoder

KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN
CHƯƠNG TRÌNH CHÍNH QUY

GIÁO VIÊN HƯỚNG DẪN
ThS. Trần Trung Kiên

Tp. Hồ Chí Minh, tháng MM/YYYY

Lời cảm ơn

Tôi xin chân thành cảm ơn ...

Mục lục

Danh sách hình

Danh sách bảng

TÓM TẮT

Chương 1

Giới thiệu

Hiện nay, với việc bùng nổ dữ liệu trên mạng Internet, người dùng có cơ hội tiếp cận nhiều hơn với đa dạng các sản phẩm trên nền tảng số. Song song đó, các nhà cung cấp dịch vụ cũng có cơ hội tiếp cận với người dùng nhiều hơn. Tuy nhiên, người dùng cũng đang gặp nhiều khó khăn khi tìm kiếm những nội dung phù hợp với nhu cầu của mình khi hiện nay có quá nhiều sự lựa chọn được đưa ra. Với mục đích nhằm giải quyết vấn đề trên, hệ thống gợi ý sản phẩm được xây dựng để có thể dự đoán cho người dùng những nội dung - hay còn được gọi là sản phẩm phù hợp với họ. Hơn nữa, nó còn đóng vai trò quan trọng trong sự phát triển của các nhà cung cấp dịch vụ - doanh nghiệp, khi góp phần giúp nâng cao trải nghiệm người dùng cũng như tăng sự thu hút khách hàng. Theo số liệu tổng hợp được, 38% lượt click từ người dùng Google đến từ hệ thống gợi ý; và Amazon - một nền tảng mua bán trực tuyến mà 35% sản phẩm được bán thông qua hệ thống gợi ý sản phẩm.

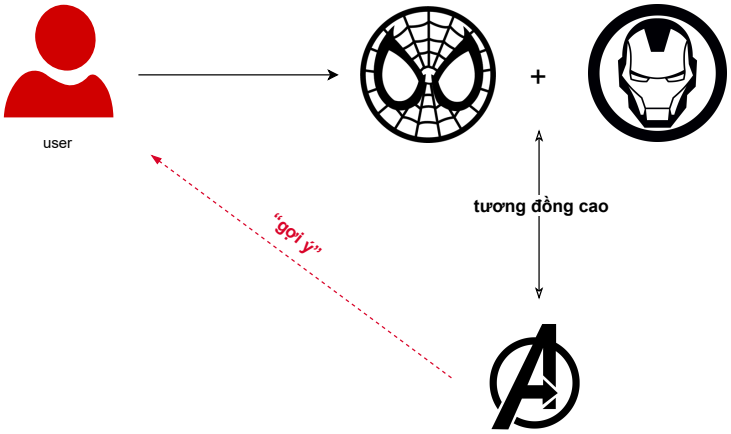
Trong lĩnh vực khoa học máy tính, hệ thống gợi ý sản phẩm là một chủ đề đang được quan tâm và nghiên cứu từ cộng đồng nghiên cứu khoa học. Bài toán xây dựng hệ thống gợi ý được phát biểu như sau:

- Đầu vào là lịch sử tương tác của người dùng (user) với các sản phẩm (items) (các sản phẩm ở đây có thể là: quảng cáo, bộ phim, bài hát, văn bản để đọc, ... tùy thuộc vào lĩnh vực cụ thể).

- Yêu cầu máy tính tự động đưa ra các sản phẩm (không có trong lịch sử tương tác) được dự đoán là phù hợp với người dùng.

Tuy vậy, việc xây dựng một hệ thống gợi ý sản phẩm một cách hiệu quả là không đơn giản. Đầu tiên, không có một "lời giải" chung cho tất cả trường hợp, mặc dù đa số các lĩnh vực hiện nay đều có thể áp dụng các hệ thống gợi ý, tuy nhiên không phải là tất cả, ta cần xét đến nhiều yếu tố khác nhau, từ đó mới có thể lựa chọn được "cách" để xây dựng hệ thống gợi ý phù hợp. Từ thực tế cho thấy, các lĩnh vực mà sản phẩm "tiêu thụ" và "sản xuất" nhanh như: phim, hình ảnh, âm nhạc, ... thì hệ thống gợi ý sẽ ít nhiều đóng vai trò quan trọng. Theo số liệu tổng hợp được, một nền tảng cung cấp video nổi tiếng hiện nay - Netflix, 75% số bộ phim được thuê đến từ hệ thống gợi ý, chứng tỏ sự ảnh hưởng lớn của hệ thống gợi ý đối với lĩnh vực này. Mặt khác, hệ thống gợi ý tác động không nhiều đến các lĩnh vực cung cấp dịch vụ hay sản phẩm giá trị cao như: thuê nhà, phương tiện giao thông, thiết bị điện tử, ... người dùng cần đánh giá thông qua nhiều yếu tố mới có thể quyết định được. Thứ hai, tùy thuộc vào nhu cầu của người sử dụng mới có thể lựa chọn "cách" mà hệ thống gợi ý hoạt động. Việc gợi ý các sản phẩm phù hợp với người dùng dựa vào nhóm người dùng có sở thích tương tự với họ hay cách dựa trên các sản phẩm có liên quan với các sản phẩm mà họ đã "thích" trước đó là khác nhau. Bài toán gợi ý sản phẩm có thể xem như là một bài toán hồi quy ("regression") nếu kết quả trả về là điểm số được dự đoán của người dùng trên tập các sản phẩm hoặc có thể xem là một bài toán xếp hạng ("top-N ranking") nếu ta cần kết quả trả về là tập các sản phẩm phù hợp nhất với người dùng. Một điều nữa cũng có thể được xem là khó khăn thứ ba khi xây dựng hệ thống gợi ý, cả trong cộng đồng nghiên cứu khoa học cũng như thực tiễn, đó là ta cần một độ đo và một phương pháp để đánh giá một cách tổng thể và khách quan nhất, khi mà dữ liệu và các thuật toán để xây dựng hệ thống gợi ý là rất đa dạng.

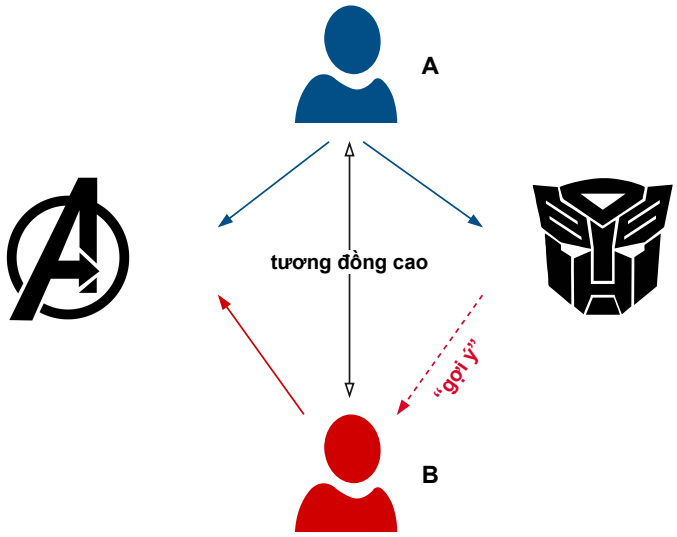
Để xây dựng hệ thống gợi ý, một hướng tiếp cận chúng ta thường nghĩ ngay đến đầu tiên là dự đoán các sản phẩm có "độ tương đồng" cao so với



Hình 1.1: Minh họa cách hoạt động của "Content-Based Filtering": mô hình gợi ý bộ phim có độ tương đồng cao với các bộ phim người dùng đã xem trước đó

các sản phẩm người dùng đã "thích" trước đó, hướng tiếp cận này được gọi là "Content-Based Filtering" (lọc dựa trên nội dung) (hình 1.1 mô tả hướng tiếp cận này). Với hướng tiếp cận này, mô hình chỉ cần các thuộc tính mô tả của sản phẩm mà không đòi hỏi dữ liệu tương tác từ người dùng khác vì các gợi ý là dành riêng cho từng cá nhân, do đó nó có khả năng nắm bắt tốt các sở thích đặc biệt của người dùng. Vì dựa trên "tính tương đồng" của sản phẩm, hệ thống có thể gợi ý cho người dùng một sản phẩm mà có thể người dùng sẽ quan tâm đến nhưng sản phẩm này cũng sẽ có thể nhận được ít sự quan tâm từ người dùng khác. Để áp dụng "Content-Based Filtering" cho từng loại dữ liệu cụ thể, ta cần "domain knowledge" cho lĩnh vực đó để thiết kế mô hình. Trong trường hợp các lĩnh vực có ít thông tin chi tiết về sản phẩm, hay các dữ liệu về sản phẩm thường không đầy đủ, rõ ràng thì hướng tiếp cận này sẽ tỏ ra không hiệu quả.

Một hướng tiếp cận khác là tìm ra "độ tương đồng" giữa các người dùng với nhau, hay tìm ra được một nhóm người dùng có cùng sở thích dựa trên dữ liệu tương tác của tất cả người dùng. Khi đó, để có thể gợi ý cho một người dùng cụ thể, hệ thống sẽ tìm ra các sản phẩm không có trong lịch sử tương tác của người dùng đó, và đã được những người dùng "tương đồng" với họ tương tác trước đó, thì hướng tiếp cận này được gọi



Hình 1.2: Minh họa cách hoạt động của "Collaborative Filtering": hai người dùng cùng xem một (hoặc nhiều) bộ phim sẽ được hệ thống đánh giá là hai người dùng "tương đồng" nhau, khi đó một bộ phim được người dùng A xem sẽ được gợi ý cho người dùng B

là "Collaborative Filtering" (lọc cộng tác) (hình 1.2 mô tả hướng tiếp cận này). Đối với hướng tiếp cận "Collaborative Filtering", mô hình dựa vào lịch sử tương tác từ người dùng khác và không dùng các thuộc tính mô tả của sản phẩm, do đó nó có khả năng tạo ra sự tình cờ cho người dùng: hệ thống có thể gợi ý một sản phẩm "tốt" cho người dùng trong trường hợp sản phẩm đó có ít điểm tương đồng so với các sản phẩm người dùng đã "thích" trước đó. Dữ liệu đầu vào của hướng tiếp cận này là các tương tác của người dùng với các sản phẩm, do đó có thể áp dụng cho nhiều lĩnh vực khác nhau mà không cần thiết phải thay đổi cấu trúc hệ thống hoặc nếu có thì cũng không cần phải thay đổi quá nhiều. Tuy nhiên, "Collaborative Filtering" gặp phải một vấn đề được gọi là "khởi động nguội" ("cold-start"), đó là khi mà một người dùng mới đến với hệ thống thì hệ thống sẽ thường khó đưa ra được gợi ý tốt cho họ, hoặc khi một sản phẩm được ít được người dùng tương tác, hệ thống thường sẽ không gợi ý sản phẩm đó cho những người dùng khác.

Ngoài ra, một hướng tiếp cận khác là "Hybrid", là sự kết hợp giữa hai hướng tiếp cận bên trên. Trong giới hạn của luận văn này, chúng tôi chỉ

tập trung tìm hiểu về hướng tiếp cận "Collaborative Filtering" vì ba lý do chính là:

- "Collaborative Filtering" tổng quan hơn so với "Content-Based Filtering" - hướng tiếp cận cần "domain knowledge" để thiết kế hệ thống cho từng lĩnh vực cụ thể.

- Với số lượng lớn và cùng với sự đa dạng của các "sản phẩm" hiện nay, tận dụng điều này, việc hệ thống gợi ý các sản phẩm đa dạng hơn dựa trên sở thích của các người dùng sẽ phù hợp hơn so với việc đưa ra các sản phẩm tương đồng với nhau.

- Ngoài ra, khi số lượng người dùng trên mạng Internet càng ngày càng tăng nhanh, thì "collaborative filtering" sẽ có được nhiều lợi thế hơn khi có thể kết hợp dữ liệu tương tác của các người dùng để đưa ra gợi ý.

Việc kết hợp các thông tin chi tiết về người dùng hay từ sản phẩm sẽ là một thông tin hữu ích cho việc xây dựng một hệ thống gợi ý sản phẩm hiệu quả hơn. Nhưng đây là một điều không đơn gảin bởi nó phụ thuộc vào "domain knowledge" ở từng lĩnh vực và chúng tôi để lại như một định hướng trong việc nghiên cứu và phát triển trong tương tai.

Phương pháp đầu tiên trong việc xây dựng một hệ thống gợi ý sản phẩm theo hướng tiếp cận "Collaborative Filtering" là thuật toán "Matrix Factorization" được giới thiệu bởi Hu [1]. Với ý tưởng là xây dựng một mô hình có khả năng "tái tạo" lại tương tác của người dùng, trong đó các tương tác được tái tạo cũng bao gồm các gợi ý cho họ. Cho đến hiện nay, "matrix factorization" vẫn là một phương pháp đơn giản nhưng vẫn mang lại kết quả cao. Tuy nhiên, thuật toán này có các nhược điểm chí mạng mà khó có thể được áp dụng để xây dựng một hệ thống gợi ý sản phẩm quy mô lớn đó là số lượng tham số của mô hình tỉ lệ tuyến tính vào cả số lượng người dùng và số lượng sản phẩm. Khi mà ngày nay, số lượng người dùng và sản phẩm tăng rất nhanh theo thời gian, ngoài ra sau khi huấn

luyện mô hình, mô hình cần thực hiện các bước tối ưu đặc biệt để có thể gợi ý cho người dùng mới. Ngoài ra, "matrix factorization" vẫn còn hạn chế đó là mô hình này là một mô hình tuyến tính, do đó nó chưa có khả năng "học" được các "đặc trưng phi" tuyến của dữ liệu. "Asymetric matrix factorization" là một phương pháp cải tiến từ "matrix factorization" với ý tưởng là trích xuất đặc trưng của người dùng thông qua các sản phẩm mà họ đã tương tác. Phương pháp này đã khắc phục được nhược điểm của "matrix factorization" khi mà sô lượng tham số của mô hình giờ chỉ phụ thuộc vào số lượng sản phẩm có trong hệ thống. Hiện nay, khi mà số lượng sản phẩm sẽ tăng chậm hơn so với số lượng người dùng trong hệ thống thì khắc phục này sẽ là một điểm mạnh của "Asymetric matrix factorization". Ngoài ra nó cũng đã giảm bớt được chi phí để đưa ra dự đoán cho người dùng mới. Tuy nhiên đây vẫn là một phương pháp tuyến tính do đó mô hình này vẫn chưa thực sự "mạnh". Ở công trình nghiên cứu [4] của tác giả Steck đã chỉ ra rằng "Asymetric Matrix Factorization" có thể được xem như là một mô hình "Auto-Encoder" tuyến tính. "Auto-Encoder" là một mô hình học đặc trưng ẩn không giám sát. Mô hình này thường được sử dụng trong những tác vụ như rút trích đặc trưng hay giảm chiều dữ liệu, ... Dựa trên ý tưởng rằng, giả định tương tác của người dùng sẽ được "phát sinh" từ một "đặc trưng ẩn" , ta có thể xem rằng "đặc trưng ẩn" này là sở thích của họ, và ta sẽ xây dựng một mô hình phát sinh được đặc trưng ẩn của người dùng từ dữ liệu tương tác của người dùng với hệ thống các sản phẩm. Sau đó đặc trưng ẩn này được sử dụng để đưa ra các gợi ý cho người dùng. Hiện nay, đã có nhiều nghiên cứu áp dụng mô hình "Auto-Encoder" trong bài toán xây dựng hệ thống gợi ý sản phẩm [3, 5, 2] để có thể tận dụng sức mạnh của các hàm phi tuyến, cụ thể là sử dụng mạng nơ-ron với các hàm kích hoạt phi tuyến (là kiến trúc cơ bản của các mô hình được dùng huấn luyện trong lĩnh vực học máy) để có được một mô hình "mạnh" hơn so với các phương pháp tuyến tính trước đó. "AutoRec" [3] được Sedhain giới thiệu, là mô hình được coi là đầu tiên trong việc sử dụng kiến trúc mô hình "Auto-Encoder" để đưa ra gợi ý cho người dùng bằng cách huấn

luyện mô hình để tái tạo lại dữ liệu tương tác của người dùng sau khi trích xuất đặc trưng ẩn từ dữ liệu tương tác của họ. "Collaborative denoising auto-encoders for top-n recommender systems" [5] (CDAE) được Wu với các cộng sự đề xuất nhằm hướng đến bài toán đưa ra gợi ý theo hướng "top-N sản phẩm" phù hợp với người dùng hay nói cách khác là dự đoán tập các sản phẩm mà người dùng "thích" nhất. Mô hình này dã được xây dựng dựa trên "AutoRec" nhưng có các chỉnh sửa để phù hợp hơn với bài toán xây dựng hệ thống gợi ý sản phẩm khi mà ta quan tâm đến việc đưa ra top các sản phẩm phù hợp với người dùng thay vì tái tạo lại tương tác của họ. Ngoài ra, CDAE còn cải tiến từ "AutoRec" đó là thêm "nhiễu" vào dữ liệu huấn luyện nhằm giúp mô hình tránh được tình trạng "Overfitting" (là trình trạng mà mô hình học "tủ" trên tập dữ liệu được huấn luyện nên mô hình đạt kết quả thấp trên các tập dữ liệu kiểm định), thường gặp phải khi huấn luyện mạng nơ-ron với hàm kích hoạt phi tuyến.

Một trong những phương pháp nổi bật nhất tới thời điểm hiện tại trong việc sử dụng kiến trúc mô hình "Auto-Encoder" để xây dựng mô hệ thống gợi ý sản phẩm là "Variational Autoencoder for Collaborative Filtering"[2] được giới thiệu bởi tác giả Liang cùng các cộng sự. Công trình này: "Variational Autoencoders for Collaborative Filtering", đã được công bố tại hội nghị "International World Wide Web Conference Committee 2018". Trong khóa luận này, chúng tôi tìm hiểu và cài đặt lại mô hình được đề xuất trong bài báo, và thực hiện một số thí nghiệm nhằm phân tích khả năng học của mô hình. Đây là một phương pháp sử dụng mô hình "Variational Auto-encoder" (VAEs) - một biến thể của mô hình "Auto-Encoder" cơ bản để có thể xây dựng một hệ thống gợi ý sản phẩm hiệu quả.

Lý do đầu tiên mà chúng tôi quyết định tìm hiểu về mô hình này cũng như là điểm khác biệt của VAEs so với các phương pháp đã được kể đến ở trên đó là đặc trưng ẩn được phát sinh là một phân phối xác suất thay vì là "một điểm dữ liệu cố định" ("Auto-Encoder" và "Denosing Auto-Encoder" sẽ nhận dữ liệu đầu vào ở chiều không gian "cao" và trích xuất đặc trưng

ẩn là một điểm dữ liệu ở chiều không gian "thấp" hơn mà vẫn thể thể hiện được "tính chất" của dữ liệu ban đầu). Việc đặc trưng ẩn là một phân phối xuất sẽ giúp cho mô hình có thể sử dụng dặc trưng ẩn này để phát sinh dữ liệu, ứng dụng với bài toán ta đang quan tâm đó chính là sử dụng sở thích nắm bắt được từ các người dùng để đưa ra gợi ý cho họ. Nền tảng của VAEs đó là dựa trên phương pháp "Variational Inference", một phương pháp trong lĩnh vực xác suất thống kê. Phương pháp này thường được dùng để suy diễn "dữ liệu ẩn" dựa trên những "dữ liệu ta quan sát được". Đặc điểm của phương pháp này là có thể áp dụng tốt cho dữ liệu thưa, có nghĩa là đối với "dữ liệu quan sát" được bị hạn chế thì việc "suy diễn" dữ liệu vẫn đạt được kết quả tốt. Ở đây, trong bài toán xây dựng hệ thống gợi ý sản phẩm thì, dữ liệu tương tác của người dùng sẽ được xem là dữ liệu quan sát được và đặc trưng ẩn là "dữ liệu ẩn" mà ta không quan sát được. Đặc biệt ở đây, đối với bài toán xây dựng hệ thống gợi ý sản phẩm thì thông thường mỗi người dùng chỉ tương tác với một lượng tỉ lệ nhỏ các sản phẩm so với toàn bộ sản phẩm có trong hệ thống, do đó việc áp dụng phương pháp "Variational Inference" hay nói cách khác là VAEs sẽ phù hợp với bài toán xây dựng hệ thống gợi ý sản phẩm dựa trên ý tưởng phát sinh tương tác của người dựa vào đặc trưng ẩn của người dùng.

Bên cạnh đó, để có thể xây dựng một mô hình giải quyết giải quyết vấn đề mà hệ thống gợi ý sản phẩm quan tâm đó là có thể đưa ra được tập các sản phẩm có khả năng cao sẽ phù hợp với người dùng, mục đích này khác hoàn toàn với việc tái tạo lại dữ liệu hay nói cách khác là tái tạo lại tương tác của người dùng. Với mục tiêu để mô hình thể hiện được rõ khả năng mà một sản phẩm có phù hợp với người dùng hay không thì tác giả Liang đã sử dụng hàm "Multinomial log-likelihood" (một phương pháp đánh giá bộ tham số của mô hình với giả định dữ liệu tương tác của người dùng tuân theo một phân phối đa thức). "Multinomial log-likelihood" sẽ khiến kết quả trả về từ mô hình thể hiện "giá trị độ lớn về xác suất". Với "multinomial log-likelihood" các sản phẩm sẽ phải "cạnh tranh" với nhau để có thể đạt được xác suất được chọn cao hơn (vì kết quả trả về sẽ phải

ràng buộc là tổng kết quả "xác suất" trả về sẽ bằng 1 hay xác suất được chọn của các sản phẩm trả về từ mô hình sẽ có tổng bằng 1). Tuy không được sử dụng nhiều trong lĩnh vực xây dựng hệ thống gợi ý sản phẩm, thay vào đó, "multinomial log-likelihood" lại thường được sử dụng nhiều trong các lĩnh vực về mô hình ngôn ngữ (Language models) hay về các bài toán trong lĩnh vực kinh tế (Ecomomics) nhưng chúng tôi tin rằng "multinomial log-likelihood" sẽ là một lựa chọn phù hợp trong lĩnh vực xây dựng hệ thống gợi ý sản phẩm.

Cuối cùng, một điểm đặc biệt của phương pháp này đó là tác giả đã đề xuất thêm một siêu tham số để kiểm soát việc đánh đổi giữa việc tối thiểu độ lỗi trong việc mô hình hóa dữ liệu và việc đảm bảo đặc trưng ẩn được phát sinh tuân theo phân phối xác suất được giả định từ mô hình. Siêu tham số là một tham số được chỉ định bởi lập trình viên cho mô hình thay vì mô hình học phải học như các tham số thông thường. Nguyên nhân dẫn đến việc cần phải có một siêu tham số để kiểm soát được việc đánh đổi nói trên là vì với việc xây dựng một hệ thống gợi ý sản phẩm thì chúng ta sẽ quan tâm tới việc mô hình hoá dữ liệu nhiều hơn là việc đảm bảo đặc trưng ẩn sẽ tuân theo phân phối xác suất. Nói một cách khác, thì việc mô hình hoá dữ liệu sẽ đảm bảo rằng đặc trưng ẩn có được từ mô hình sẽ thể hiện được cho dữ liệu ban đầu, hay trong bài toán xây dựng hệ thống gợi ý thì điều này đảm bảo mô hình sẽ nắm bắt được xu hướng sở thích giữa các người dùng nhưng nếu mô hình chỉ tập trung để mô hình hoá dữ liệu thì mô hình sẽ bị giới hạn lại thành việc tái tạo lại tương tác của người dùng. Còn về việc đảm bảo tính chất về phân phối xác suất sẽ giúp cho đặc trưng có thể phát sinh được dữ liệu một cách hợp lý, nếu quá tập trung để đảm bảo điều này thì mô hình sẽ khó nắm bắt được sở thích chung giữa các người dùng, ngược lại, nếu bỏ qua tính chất này, thì đặc trưng ẩn sẽ khó có thể đưa ra được những sản phẩm không nằm trong lịch sử tương tác mà phù hợp với người dùng. Do vậy, khi xây dựng hệ thống gợi ý sản phẩm thì ta sẽ cần phải kiểm soát được việc đánh đổi này thì hệ thống gợi ý mới có thể hoạt động hiệu quả hơn.

Mặc dù không phải là mô hình đạt được kết quả tốt nhất hiện nay trong việc xây dựng hệ thống gợi ý sản phẩm, tuy nhiên kiến thức nền tảng để xây dựng mô hình này bao phủ về lĩnh vực học máy cũng như là kiến thức về mô hình xác suất. Với những lợi điểm như trên, thì chúng tôi đã quyết định sẽ tập trung tìm hiểu model được đề xuất ở bài báo [2].

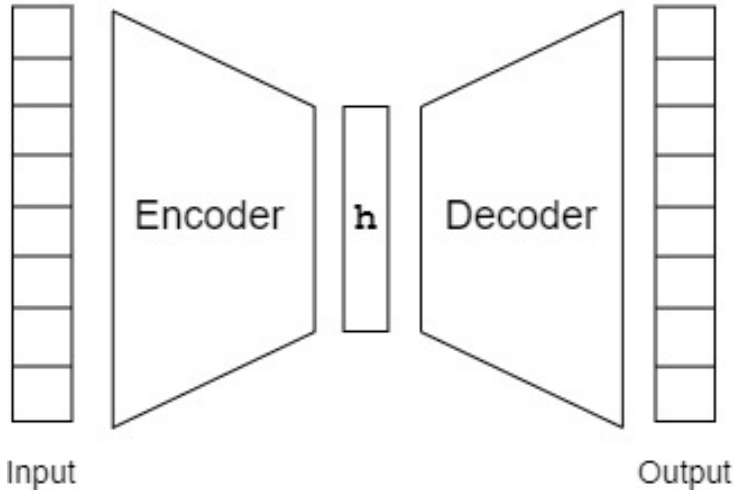
Phần còn lại của khóa luận được trình bày như sau:

- Chương 2 trình bày sơ lược về mô hình "Auto-Encoder" và các kiến thức nền tảng của mô hình "Variational Auto-Encoder".
- Chương 3 trình bày về cách áp dụng mô hình "Variational Auto-Encoder" cùng với hàm lỗi multinomial log-likelihood cho bài toán xây dựng hệ thống gợi ý. Bên cạnh đó, chương này cũng phân tích các hạn chế của mô hình đồng thời đề xuất phương pháp giúp giải quyết các hạn chế đó. Chương này là phần chính của khóa luận.
- Chương 4 trình bày về các thí nghiệm và các kết quả đạt được.
- Cuối cùng, kết luận và hướng phát triển được trình bày ở chương 5.

Chương 2

Kiến thức nền tảng

Tong chương này, đầu tiên chúng tôi sẽ trình bày về mô hình "Auto-Encoders", một mạng nơ-ron được dùng để học đặc trưng ẩn dựa trên phương pháp học không giám sát. Sau đó, chúng tôi giới thiệu và trình bày về nền tảng xác suất của "Variational Auto-encoders" (VAEs) và lợi ích mang lại của mô hình này so với "Auto-Encoder" trong tác vụ học đặc trưng ẩn; Những điểm lợi này chính là lý do mà chúng tôi tập trung nghiên cứu VAEs. Bên cạnh đó, chúng tôi sẽ trình bày về "Maximum Likelihood Estimation", một phương pháp dùng để đánh giá các tham số của mô hình, đại diện cho các tham số của các phân phối xác suất dựa trên dữ liệu huấn luyện. Chương này đặc biệt là phần về "Variational Auto-Encoders" cung cấp những kiến thức nền tảng để có thể hiểu rõ về những đề xuất của chúng tôi ở chương kế tiếp.



Hình 2.1: Minh họa "Auto-Encoder"

2.1 Mô hình rút trích đặc trưng "Auto-Encoder"

Mô hình "Auto-Encoder" là một mạng nơ-ron truyền thẳng được huấn luyện để cố gắng sao chép đầu vào của nó thành đầu ra. Bên trong "Auto-Encoder" có một lớp ẩn $\boldsymbol{h}$ mô tả đặc trưng ẩn, gọi là véc-tơ biểu diễn ẩn đại diện cho đầu vào của nó.

Kiến trúc của một "Auto-Encoder" (được minh họa trong hình 2.1) bao gồm hai phần:

- Bộ mã hóa (encoder) ánh xạ véc-tơ đầu vào sang véc-tơ biểu diễn ẩn:

$$\boldsymbol{h} = f(x)$$

- Bộ giải mã (decoder) có nhiệm vụ cố gắng tái tạo lại véc-tơ đầu vào từ véc-tơ biểu diễn ẩn:

$$\hat{x} = g(\boldsymbol{h}) = g(f(x))$$

"Auto-Encoder" được huấn luyện bằng cách cực tiểu hóa hàm lỗi là độ sai lệch giữa dữ liệu được tái tạo với dữ liệu đầu vào.

$$L(x, g(f(x))) \tag{2.1}$$

Các hàm để tính độ lỗi thường được dùng là "Mean-square error" hoặc "Binary cross-entropy". Tương tự như các mạng nơ-ron khác, "Auto-Encoder" có thể được huấn luyện bằng phương pháp "Gradient-descent" với thuật toán lan truyền ngược ("back-propagation").

Khi thiết kế mô hình, kiến trúc của encoder, decoder và kích thước của véc-tơ $\boldsymbol{h}$ được xem như những siêu tham số của mô hình. Bằng các cách thiết lập khác nhau, mô hình sẽ có những tính chất khác nhau. "Auto-Encoder" với encoder và decoder là những hàm phi tuyến (cụ thể là mạng nơ-ron với hàm kích hoạt phi tuyến) với khả năng tính toán quá mạnh hay trường hợp kích thước của véc-tơ $\boldsymbol{h}$ lớn hơn hoặc bằng so với véc-tơ đầu vào sẽ dẫn đến mô hình chỉ học cách sao chép thay vì trích xuất các đặc trưng ẩn từ dữ liệu.

Thông thường, một "Auto-Encoder" sao chép một cách "hoàn hảo" đầu vào thành đầu ra sẽ không có nhiều ý nghĩa. Thay vào đó, "Auto-Encoder" được thiết kế với các ràng buộc để không thể học cách sao chép "hoàn hảo" mà chỉ có thể sao chép gần đúng, từ đó ta hy vọng quá trình huấn luyện "Auto-Encoder" sẽ thu được véc-tơ biểu diễn ẩn có những thông tin hữu ích.

Từ véc-tơ biểu diễn ẩn thu được trong quá trình huấn luyện "Auto-Encoder", ta có thể áp dụng mô hình này như một mô hình trích xuất đặc trưng ẩn từ dữ liệu, làm đầu vào cho các tác vụ khác. Hoặc véc-tơ biểu diễn ẩn này có thể áp dụng được trong các tác vụ giảm chiều dữ liệu hỗ trợ cho các tác vụ lưu trữ, truy vấn, tìm kiếm.

2.1.1 "Undercomplete Auto-Encoder"

Như đã trình bày trước đó, việc sao chép đầu vào thành đầu ra của "Auto-Encoder" không mang nhiều ý nghĩa. Ta cần các ràng buộc để có được $\boldsymbol{h}$ nhận các thuộc tính hữu ích với các ràng buộc khi thiết kế mô hình.

Một cách ràng buộc để mô hình có thể học được các đặc trưng ẩn từ

dữ liệu là giới hạn véc-tơ đặc trưng ẩn $\boldsymbol{h}$ có kích thước nhỏ hơn đáng kể so với véc-tơ đầu vào; tính chất này được gọi là "under-complete".

Mô hình "Auto-Encoder" với kích thước $\boldsymbol{h}$ nhỏ hơn đáng kể so với kích thước của véc-tơ đầu vào được gọi là "Undercomplete Auto-Encoder". Việc giới hạn này sẽ buộc mô hình phải nắm bắt các đặc trưng nổi bật nhất.

Quá trình huấn luyện "Undercomplete Auto-Encoder" cũng giống với mô hình "Auto-Encoder", ta cần cực tiểu hóa hàm lỗi (công thức 2.1) là độ sai lệch giữa dữ liệu được tái tạo với dữ liệu đầu vào.

"Undercomplete Auto-Encoder" là mô hình tốt để sử dụng cho các tác vụ tiêu biểu của "Auto-Encoder" truyền thống như trích xuất đặc trưng, giảm chiều dữ liệu bởi vì tính chất "under-complete" của mô hình giúp dễ dàng thu được véc-tơ biểu diễn ẩn mang những thông tin hữu ích.

2.1.2 "Denoising Auto-Encoder"

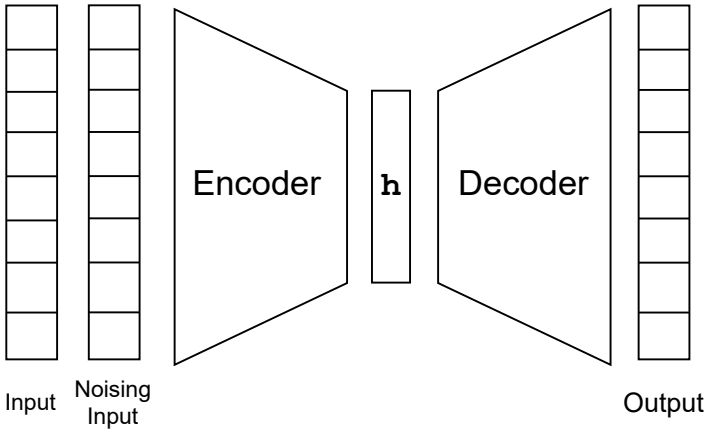
Hàm lỗi của một "Auto-Encoder" thông thường sẽ "phạt" một mức nhất định với các mẫu dữ liệu được tái tạo lại khác với dữ liệu đầu vào. Điều này vô hình chung khuyến khích nghĩa là $f \circ g$ là một hàm đồng nhất nếu khả năng tính toán của f và g cho phép. Nói đơn giản hơn, điều này là việc mô hình sao chép "hoàn hảo" đầu vào thành đầu ra của nó. Khi đó, véc-tơ biểu diễn ẩn sẽ không có các thông tin hữu ích.

Bằng cách thay đổi cách tính toán độ lỗi khi tái tạo lại, cụ thể là thêm nhiễu vào véc-tơ đầu vào, sau đó tính toán độ lỗi là đầu ra được mô hình tái tạo lại so với đầu vào ban đầu như sau:

$$L(x, g(f(\tilde{x}))) \tag{2.2}$$

với $\tilde{x}$ là véc-tơ đầu vào x được thêm một độ nhiễu, ta có được mô hình "Denoising Auto-Encoder" (hình 2.2).

"Denoising Auto-Encoder" phải học cách khử độ nhiễu đã được thêm vào véc-tơ đầu vào, giảm khả năng sao chép của mô hình.



Hình 2.2: Minh họa "Denoising Auto-Encoder"

2.2 "Variational Auto-Encoder"

Variational Auto-encoders (VAEs) là một biến thể đặc biệt của Auto-encoder cơ bản. VAEs, ngoài là một mô hình rút trích đặc trưng ẩn dựa trên phương pháp học không giám sát, còn là một mô hình phát sinh dữ liệu hiệu quả. Khả năng phát sinh thêm dữ liệu là việc dựa trên những đặc trưng ẩn đã học được, VAEs dựa vào những đặc trưng này để thực hiện tác vụ phát sinh dữ liệu. Đây là một điểm khác biệt so với mô hình Auto-Encoder khi mà đặc trưng ẩn học từ Auto-encoder cơ bản không thể được sử dụng để phát sinh. Điều tạo nên sự khác biệt này là bởi đặc trưng ẩn có được từ VAEs là một phân phối xác suất. Auto-encodêr hay kể cả denosing auto-encoder, việc nhận dữ liệu đầu vào, và trích xuất đặc trưng ẩn đều có thể được xem như là một phép chiếu dữ liệu ở chiều không gian cao lên một chiều không gian thấp hơn (thông thường thì số chiều của đặc trưng ẩn dược trích xuất sẽ nhỏ hơn so với dữ liệu đầu vào). Do đó, đặc trưng ẩn này, ta có thể xem như là một điểm dữ liệu mới thể hiện cho dữ liệu ban đầu ở một chiều không gian khác với số chiều thấp hơn. Còn VAEs, thì đặc trưng ẩn không còn là một điểm dữ liệu, thay vào đó sẽ là một "phân phối xác suất". Phân phối xác suất là quy luật cho ta biết với mỗi giá trị cụ thể của một đại lượng, một biến số nào đó, sẽ tương ứng với giá trị xác suất là bao nhiêu.

Tuy nhiên, bản chất của mô hình Variational auto-encoder được xuất phát từ lĩnh vực xác suất thống kê chứ không phải lĩnh vực khoa học máy tính. Bản chất của một mô hình VAEs là một mô hình đồ thị (graphical models) - là một mô hình dùng để giải thích các mối quan hệ giữa các biến ngẫu nhiên trong xác suất thống kê. Và nền tảng của mô hình là Variation Inference - là một phương pháp cũng thuộc lĩnh vực xác suất thống kê với mục đích có thể "giải thích" được dữ liệu mà ta không quan sát được từ những dữ liệu mà ta đã có. Tận dụng sức mạnh của mạng nơ-ron trong lĩnh vực học máy, các hàm số xác suất được thay thành các mạng nơ-ron. Và thông qua việc huấn luyện mô hình để tìm ra bộ trọng số tốt nhất để giải quyết bài toán được giả định mà mô hình cần giải quyết.

Do sự liên hệ chặt chẽ với lĩnh vực xác suất, ở mục này, chúng tôi sẽ trình bày về nền tảng xác suất liên quan với mô hình Variational Auto-Encoder, bao gồm các khái niệm, định lý trong lĩnh vực xác suất thống kê để có thể dễ dàng trình bày nội dung của VAEs ở các mục tiếp theo, cũng như là cách huấn luyện cho mô hình VAEs.

2.2.1 Nền tảng xác suất

Với sự tăng nhanh về số lượng dữ liệu có trên các nền tảng số thì nhu cầu cần một phương pháp có thể phân tích dữ liệu một cách tự động đang là một nhu cầu càng ngày càng tăng theo. Mục tiêu của học máy đó là phát triển các phương pháp mà có thể tự động phát hiện các mẫu "patterm" trong dữ liệu và sau đó sử dụng những "patterm" vừa khám phá được để có thể dự đoán dữ liệu trong tương lai hoặc để thực hiện các mục đích khác như thực hiện các quyết định/ dự đoán dựa trên "những điều chưa chắc chắn". Lý thuyết xác suất "probability theory" có thể được áp dụng cho bất kỳ vấn đề nào liên quan đến "những điều chưa chắc chắn". Trong máy học, "những điều chưa chắc chắn" đến từ nhiều dạng như: dự đoán/ quyết định nào là tốt nhất khi cho trước một vài điểm dữ liệu? Mô hình nào là tốt nhất khi cho trước các một vài điểm dữ liệu? ... Do đó học máy

có liên quan khá là gần gũi với lĩnh vực xác suất thống kê và khai thác dữ liệu, nhưng khác ở các trọng tâm và các thuật ngữ.

Trên lý thuyết thì có ít nhất hai cách diễn giải của xác suất: "diễn giải tần suất" (frequentist interetation) và "diễn giải bayesian". Ở cách diễn giải thứ nhất thì xác suất được thể hiện thông qua việc thực hiện các thí nghiệm nhiều lần. Ví dụ như nếu ta thực hiện thí nghiệm tung đồng xu thì ta kì vọng rằng việc đồng xu xuất hiện mặt ngửa khoảng một nửa lần trong quá trình thực hiện. Còn ở cách diễn giải bayesian của xác suất thì thường được sử dụng để định lượng về "những điều chưa chắc chắn". Vậy nên ở góc nhìn này sẽ liên quan đến các thông tin hơn là việc lặp lại các thí nghiệm. Một trong những ưu điểm của cách diễn giải này đó là nó có thể được sử dụng để mô hình "những điều chưa chắc chắn" của sự việc/sự kiện mà ta đang quan tâm đến mà không có tần suất xuất dài hạn. Ví dụ liên hệ với các bài toán trong lĩnh vực học máy như chúng ta nhận một email và ta quan tâm đến việc tính phân phôi xác suất mà email vừa nhận là spam; hay trong bài toán chúng ta nhận thấy được một vật thể thông qua màn hình radar và ta muốn tính phân phối xác suất theo vật thể vừa được phát hiện chính xác là gì ? một con chim, hay máy bay? Trong những trường hợp trên thì ý tưởng việc lặp lại các thí nghiệm sẽ không giúp ích cho chúng ta trong việc giải quyết các vấn đề nhưng với Bayesian thì điều này khá là tự nhiên và có thể được áp dụng để giải quyết bất kỳ vấn đề nào liên quan tới những "điều không chắc chắn".

Định lý Bayes và ứng dụng trong lĩnh vực học máy

Trong lĩnh vực "máy học" và "thống kê Bayesian", chúng ta thường quan tâm đến việc thực hiện các phép suy diễn dữ liệu ẩn ta không quan sát được khi cho trước các dữ liệu ta quan đã quan sát được. Ví dụ như một mô hình phát hiện sản phẩm lỗi, thì ta sẽ quan tâm đến việc khi ta đã có các thông tin về sản phẩm, đây là dữ liệu ta đã quan sát được và từ đó ta suy diễn hay còn được gọi là dự đoán về tình trạng của sản phẩm mà tình trạng này chưa quan sát được. Giả sử rằng, ta có x biến ngẫu nhiên thể

hiện cho dữ liệu mà ta đã có, và y là biến ngẫu nhiên của dữ liệu mà ta chưa quan sát được. Theo đó, ta sẽ quan tâm đến việc tìm ra được giá trị y cụ thể khi cho trước giá trị x. Về xác suất, hay cụ thể ở đây, theo định lý Bayes, nếu ta có $p(x)$ là thông tin mà ta đã nắm được; và một vài mẫu dữ liệu $p(y|x)$ mà ta đã có trong việc thể hiện mối quan hệ giữa y và x, cụ thể là giá trị của y khi ta đã có x, thì theo công thức Bayes ta có:

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}$$

Trong đó, $p(x)$ được gọi là "prior", "prior" thể hiện "kiến thức" chủ quan ban đầu của chúng ta trước khi ta có bất kỳ về thông tin nào về liệu ẩn mà chưa quan sát được. Ta có thể chọn "prior" nào bất kỳ sao cho phù hợp với chúng ta, nhưng một điều chúng ta cần phải đảm bảo đó là "prior" phải là có giá trị khác không trên tất cả các giá trị có thể xuất hiện của x, kể cả khi giá trị đó rất hiếm khi xảy ra. $p(y|x)$ là "likelihood", mô tả mối quan hệ giữa x và y liên quan với nhau như thế nào, và trong trường hợp dữ liệu không liên tục (miền giá trị của x và y không phải là một miền giá trị liên tục), thì nó là xác suất của việc giá trị y cụ thể khi ta đã biết về dữ liệu ẩn x. Chú ý rằng "likelihood" không phải là một phân phối xác suất của x, nó chỉ là phân phối cho y. Ta gọi $p(y|x)$ là "likelihood của x (khi cho trước y)" hoặc là "xác suất của y khi cho trước x" chứ không được gọi là "likelihood của y"[**MacKay2003**]. "posterior" $p(x|y)$ sẽ là giá trị mà ta quan tâm theo này quan điểm của Bayes. Nó thể hiện rằng chúng ta có được thông tin gì về x khi ta có dữ liệu về y. Theo công thức trên, thì ta còn một giá trị nữa đó là

$$p(y) = \int p(y|x)p(x)\mathrm{d}x = \mathrm{E}_X[p(y|x)]$$

là "marginal likelihood" hay còn được gọi là "evedience". Trong công thức trên thì kí hiệu E thể hiện giá trị kỳ vọng của phân phối xác suất. Theo công thức của bayes thì, "marginal likelihood" độc lập với dữ liệu ẩn x, do

đó "marginal likelihood" sẽ đóng vai trò đảm bảo giá trị của "posterior" sẽ được chuẩn hoá, có nghĩa là "posterior" sẽ có khoảng giá trị từ 0 đến 1. Ngoài tác dụng để chuẩn hoá ra thì "marginal likelihood" sẽ đóng vai trò trong việc lựa chọn mô hình "model selection" (là việc chọn ra mô hình tốt nhất, giữa các mô hình hoặc giữa các bộ siêu tham số). Tuy nhiên, với việc tính "marginal likelihood" là một giá trị tích phân, do đó thông thường việc tính toán giá trị chính xác cho "marginal likelihood" sẽ không dễ dàng.

Mô hình xác suất

Mô hình xác suất là một mô hình được dùng để mô tả một phân phối xác suất bằng cách sử dụng một đồ thị để mô tả các biến ngẫu nhiên tương tác với nhau trong phân phối xác suất. Ở đây chúng tôi sử dụng từ "đồ thị" là một định nghĩa về cấu trúc dữ liệu được mô tả trong lĩnh vực lý thuyết đồ thị. Đồ thị bao gồm các đỉnh được kết nối trực tiếp với nhau thông qua các cạnh. Vì cấu trúc của mô hình đươc mô tả bằng đồ thị cho nên những mô hình này còn được gọi với một tên gọi khác là "Graphical model". Mô hình xác suất thể hiện những khía cạnh mà mang tính chất "chưa chắc chắn" (uncertain) của một "thí nghiệm" dưới dạng của một phân phối xác suất. Ở đây, xét trong quá trình giải quyết một vấn đề nào đó bất kỳ, một thí nghiệm được thực hiện để kiểm chứng lý thuyết, ... thì cụm từ "chưa chắc chắn" mang ý nghĩa là chỉ những điểm dữ liệu mà chúng ta chưa quan sát hay chưa thu thập được, do đó về giá trị cụ thể, khoảng dữ liệu và tính chất của những điểm dữ liệu này chúng ta chưa có thể kiểm chứng được. Mô hình xác suất mang lại cho chúng các công cụ để có thể xây dựng một mô hình học, thuật toán học, cách đưa ra dự đoán từ mô hình đã được huấn luyện và cả việc lựa chọn mô hình. Các công cụ này nhất quán và đồng nhất do nó dựa trên nền tảng

Mục tiêu của một mô hình học máy hay mô hình học sâu là có thể giải quyết được các vấn đề mà ta không có một cách trực tiếp để giải quyết, tuy nhiên bên cạnh đó lại có dữ liệu được thu thập về vấn đề đó. Điều đó

có nghĩa là ta cần xây dựng các mô hình học có thể "hiểu" được hình ảnh, âm thanh tiếng nói hay một đoạn văn bản. Nhưng để xây dựng được các mô hình này ta sẽ phải đối mặt với nhiều khó khăn cần giải quyết. Một trong những khó khăn đó là việc dữ liệu ta thu thập được thường có các cấu trúc phức tạp và dữ liệu ở chiều không gian cao, để có thể xử lý được những dữ liệu như vậy là điều không dễ dàng.

Điểm mạnh của một mô hình xác suất đó là khả năng có thể giảm được chi phí cho việc thể hiện một mô hình xác suất và cũng như là chi phí cho việc huấn luyện cũng như suy diễn. Cơ chế hoạt động chính của mô hình xác suất cho phép tát cả các phép tính được thực hiện với thời gian thực thi và bộ nhớ hơn so với các mô hình mô hình hoá dữ liệu. Một lợi ích khác trong việc sử dụng mô hình xác suất khác đó là cho phép chúng ta tách biệt các thể hiện của "kiến thức" một cách chi tiết từ quá trình huấn luyện hay quá trình suy diễn. Điều này làm cho các mô hình dễ dàng để cài đặt và "debug". Chúng ta có thể thiết kế, phân tích và đánh quá thuật toán huấn luyện và thuật toán suy diễn mà có thể áp dụng rộng rãi ở tất cả các loại đồ thị. Một cách độc lập, chúng ta có thể thiết kế các mô hình mà có thể nắm bắt được mối quan hệ mà chúng ta tin rằng nó quan trọng trong dữ liệu mà chúng ta có. Sau đó chugn ta có thể kết hợp các thuật toán và các cấu trúc khác nhau và có các "tích Đề-Các" bất kỳ nào mà phù hợp có thể để áp dụng. Tuy nhiên nó sẽ khó hơn trong việc thiết kế một thuật toán "end-to-end" cho tất cả các trường hợp.

"Maximum Likelihood Estimation"

2.2.2 Phương pháp "Variational Inference"

Inference là một lớp bài toán để giải quyết vấn đề tìm hiểu về nhưng thứ mà ta biết được dựa trên những thứ mà ta đã biết. Nói một cách khác thì bài toán này là tiến trình để có thể dưa ra kết luận cho một ước lượng, hay khoảng tin cậy hoặc xấp xỉ một phân phối về một "biến ẩn" (lateent variable) thường được gọi kết quả trong mẫu dữ liệu, dựa trên một vài các

biến mà ta đã quan sát được thường được gọi là nguyên nhân trong mẫu dữ liệu.

Một cách cụ thể thì, "Bayesian inferene" là quá trình đưa ra các suy diễn thống kê dựa trên "định lý Bayes". Phương pháp Bayesian là một phương pháp trong lĩnh vực xác suất thống kê mà ở đó kiến thức biết được biết trước "prior knowledge" được mô hình hoá bởi một phân phối xác suất và được cập nhật mỗi khi có một quan sát mới và những thứ mà ta không chắc chắn hay không quan sát được sẽ được mô hình bởi một phân phôi xác suất khác. Một ví dụ kinh diển là về các tham số của bayesian inference, giả định rằng một mô hình mà dữ liệu x được phát sinh từ một phân phối xác suất mà phân phối xác suất này được xác định bởi các tham số θ, tuy nhiên giá trị của θ thì ta chưa biết. Bên cạnh đó, ta giả định rằng, ta có một vài kiến thức được biết từ θ được gọi là "prior knowledge", nó có thể là phân phối xác suất $p(\theta)$. Sau đó, mỗi khi ta có một quan sát x mới, ta có thể cập nhật lại "prior knowledge" về tham số θ thông qua định lý Bayes theo công thức :

trong đó

Bayesian Inference là một vấn đề thường được phải giải quyết trong các bài toán trong lĩnh vực xác suất thống kê tuy nhiên trong lĩnh vực học máy, nhiều phương pháp được xây dựng dựa trên việc giải quyết vấn đề Bayesian Inference. Ví dụ: "Gaussian mixture models" được dùng để giải quyêt bài toán phân lớp, hay "Latent Dirichlet Allocation" để giải quyết bài toán phân loại chủ đề văn vản. Và cả hai mô hình kể trên đều được xây dựng dựa trên việc giải quyết bài toán Bayes Inference.

Computational diffficulties

Theo công thứ thì để tính toán "posterior" ta cần phải có: "prior", "likelihood" và "evedience". Hai giá trị ở trên tử số ta có thể dễ dàng ác định được trong hầu hết các trường hợp vì đó một phần là giả định của chúng ta về mô hình. Tuy nhiên, ở mẫu số ta cần tính:

để tính giá trị này với dữ liệu ở chiều không gian thấp có thể không gặp

nhiều khó khăn, nhưng khi tính toán ở những chiều không gian cao thì nó có thể trở thành một vấn đề nan giải. Khi dữ liệu có số chiều lớn thì việc tính chính xác giá trị "posterior" trong thực tiễn thường sẽ là một việc cực kỳ khó khăn và bất khả thi và ta cần một vài kĩ thuật xấp xỉ thường được dùng để giải quyết việc tính "posterior".

Chúng ta ca cần chú ý thêm một vài khó khăn khác có thể phải đối mặt khi giải quyết bài toán bayesian inference như là việc lấy "tổ hợp" khi dữ liệu là rời rạc thay vì giá trị liên tục.

Bài toán bayesian inference thông thường sẽ xuất hiện trong các phương pháp học máy mà giả định rằng có một Graphical model và khi mà cho trước một vài dữ liệu mà ta có thể quan sát được và mục đích của chúng ta là muốn tái tạo lại dữ liệu ẩn của mô hình. Xét ví dụ trong phương pháp Latent Dircichlet Allocation, một phương pháp để xác định chủ đề của một đoạn văn bản. Ta được cho trước một tập "từ điển" với kích thước V từ và có T chủ đề có thể có, mô hình này giải định rằng

- Với mỗi chủ đề, tồn tại một phân phối xác suất "topic-word" trên toàn bộ tập từ điển (giả định về prior)
- Với mỗi đoạn văn bản, có tồn tại một phân phối xác suất "document-topic" trên toàn bộ tập các chủ đề (một giả định prior khác)
- Với mỗi từ trong trong văn bản được lấy mẫu từ các phân phối giải định ở trên, cụ thể là đầu tiên chúng ta sẽ lấy mẫu một chủ đè từ phân phối xác suất "document-topic" của đoạn văn bản, tiếp theo, từ phân phối xác suất "topic-word" ta lấy mẫu một từ từ phân phối xác suất đi kèm với chủ đề được lấy mẫu ở bước trước.

Tên của phương pháp này là xuất phát từ giả định Dirichlet pior của mô hình. Mục tiêu của mô hình là có thể suy diễn "latent topic" từ từ điển ta quan sát được cũng như là có thể phân rã chủ đề của từng đoạn văn bản. Kể cả khi nếu chúng ta không đi sâu vào chi tiết của mô hình LDA, chúng ta có thể nói một cách đại khái rằng với w là một véc-tơ các từ có trong từ

điển và z là véc-tơ liên hệ với những từ đó, chúng ta muốn suy diễn được z dựa trên các quan sát từ w theo công thức bayes đó là:

có một công thức ở đây

Variational inference

Variational inference(VI) là một phương pháp thường hay được sử dụng để giải quyết bài toán bayesian inference. Phương pháp này sử dụng hướng tiếp cận là tìm ra xấp xỉ tốt nhất cho một phân phối xác suất bằng cách tìm ra bộ tham số tốt nhất định nghĩa cho phân phối.

Phương pháp VI bao gồm việc tìm ra một xấp sỉ tốt nhất cho một mục tiêu là phân phối xác suất giữa một lớp các phân phối xác suất cho trước. Cụ thể hơn, ý tưởng của VI đó là ta định nghĩa một lớp các phân phối xác suất và tìm ra trên đó bộ tham số sao cho ta có đạt được một phần tử gần nhất với mục tiêu của chúng ta tương ứng với một độ lỗi được định nghĩa cụ thể.

Ta xét một phân phối xác suất π được định nghĩa từ một "normalisation factor" C:

$$\pi(.) = C \times g(.)g(.)$$

Tiếp theo, về cho tiết toán học thì ta nếu ta ký hiệu lớp tham số hoá của các phân phối xác suất như sau

$$\mathcal{F}_\Omega = f_\omega; \omega \in \Omega \Omega \text{tập các tham số có thể có}$$

và chúng ta xét độ lỗi E(q,p) giữa hai phân phối xác suất p và q, việc tìm ra bộ tham số tốt nhất được thể hiện bởi:

$$\omega^* = \arg_{\omega \in \Omega} \min E(f_\omega, \pi)$$

Nếu chúng ta có thể thối thiểu hoá được bài toán trên mà không cần có một phân phối π được chuẩn hoá một cách chi tiết, ta có thể sử dụng f^*_ω như là một xấp xỉ để ước lượng thay vì phải tính toán các biểu thức mà

gần như không thể tính được khi ở chiều không gian lớn.

Do đó phương pháp này có thể dễ dàng được áp dụng và mở rộng cho những trường hợp mà ta cần giải quyết một bài toán với quy mô dữ liệu lớn.

Kullback-Leiber Devergence

2.2.3 Độ sai biệt "Kullback-Leiber Devergence" giữa hai phân phối xác suất

Chương 3

Mô hình "Variational Auto-Encoder" cho bài toán xây dựng hệ thống gợi ý

Chương này trình bày về những đóng góp của luận văn. Ở đây, Chúng tôi phân tích hai loại dữ liệu phản hồi chính từ người dùng là: "explicit feedback" và "implicit feedback". Đặc biệt, chúng tôi tập trung nghiên cứu mở rộng mô hình "Variational Auto-Encoders" cho implicit feedback với hàm lỗi là "Multinomial Log-likelihood" ở hàm mục tiêu. Chúng tôi gọi "Variational Auto-Encoders" với hàm lỗi như vậy là "Mult-VAEs". Đóng góp của chúng tôi là làm rõ Mult-VAEs ở hai điểm:

- *Tính xếp hạng: Chúng tôi chỉ ra điểm phù hợp của Multinomial Log-likelihood cho bài toán xây dựng hệ thống gợi ý sản phẩm so với các hàm Log-likelihood thông dụng khác.*
- *KL-Annealing: chúng tôi cũng đưa ra một cách "heuristic" nhằm lựa chọn siêu tham số của mô hình Mult-VAEs.*

3.1 Dữ liệu phản hồi của người dùng trong bài toán xây dựng hệ thống gợi ý sản phẩm

Như đã trình bày ở phần 1, để xây dựng một hệ thống gợi ý theo hướng tiếp cận "Collaborative filtering" ta chỉ cần dữ liệu là ma trận tương tác của người dùng. Tương tác ở đây có nghĩa là các phản hồi của người dùng dành cho sản phẩm, và các phản hồi này bao gồm hai loại:

- Phản hồi cụ thể "explicit feedback"
- Phản hồi ngầm "implicit feedback"

Trong phần này, chúng tôi sẽ làm rõ về tính chất của hai loại dữ liệu phản hồi cũng như ảnh hưởng của chúng đến hệ thống.

3.1.1 Dữ liệu phản hồi cụ thể "explicit feedback"

Dữ liệu phản hồi cụ thể ("explicit feedback") được hiểu là những phản hồi của khách hàng về sản phẩm một cách tường minh và cụ thể, ví dụ như: số điểm đánh giá, bình luận, ... "Explicit feedback" có thể thể hiện rõ về mức độ thích/không thích của người dùng về sản phẩm; ví dụ người dùng có thể thể hiện sự yêu thích của họ từ 1 đến 5 sao cho một sản phẩm (một cách đánh giá thông dụng), sản phẩm được đánh giá 5 sao chứng tỏ nó được thích hơn so với sản phẩm được đánh giá 4 sao. Trong thực tế, dữ liệu "explicit feedback" thường khó để thu thập cũng như gặp trở ngại về tính tin cậy. Thu thập loại dữ liệu này gặp khó khăn vì không phải người dùng nào cũng sẵn sàng phản hồi về sản phẩm. Sự miễn cưỡng của người dùng cũng như những tác động khi họ phản hồi có thể dẫn đến sự thiếu khách quan, làm sai lệch kết quả của hệ thống gợi ý. Thêm nữa, vì phản hồi của người dùng thể hiện mức độ thích/không thích của người dùng, mà người dùng thì chỉ tương tác với một lượng sản phẩm nhỏ trên

toàn hệ thống, những sản phẩm còn lại sẽ rơi vào trường hợp thiếu dữ liệu ("missing data"), gây khó khăn cho việc xử lí. Ngày nay, số lượng sản phẩm trong hệ thống là rất lớn, "explicit feedback" sẽ gặp khó khăn rất lớn khi có quá nhiều trường hợp thiếu dữ liệu, tác động đáng kể đến hiệu quả của hệ thống. Mặc khác, "collaborative filtering" sẽ có cơ sở đánh giá nhóm người dùng "tương đồng" với nhau một cách khắt khe hơn, giúp các gợi ý là những sản phẩm "tốt" hơn, tuy nhiên đôi lúc làm cho gợi ý không được đa dạng.

3.1.2 Dữ liệu phản hồi ngầm "implicit feedback"

Dữ liệu phản hồi ngầm ("implicit feedback") là dữ liệu được suy ra từ hành động của người dùng, nếu họ xem một bộ phim thì ta có thể hiểu là họ "thích" bộ phim đó. "Implicit feedback" cũng có thể được suy ra từ "tín hiệu ngầm" ("implicit signal"), xét ví dụ người dùng đánh giá một sản phẩm là 4 sao (trên thang đánh giá từ 1 đến 5 sao), từ "tín hiệu ngầm" dựa trên số sao họ đánh giá, ta có thể suy ra họ "thích" sản phẩm đó. "Implicit feedback" chỉ thể hiện rõ về sự "thích" cũng như chỉ thể hiện một cách tương đối mức độ yêu thích của người dùng. Cụ thể, người dùng không xem một bộ phim không có nghĩa là họ không thích bộ phim đó, có thể là họ chưa xem hoặc không biết nó có trên hệ thống. Cũng như họ xem một bài hát 10 lần chứng tỏ họ thích hơn so với một bài hát họ chỉ nghe 2 lần, và "implicit feedback" không thể thể hiện được rõ điều này. Trong thực tế, lượng dữ liệu phản hồi ngầm rất lớn và dễ dàng thu thập được, quá trình "phản hồi" của người dùng là bị động nên không bị ảnh hưởng bởi các yếu tố ngoại cảnh khác.

Ma trận tương tác của người dùng với dữ liệu phản hồi ngầm sẽ có dạng là một ma trận nhị phân, với giá trị **1** thể hiện người dùng "thích" sản phẩm đó, giá trị **0** thể hiện hệ thống chưa có cơ sở để xác định người dùng "thích" sản phẩm đó.

Với dữ liệu phản hồi ẩn, "collaborative filtering" sẽ xác định nhóm người

dùng "tương đồng" với nhau rộng hơn do chỉ quan tâm đến các sản phẩm họ thích. Điều này sẽ giúp các gợi ý của hệ thống đa dạng hơn, tuy nhiên các sản phẩm mà người dùng không thích cũng có thể sẽ được gợi ý.

Trong giới hạn luận văn này, chúng tôi chỉ tìm hiểu về một hệ thống gợi ý với dữ liệu phản hồi ngầm do tính khách quan cũng như giải quyết được các khó khăn của "explicit feedback".

3.2 "Multinomial log-likelihood" cho bài toán xây dựng hệ thống gợi ý

3.3 "Mult-VAEs"

3.3.1 Quá trình huấn luyện mô hình

3.3.2 Quá trình phát sinh gợi ý

3.4 Vấn đề "KL-Vanishing" trong việc huấn luyện "Variational Auto-Encoder"

3.4.1 Phương pháp "KL-Annealing"

1

Chương 4

Thí nghiệm

4.1 Tập dữ liệu sử dụng

4.2 Các thiết lập thí nghiệm

4.3 Các kết quả thí nghiệm

4.3.1 Kết quả mô hình cài đặt so với bài báo

4.3.2 Tại sao "Multinomial log-likelihood" phù hợp với bài toán xây dựng hệ thống gợi ý

4.3.3 So sánh với DAE

4.3.4 Vấn đề "KL-Vanishing"

4.3.5 Cải tiến...

Chương 5

Kết luận và hướng phát triển

5.1 Kết luận

5.2 Hướng phát triển

Tài liệu tham khảo

[1] Yifan Hu, Yehuda Koren, and Chris Volinsky. "Collaborative filtering for implicit feedback datasets". In: *2008 Eighth IEEE International Conference on Data Mining*. Ieee. 2008, pp. 263–272.

[2] Dawen Liang et al. "Variational autoencoders for collaborative filtering". In: *Proceedings of the 2018 world wide web conference*. 2018, pp. 689–698.

[3] Suvash Sedhain et al. "Autorec: Autoencoders meet collaborative filtering". In: *Proceedings of the 24th international conference on World Wide Web*. 2015, pp. 111–112.

[4] Harald Steck. "Gaussian ranking by matrix factorization". In: *Proceedings of the 9th ACM Conference on Recommender Systems*. 2015, pp. 115–122.

[5] Yao Wu et al. "Collaborative denoising auto-encoders for top-n recommender systems". In: *Proceedings of the ninth ACM international conference on web search and data mining*. 2016, pp. 153–162.