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 từ tổ chức Ivy Pro School [3], 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.

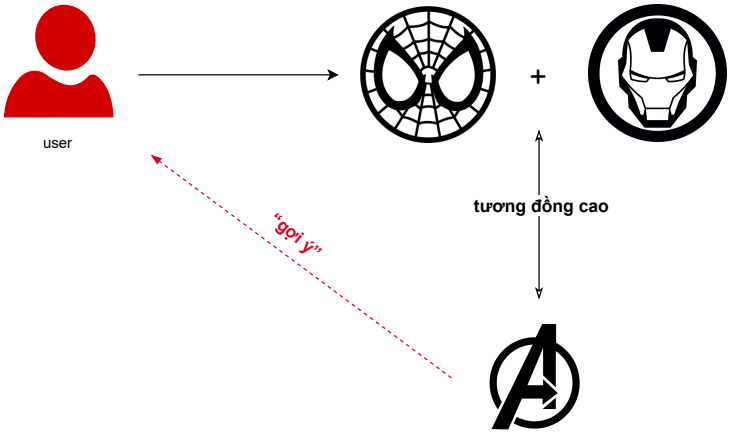
Trong lĩnh vực khoa học máy tính, hệ thống gợi ý sản phẩm là một chủ đề đang được quan tâm và nghiên cứu từ cộng đồng nghiên cứu khoa học. Bài toán xây dựng hệ thống gợi ý được phát biểu như sau:

- Đầu vào là lịch sử tương tác của người dùng (user) với các sản phẩm (items) hoặc có thêm các thông tin mô tả của sản phẩm (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.

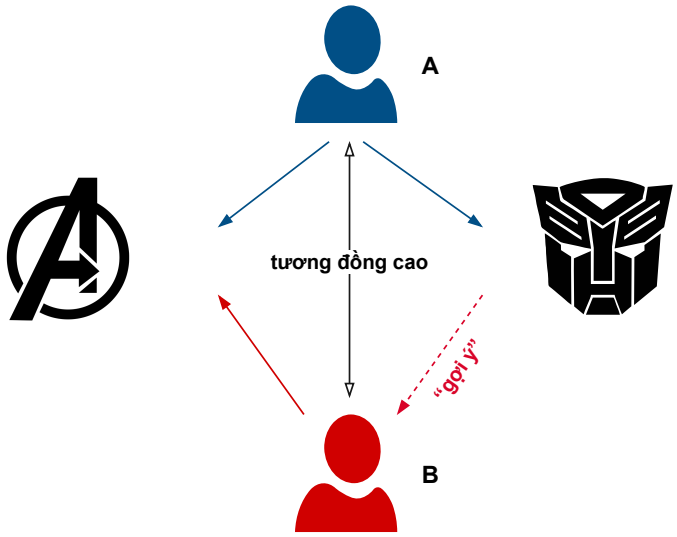
Tuy vậy, việc xây dựng một hệ thống gợi ý sản phẩm một cách hiệu quả là không đơn giản. Đầu tiên, không có một "lời giải" chung cho tất cả trường hợp, mặc dù đa số các lĩnh vực hiện nay đều có thể áp dụng các hệ thống gợi ý, tuy nhiên không phải là tất cả, ta cần xét đến nhiều yếu tố khác nhau, từ đó 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. Cũng theo số liệu từ Ivy Pro School [3], 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, ... vì 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.



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

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 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 một sản phẩm phù hợp với người dùng nhưng sản phẩm này có thể không được nhiều người dùng khác quan tâm. 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



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

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 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 khóa luận 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 giản 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 [2]. 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 rút trích các 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 số lượng tham số của mô hình giờ chỉ phụ thuộc vào số lượng sản phẩm có trong hệ thống. Trong thực tế, số lượng sản phẩm sẽ tăng chậm hơn đáng kể 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, với hạn chế của các hàm tuyến tính nên phương pháp này vẫn chưa thực sự mạnh. Ở công trình nghiên cứu [6] 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. Trong thời gian gần đây đã có nhiều nghiên cứu áp dụng mô hình "Auto-Encoder" trong bài toán xây dựng hệ thống gợi ý sản phẩm [5, 7, 4] 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" [5] được Sedhain giới thiệu tại hội nghị WWW2015, 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" [7] (CDAE) được Wu cùng các cộng sự đề xuất nhằm hướng đến bài toán đưa ra gợi ý theo hướng xếp hạng và đưa ra "top-N sản phẩm" phù hợp nhất 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 xếp hạng các sản phẩm phù hợp với người dùng thay vì tái tạo lại tương tác của họ. Ngoài ra, việc thêm "nhiễu" vào dữ liệu huấn luyện của CDAE giúp mô hình "Auto-Encoder" phải học cách trích xuất các đặc trưng ẩn tốt và tạo ra gợi ý thay vì chỉ cố gắng tái tạo lại dữ liệu tương tác "giống" đầu vào nhất có thể, cũng như "nhiễu" sẽ giúp quá trình huấn luyện mạng nơ-ron với hàm kích hoạt phi tuyến tránh được tình trạng "over-fitting" (tình trạng học "tủ" trên tập dữ liệu huấn luyện, dẫn đến mô hình đạt được kết quả thấp trên các tập dữ liệu kiểm định).

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"[4] được giới thiệu bởi tác giả Liang cùng các cộng sự công bố tại hội nghị "International World Wide Web Conference Committee 2018". 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. "Variational Auto-encoder" đã đạt được một số thành công nhất định trong bài toán xây dựng hệ thống gợi ý. Theo nghiên cứu của Dacrema [1], mô hình được đề xuất trong [4] đạt được các kết quả tốt trong việc gợi ý và xếp hạng. Trong khóa luận này, chúng tôi tìm hiểu và cài đặt lại mô hình được đề xuất trong bài báo [4], và thực

hiện một số thí nghiệm nhằm phân tích khả năng học của mô hình.

Với một mô hình "Auto-encoder" nói chung, ta thường chỉ quan tâm đến việc thu được véc-tơ biểu diễn ẩn có thể hiện đúng tính chất của dữ liệu đầu vào sau quá trình huấn luyện. Ứng dụng của mô hình này thường là trích xuất các đặc trưng ẩn hoặc giảm chiều dữ liệu - đồng nghĩa với việc tái tạo dữ liệu là một quá trình phụ trợ cho việc huấn luyện mô hình. Nghĩa là mô hình sẽ tái tạo hoặc trích xuất được các "đặc trưng ẩn" của dữ liệu mà không thể sinh ra dữ liệu mới. Ứng dụng của "Auto-encoder" trong các bài báo [5, 7] trong hệ thống gợi ý là cố gắng tái tạo lại "đặc trưng ẩn của người dùng", từ đó cũng sinh ra các gợi ý. Vô tình, ý tưởng này dẫn đến rằng việc đưa ra gợi ý không phải là mục đích chính của mô hình, mà gợi ý được tạo một cách gián tiếp thông qua việc tái tạo lại dữ liệu. Với bài toán xây dựng hệ thống gợi ý, việc tạo ra các gợi ý cho người dùng phải là một việc được ưu tiên. Mô hình VAEs với "đặc trưng ẩn" được phát sinh từ một phân phối xác suất, và được xem như một mô hình có thể phát sinh dữ liệu mới từ "đặc trưng ẩn". Khi ứng dụng trong bài toán xây dựng hệ thống gợi ý, việc phát sinh dữ liệu mới này tương đồng với việc tạo ra các gợi ý, đồng thời cũng vẫn giữ được tính chất của "Auto-encoder" trong bài toán này là tương tác người dùng được phát sinh từ một "đặc trưng ẩn" (cũng chính là sở thích của người dùng).

Hơn nữa, với nền tảng của mô hình VAEs dựa trên phương pháp "Variational Inference" trong lĩnh vực xác suất thống kê. "Variational Inference" dùng để suy diễn dữ liệu ẩn từ dữ liệu ta quan sát được, hay cụ thể trong bài toán này là suy diễn ác "đặc trưng ẩn" dựa vào dữ liệu quan sát được là các tương tác của người dùng. 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. Trong hệ thống gợi ý, tính chất của dữ liệu thường là thưa, do mỗi người dùng chỉ tương tác với một lượng nhỏ sản phẩm trên toàn hệ thống, từ đó việc suy diễn trở nên hiệu quả trong hệ thống gợi ý.

Cùng với mục tiêu đánh giá xếp hạng các sản phẩm, với mục đích sản

phẩm liên quan đến người dùng hơn được ưu tiên gợi ý hơn. Tác giả Liang đã giới thiệu "Multinomial log-likelihood" cho việc tính toán độ lỗi. Với tính chất trả về một giá trị xác suất cho mỗi sản phẩm, và tổng giá trị xác suất trên toàn bộ sản phẩm là 1. Ngoài ra "likelihood" sẽ khuyến khích mô hình đánh giá các sản phẩm là một giá trị lớn hơn 0. Các sản phẩm sẽ phải "cạnh tranh" với nhau để có được xác suất cao hơn. 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.

Mặc dù mô hình được tác giả Liang đề xuất ở bài báo [4] 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 với những lí do nêu trên, chúng tôi tin rằng 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. Vậy nên chúng tôi quyết định sẽ tập trung tìm hiểu mô hình này.

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, tổng kết và hướng phát triển được trình bày ở chương 5.

Chương 2

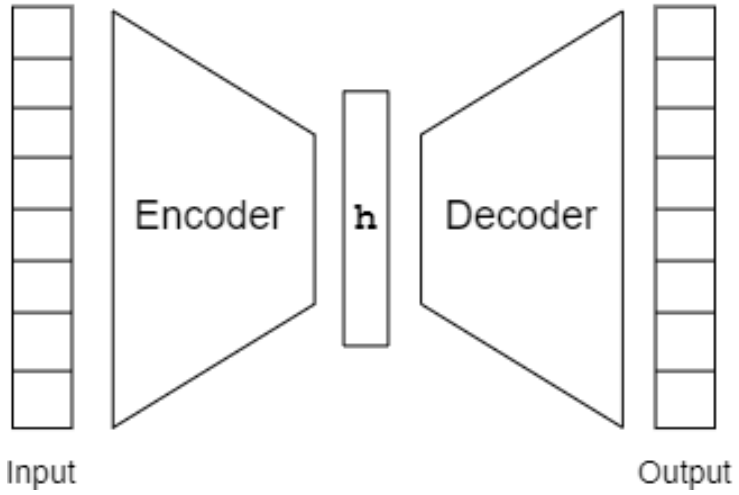
Kiến thức nền tảng

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.

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



Hình 2.1: Minh họa "Auto-Encoder"

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. Với mục đích thu được véc-tơ biểu diễn ẩn của dữ liệu thông qua quá trình huấn luyện, ta cần các ràng buộc để có được $\boldsymbol{h}$ nhận các thuộc tính hữu ích 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. Đây cũng là kiến trúc đa số các mô hình "Auto-Encoder" thường hay được

sử dụng.

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"

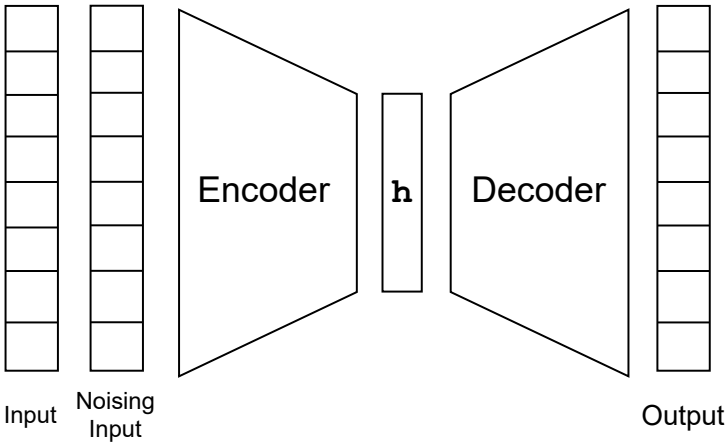
Hàm lỗi của một "Auto-Encoder" thông thường sẽ "phạt" một mức nhất định với các mẫu dữ liệu được tái tạo lại khác với dữ liệu đầu vào. Điều này vô hình chung khuyến khích việc $f \circ g$ là một hàm đồng nhất nếu khả năng tính toán của f và g cho phép. Nói đơn giản hơn, điều này là việc mô hình sao chép "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).

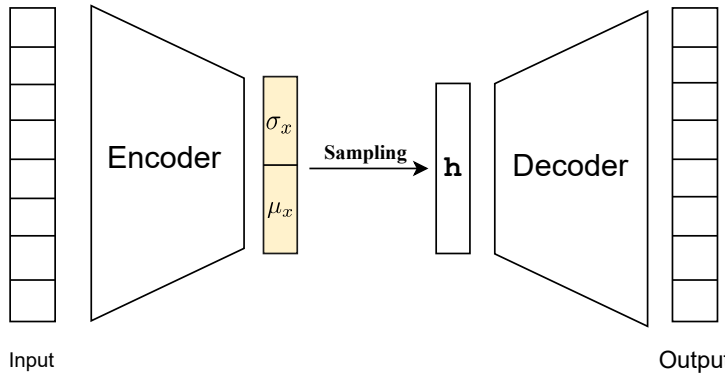
"Denoising Auto-Encoder" phải học cách khử độ nhiễu đã được thêm vào véc-tơ đầu vào, giảm khả năng sao chép của mô hình.



Hình 2.2: Minh họa "Denoising Auto-Encoder"

2.2 Mô hình "Variational Auto-Encoder"

"Variational Auto-encoder" (VAEs) là một biến thể đặc biệt của "Auto-encoder" cơ bản (được minh họa trong hình 2.3). 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-Encoder" 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 kích thước của véc-tơ biểu diễn ẩn của một "Auto-Encoder" sẽ có tính chất "under-complete" như đã đề cập ở phần 2.1.1). Do đó, ta có thể xem đặc trưng ẩn này 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. Mặt khác, với 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



Hình 2.3: Minh họa "Variational Auto-Encoder"

đó sẽ tương ứng với giá trị xác suất là bao nhiêu.

Tuy nhiên, để làm rõ được sự hiệu quả của VAEs trong việc phát sinh đặc trưng và phát sinh dữ liệu từ đặc trưng thì ta cần phải xét qua góc nhìn xác suất của mô hình này. 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 mục tiếp theo, cũng như là cách huấn luyện cho mô hình VAEs.

2.2.1 Nền tảng xác suất của mô hình

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

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 "pattern" trong dữ liệu và sau đó sử dụng những "pattern" 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 phối 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

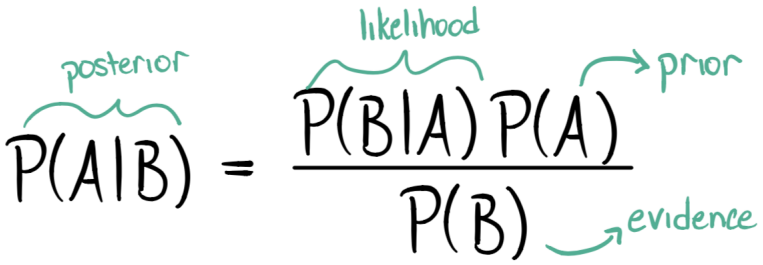
Trong lĩnh vực "máy học" và "thống kê Bayesian", chúng ta thường quan tâm đến việc thực hiện các phép suy diễn dữ liệu ẩn ta không quan sát được khi cho trước các dữ liệu ta quan đã quan sát được. Ví dụ như ứng dụng một mô hình học máy trong việc phát hiện sản phẩm lỗi, khi ta đã có ghi nhận lại một số lượng dữ liệu mô tả của sản phẩm và đã biết được sản phẩm nào có lỗi hay không, đây chính là dữ liệu mà ta đã quan sát được. Điều mà ta quan tâm đến đó là mô hình có thể phát hiện được những "pattern" hay những đặc trưng quyết định đến việc xác định một sản phẩm có được xem là sản phẩm lỗi hay không, thì những "pattern" hoặc những đặc trưng này sẽ được xem là những dữ liệu ẩn.

Giả sử rằng, ta có a là biến ngẫu nhiên thể hiện cho dữ liệu ẩn mà ta không có dữ liệu về nó, và b là biến ngẫu nhiên của dữ liệu mà ta có thể quan sát được. Theo đó, ta sẽ quan tâm đến việc tìm ra được giá trị a cụ thể khi cho trước giá trị b. Về xác suất, hay cụ thể ở đây, theo định lý Bayes, nếu ta có $p(a)$ là thông tin mà ta đã biết trước về biến ta không quan sát được a; và một số mẫu dữ liệu thể hiện mối quan hệ giữa a và b được thể hiện bởi $p(b|a)$, theo công thức Bayes, ta có:

$$p(a|b) = \frac{p(b|a)p(a)}{p(b)} \tag{2.3}$$

Trong đó:

- $p(a)$ được gọi là "prior", prior thể hiện cho "kiến thức biết trước" theo góc nhìn chủ quan ban đầu của chúng ta trước khi ta có bất kỳ về thông tin nào về liệu mà ta quan sát được. prior có thể được thể hiện thông qua một phân phối xác suất theo biến ẩn, nó có thể là phân phối xác suất bất kỳ sao cho phù hợp với chúng ta, nhưng một điều chúng ta cần phải đảm bảo đó là phân phối prior phải là có giá trị



Hình 2.4: Định lý "Bayes"

khác không trên tất cả các giá trị có thể xuất hiện của a, kể cả khi giá trị đó rất hiếm khi xảy ra.

- $p(b|a)$ là "likelihood", mô tả mối quan hệ giữa a và b liên quan với nhau như thế nào, và cụ thể thì nó là khả năng của việc xảy ra giá trị b khi ta đã biết về dữ liệu ẩn a cụ thể.

- Phân phối "posterior" $p(a|b)$ sẽ là giá trị mà ta quan tâm theo quan điểm của Bayes. Nó thể hiện rằng chúng ta có được thông tin gì về dữ liệu ẩn a không quan sát được khi ta có dữ liệu quan sát được là b.

- $p(b)$ là "evidence" hay cũng còn được biết đến là "marginal likelihood". Phân phối thể hiện cho khả năng xảy ra của một giá trị B cụ thể. Ngoài ra evidence độc lập với a nó còn có vai trò để chuẩn hoá cho posterior, có nghĩa là posterior sẽ có khoảng giá trị từ 0 đến 1.

Các đại lượng trong công thức 2.3 được chú thích trong hình 2.4

Khó khăn trong việc tính toán

Với góc nhìn này, "posterior" sẽ là giá trị mà chúng ta quan tâm đến, nó thể hiện mối quan hệ giữa "prior" của chúng ta và dữ liệu. Việc tính toán "posterior" sẽ giúp chúng ta giải quyết các vấn đề trong thực tế. Theo công thức 2.3, để tính toán "posterior" ta cần phải có: "prior", "likelihood" và "evidence". Hai giá trị ở trên tử số ("prior" và "likelihood") ta có thể dễ

dàng xác 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:

$$p(b) = \int p(b|a)p(a)\mathrm{d}a = \mathbb{E}_a[p(b|a)] \tag{2.4}$$

theo đó ta thấy được để tính được "marginal likelihood" thì ta cần tính biểu thức với dấu tích phân. 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. Cụ thể ta thấy được rằng việc tính "marginal likelihood" sẽ thể hiện giá trị "likelihood" trung bình trên toàn bộ giá trị có thể xuất hiện của x, do đó x ở chiều không gian càng cao thì việc tính toán càng trở nên phức tạp hơn.

Chúng ta ca cần chú ý thêm một vài khó khăn khác có thể phải đối mặt khi tính toán "posterior" đó là việc lấy "tổ hợp" khi dữ liệu là rời rạc thay vì giá trị liên tục. Ở miền không gian liên tục thì ta có thể áp dụng hàm số trong lĩnh vực giải tích để tính toán, tuy nhiên trong những trường hợp mà chiều không gian của dữ liệu không liên lục, dữ liệu rời rạc thì việc tính toán sẽ còn phải xét thêm việc lấy tổ hợp dữ liệu.

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".

Bài toán 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 giá trị ước lượng, hay khoảng tin cậy hoặc xấp xỉ một phân phối cho một "biến ẩn" ("latent variable") thường được gọi là kết quả hay nhãn trong mẫu dữ liệu, dựa trên một vài các biến mà ta đã quan sát được thường được gọi là nguyên nhân hay dữ là dữ liêu đầu vào trong mẫu dữ liệu. Ví dụ như ta có dữ liệu là

hình ảnh của các đối tượng trong tự nhiên và có nhãn đi kèm mỗi ảnh, bài toán inference sẽ trả lời câu hỏi rằng nếu một tấm ảnh mới không có trước đó thì liệu ta có biết được nhãn của đối tượng trong ảnh hay không?.

"Bayesian inferene" là việc giải quyết bài toán inference dựa trên "định lý Bayes". Phương pháp Bayesian inference 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.

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 giá trị của bộ tham số tốt nhất định nghĩa cho một phân

phối khác, sao cho phân phối này sẽ "gần" với phân phối mà ta quan tâm.

Với phương pháp VI, đầu tiên ta sẽ tìm một phân phối xác suất có cùng "họ" (family) với phân phối xác suất mà ta quan tâm. Một họ phân phối xác suất là tập các phân phối xác suất được định nghĩa bởi cùng một bộ tham số. Ví như họ phân phối Gaussian sẽ được định nghĩa bởi μ là giá trị kỳ vọng (mean) và σ là độ lệch chuẩn (standard deviation) Việc lựa chọn "họ" phân phối sẽ kiểm soát giữa độ phức tạp và độ chính xác của của phương pháp này. Nếu ta giả định rằng dữ liệu tuân theo một phân phối đơn giản thì kết quả suy diễn được sẽ không quá chính xác nhưng có thể dễ dàng tìm được nghiệm tối ưu. Ngược lại nếu ta lựa chọn "họ" phân phối phức tạp thì sẽ khó tìm đươc nghiệm tối ưu nhưng kết quả suy diễn sẽ có kết quả tốt hơn.

Sau khi xác định được "họ" phân phối xác suất dùng để xấp xỉ phân phối xác suất mà chúng ta quan tâm thì việc tiếp theo là làm sao để tìm ra được xấp xỉ tốt nhất.

Giả sử rằng chúng ta cần sấp xỉ phân phối p bởi một phân phối q cùng thuộc họ phân phối $\mathcal{F}$. Chúng ta xét độ lỗi $\mathbb{E}(q, p)$ giữa hai phân phối xác suất p và q, việc tìm ra bộ tham số tốt nhất được thể hiện bởi:

$$q^* = \arg_{q \in \mathcal{F}} \min \mathbb{E}(q, p) \tag{2.5}$$

Vậy trong bài toán variational inference thì làm sao để xác định hai phân phối xác suất có "gần" nhau hay không hay làm sao để xác định độ lỗi $\mathbb{E}(q, p)$. Sự sai biệt Kullback-Leiber (KL) là một cách để tính mức độ lệch của một phân bố đối với một phân bố được chỉ định và thường được sử dụng để đo sự khác nhau giữa hai phân phối xác suất.

KL là môt thuật ngữ đến từ lĩnh lý thuyết thông tin, nó còn có tên gọi khác là entropy tương đối. Nói theo ngôn ngữ lý thuyết thông tin, nó đo lượng trung bình thông tin thêm vào nếu chúng ta mã hóa thông tin của phân bố q thay cho mã hóa thông tin phân bố p.

Nếu $p(x)$ và $q(x)$ là hai phân phối xác suất với x là biến ngẫu nhiên

bất kỳ, thì sự sai biệt KL sẽ được định nghĩa như sau:

$$KL(q,p) = \mathbb{E}_q[\log q(x)] - \mathbb{E}_q[\log p(x)] \tag{2.6}$$

Sau khi xác định được một hàm lỗi để xấp xỉ phân phối xác suất mà chúng ta quan tâm p, bởi phân phối q thì bài toán sẽ trở thành việc tìm phân phối q^* như sau:

$$q^* = \arg_{q\in\mathcal{F}}\min\mathbb{E}_q[\log q(x)] - \mathbb{E}_q[\log p(x)]$$

Việc tìm ra phân phối xác suất tốt nhất này trở thành một bài toán tìm nghiệm tối ưu 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.

"Maximum Likelihood Estimation"

Trong lĩnh vực máy học, chúng ta sử dụng một mô hình để mô tả một tiến trình mà tổng hợp, phân tích tự động dữ liệu được thu thập để có thể giải quyết các vấn đề như tìm ra các đặc trưng, đưa ra các dự đoán dựa trên dữ liệu quan sát được. Xét ví dụ bài toán *Hồi quy tuyến tính* ("Linear regression"), ta cần dự đoán giá trị y dựa trên giá trị của véc-tơ x theo công thức:

$$y = wx + b \tag{2.7}$$

Điều ta cần làm ở mô hình này là "ước lượng" giá trị tham số w và b để mô hình có thể dự đoán giá trị y một cách tốt nhất.

Với một mô hình máy học được mô tả bởi bộ tham số θ, ta cần thực hiện "ước lượng" bộ tham số θ sao cho mô hình có thể trả về kết quả tốt nhất. Thay vì dự đoán một hàm số nào đó có khả năng ước lượng tham số tốt, ta cần một nguyên tắc để suy ra các hàm số cụ thể cho các mô hình khác nhau. "Maximum likelihood Estimation" (MLE) là một công cụ phổ biến để thực hiện việc này.

Xét tập dữ liệu gồm m phần tử $\mathbb{X} = \{x_1, x_2, ..., x_m\}$ là độc lập với nhau và được phát sinh từ một phân phối xác suất $p_{data}(x)$. Giả sử mô hình được mô tả bởi bộ tham số θ. Khi đó, $p(x|\theta)$ là xác suất xảy ra sự kiện x khi ta biết θ. $p(x_1, x_2, ..., x_m|\theta)$ chính là xác suất các sự kiện $x_1, x_2, ..., x_m$ xảy ra đồng thời, xác suất đồng thời này được gọi là "likelihood". "Likelihood" của mô hình thể hiện khả năng mà bộ tham số của mô hình thể hiện mối quan hệ giữa dữ liệu mà ta có. Quá trình cực đại hóa "likelihood" là việc tối ưu khả năng mô hình thể hiện đúng nhất có thể mối quan hệ của dữ liệu. MLE là việc đi tìm bộ tham số θ sao cho likelihood là lớn nhất:

$$\theta = \max_{\theta} p(\mathbb{X}; \theta) \tag{2.8}$$

Vì các phần tử trong $\mathbb{X}$ là độc lập và cố định, do đó công thức 2.8 tương đương với:

$$\theta = \max_{\theta} \prod_{i=1}^{m} p(x_i; \theta) \tag{2.9}$$

Bài toán MLE là quá trình tối ưu công thức 2.9. Mà công thức này là một tích, thường thì việc tối ưu hóa một tích sẽ gặp rất nhiều khó khăn trong việc tính toán. Thay vào đó, ta tối ưu hàm logarit của "likelihood" bởi vì:

- logarit là một hàm đồng biến, "likelihood" sẽ lớn nhất khi logarit của 'likelihood' là lớn nhất.
- logarit của một tích sẽ bằng tổng các logarit

Khi đó, bài toán MLE được đưa về bài toán "Maximum log-likelihood Estimation"

$$\theta = \max_{\theta} \sum_{i=1}^{m} \log(p(x_i; \theta)) \tag{2.10}$$

Gaussian likelihood ám chỉ "likelihood" của mô hình với việc phân phối xác suất $p_{data}(x)$ thuộc "họ" Gaussian. Tương tự, các hàm likelihood khác

cũng thường dùng trong các bài toán máy học là: *Bernoulli likelihood, Multinomial likelihood.*

2.2.2 Mô hình "Variational Auto-encoder"

Mô hình xác suất

Mô hình xác suất là một mô hình được dùng để mô tả một phân phối xác suất hợp của dữ liệu 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". Một graphical model sẽ thể hiện phân phối hợp như sau:

$$p(x_1, x_2, .., x_N)$$

trong đó $[x_1, x_2, .., x_N]$ là các đặc trưng dữ liệu, hoặc các tham số của mô hình,... Ví dụ như với bài toán phân loại thì một graphical model sẽ thể hiện cho phân phối $p(y, x, \theta)$ trong đó x là đặc trưng đầu vào, y là nhãn của dữ liệu và θ là trọng số của mô hình.

Graphical model cũng chính là một nhánh trong học máy, bằng cách thể hiện bài toán học máy dưới dạng một đồ thị. Sự kết hợp giữa xác suất vào một mô hình học máy sẽ giúp mô hình "giải thích" vấn đề thực tế một cách tốt hơn. Theo đó thì vấn đề cần quan tâm hoặc dữ liệu liên quan bởi các biến ngẫu nhiên. Và các biến ngẫu nhiên này chính là các đỉnh trong đồ thị và mối quan hệ giữa các đỉnh sẽ hình thành cạnh. Bằng cách thể hiện mối quan hệ giữa các biến dữ liệu bởi đồ thị ta dựa vào đó để có thể giải quyết các vấn đề mà chúng ta quan tâm. Ngoài ra bằng cách thể hiện bằng đồ thị, ta cũng thể hiện được tương quan giữa các biến dữ liệu, mà đa số các thuật toán máy học hiện nay thì tính tương quan giữa dữ liệu sẽ

ảnh hưởng đến kết quả của mô hình. Dữ liệu có tính tương quan càng cao thì có thể sẽ ít đóng góp thêm "thông tin" cho mô hình thì có thể dẫn đến việc kết quả mô hình mang lại sẽ không cao. Do đó khi xây dựng thuật toán hay mô hình cho một vấn đề cụ thể, ta có thể áp dụng các kiến thức ta biết trước về lĩnh vực đó, về dữ liệu để có thể xác định các đặc trưng cho mô hình thông qua đồ thị. Ngoài ra việc sử dụng graphical cũng sẽ cung cấp một cái nhìn tổng quan về mô hình cũng như dữ liệu, từ đó việc phân tích, thiết kế và cài đặt cũng sẽ dễ dàng hơn.

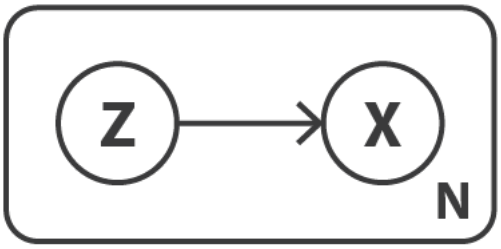
Variational Auto-encoder dưới góc nhìn xác suất

Với góc nhìn của một mô hình mạng nơ-ron thì "Variational Auto-encoder" chỉ là một mạng nơ ron đơn giản với cấu trúc hai phần như mô hình "Auto-encoder" tổng quát, gồm encoder và decoder. Encoder được dùng để trích xuất đặc trưng ẩn từ dữ liệu, điểm khác biệt so với "Auto-encoder" cơ bản là encoder của VAEs sẽ là một phân phối xác suất. Và tương tự thì decoder sẽ là mạng nơ-ron cố gắng để tái tạo lại dữ liệu ban đầu từ đặc trưng ẩn.

Tuy nhiên để hiểu rõ hơn về nền tảng toán học cũng như xác suất trong mô hình chúng tôi sẽ trình bày mô hình VAEs dưới góc độ là một mô hình xác suất. "Variational Auto-enncoder" là một mô hình đồ thị có hướng mô tả mối quan hệ giữa dữ liệu quan sát được và đặc trưng ẩn. Một đồ thị có hướng là một graphical model mà các đỉnh được kết nối với nhau có thứ tự. Có nghĩa là để có học được mẫu ở nút "cha" thì trước đó ta cần mô hình hoá được dữ liệu ở nút con. Bên cạnh đó thì một graphical có hướng còn có tên gọi khác là "Bayesian network".

Xét graphical model thể hiện cho mô hình VAEs trong hình 2.5: "Variational Auto-encoder" bao gồm một biến x thể hiện cho dữ liệu, đây là biến dữ liệu quan sát được, và z là biến ẩn thể hiện cho đặc trưng ẩn của dữ liệu. Là một đồ thị có hướng do đó quá trình phát sinh dữ liệu của VAE được thực hiện qua các bước theo thứ tự như sau:

Với mỗi điểm dữ liệu:



Hình 2.5: Graphical model thể hiện cho mô hình "Variational Auto-encoder". Dữ liệu quan sát được sẽ được giả định được phát sinh từ biến ẩn z

- Đặc trưng ẩn z_i được lấy mẫu từ phân phối p(z)
- Điểm dữ liệu x_i được lấy mẫu từ phân phối p(x|z)

Cụ thể, biến đặc trưng ẩn z được chọn ra từ một phân phối "prior" $p(z)$ chính là những kiến thức ta biết trước ta biết trước hoặc là giả định của chúng ta về z. Điểm dữ liệu x có một phân phối likelihood $p(x|z)$ thể hiện quan hệ giữa dữ liệu ta có với đặc trưng ẩn. Mô hình định nghĩa một phân phối hợp của dữ liệu và đặc trưng ẩn: $p(x, z)$. Với quy tắc nhân trong xác suất, chúng ta có thể phân tách phân phối hợp trên thành "prior" và "likelihood" như sau: $p(x, z) = p(x|z)p(z)$. Đây chính là mục tiêu chính khi ta xét "Variational Auto-encoder" dưới góc độ của xác suất.

Mô hình này sử dụng phương pháp Variational inference được nhắc đến ở phần 2.2.1 để tìm ra đặc trưng ẩn, đây cũng là lý do dẫn đến cái tên "Variational Auto-encoder". Và VAEs có thể được huấn luyện dựa trên các thuật toán học dựa trên gradient truyền thống.

Tiếp theo, ta xét đến việc suy diễn (inference) trong mô hình này. Mục tiêu của việc suy diễn là tìm ra được một giá trị "tốt" thể hiện cho đặc trưng ẩn khi ta có các điểm dữ liệu, hay nói cách khác là ta tính "posterior".

Theo công thức Bayes:

$$p(a|b) = \frac{p(b|a)p(a)}{p(b)} \tag{2.11}$$

theo như những gì đã trình bày ở những phần 2.2.1, phần mẫu số của công thức 2.11 được gọi là "marginal likelihood". Và "marginal likelihood" sẽ không dễ dàng tính toán được một cách chính xác khi đặc trưng ẩn ở không gian có số chiều cao.

Do đó, phương pháp Vairational inference (VI) được áp dụng để xấp xỉ phân phối "posterior" $p(z|x)$ này. VI xấp xỉ posterior thông qua một phân phối xác suất có cùng họ phân phối $q_\lambda(z|x)$. Trong đó λ thể hiện cho bộ tham số định nghĩa cho phân phối q, ví dụ nếu q là họ phân phối Gaussian thì $\lambda = (\mu; \sigma^2)$.

Tiếp đến, "Kullback-Leiber Devergence" được sử dụng để xấp xỉ "posterior" với:

$$\mathbb{KL}(q_\lambda(z|x)||p(z|x)) = \mathbb{E}_q[\log q_\lambda(z)] - \mathbb{E}_q[\log p(z|x)] \tag{2.12}$$

trong đó các kỳ vọng được lấy theo $q_\lambda(z)$.

Biến đổi xác suất có điều kiện $p(z|x)$ ở công thức 2.12, ta có:

$$\mathbb{KL}(q_\lambda(z|x)||p(z|x)) = \mathbb{E}_q[\log q_\lambda[(z|x)] - E_q[logp(x,z)] + \log p(x) \tag{2.13}$$

Mục tiêu của chúng ta là tìm ra bộ tham số λ sao cho tối thiểu được sự sai biệt trên. với q^* là phân phối q lý tưởng để xấp sỉ posterior thì ta có:

$$q^*_\lambda(z|x) = \arg\min_\lambda KL(q_\lambda(z|x)||p(x)) \tag{2.14}$$

Tuy nhiên ta vẫn chưa có thể tính được trực tiếp bởi trong công thức thì vân còn xuất hiện $p(x)$, cho nên "marginial likelihood" vẫn không thể tính một cách trực tiếp.

Vì không thể tính một cách trực tiếp ta xét một hàm số thay thế khác

như sau:

$$ELBO(q_\lambda) = \mathbb{E}[\log p(z, x)] - \mathbb{E}[\log q_\lambda(z)] \tag{2.15}$$

Biến đổi phân phối hợp $p(z, x)$ trong 2.15, ta lại có:

$$\begin{aligned} ELBO(q_\lambda) &= \mathbb{E}[\log p(z)] + \mathbb{E}[\log p(x|z)] - \mathbb{E}[\log q_\lambda(z)] \\ &= \mathbb{E}[\log p(x|z)] - \mathbb{KL}(q_\lambda(z)||p(z)) \end{aligned} \tag{2.16}$$

Công thức 2.16 được gọi là "evidence lower bound" (ELBO). ELBO là trừ của sai biệt KL cộng với một lượng $\log p(x)$, bởi $\log p(x)$ là hằng số theo $q_\lambda(z)$. Bây giờ, việc cực đại ELBO sẽ tương đương với việc tối thiểu độ sai biệt KL.

Bên cạnh đó, sau khi biến đổi mở rộng ELBO thì ta thấy rằng ELBO được tính từ 2 phần đó là kỳ vọng của likelihood và trừ KL giữa prior và phân phối sấp xỉ $q_\lambda(z)$. Với kỳ vọng của likelihood, nó thể hiện khả năng mô hình hoá dữ liệu còn sai biệt KL thì đảm bảo việc phân phối được xấp xỉ sẽ gần với prior của dữ liệu. Đây chính là sự đánh đổi thường gặp trong những bài toán Bayesian inference, đánh đổi giữa khả năng mô hình hoá dữ liệu và việc đảm bảo phân phối được xấp sỉ hay cụ thể là đặc trưng ẩn sẽ gần với prior của chúng ta.

Sau khi nắm được lý thuyết nền tảng về xác suất của mô hình VAEs, tiếp theo ta sẽ liên hệ với mạng nơ ron để thể hiện cho mô hình. Mục tiêu cuối cùng của chúng ta là tìm ra bộ tham số có thể xấp xỉ được posterior $p(z|x, \lambda)$ bằng $q_\theta(z|x, \lambda)$ thông qua một mạng nơ ron thường được gọi là "inference network" hay chính là encoder. Mạng này nhận đầu vào là dữ liệu x và trả về kết quả là bộ tham số λ thể hiện phân phối xác suất của đặc trưng ẩn. Bên cạnh đó, sẽ có một mạng nơ ron được gọi là "genetive network" hay còn được biết đến là decoder thể hiện cho likelihood $p_\phi(x|z)$ của mô hình. Tiếp theo đặc trưng ẩn sẽ được lấy mẫu từ phân phối xác suất trả ra từ encoder, đây chính là bước đầu tiên trong mô hình xác suất đã được nói ở trên. Dữ liệu đầu vào của decoder là đặc trưng ẩn được lấy

mẫu dựa trên phân phối của đặc trưng ẩn, và decoder sẽ cố gắng tái tạo lại dữ liệu ban đầu từ đặc trưng ẩn z, đây chính là bước thứ hai trong mô hình xác suất.

Inference network và generative network sẽ có bộ tham số tương ứng là θ và ϕ. Thông thường bộ tham số là các trọng số và bias trong một mạng nơ ron thông thường. Mục tiêu của mô hình sẽ là cực đại hàm ELBO tương ứng như sau:

$$ELBO(\theta, \phi) = \mathbb{E}_{q_\theta}(z|x)[\log p_\phi(x|z)] - \mathbb{KL}(q_\theta(z)||p(z)) \tag{2.17}$$

Để huấn luyện mô hình, ta có thể sử dụng những thuật toán học như "gradient descent" để có thể tìm ra bộ trọng số θ, ϕ.

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 khóa luận. 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 gợi ý.

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. Mặt 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 các 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 của khóa luận 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 Áp dụng mô hình "Auto-Encoder" để xây dựng hệ thống gợi ý sản phẩm

Phương pháp xây dựng mô hình gợi ý sản phẩm mà chúng tôi tìm hiểu đó là sử đụng kiến trúc auto-encoder, một mạng nơ ron nhận input đầu vào là một phần tương tác của người dùng trong quá khứ và mô hình được huấn luyện để tái tạo lại toàn bộ tương tác của người dùng. Mô hình này là một mạng nơ ron bao gồm hai thành phần:

- Mạng nơ-ron encoder có chức năng rút trích đặc trưng ẩn từ những tương tác của người dùng trong quá khứ
- Mạn nơ-ron decoder có chức năng tái tạo lại tương tác của người dùng

Mặc dù cách hoạt động của mô hình trong giai đoạn huấn luyện và trong giai đoạn kiểm tra có phần khác nhau, nhưng ở phần này chúng tôi chỉ trình bày cách hoạt động của mô hình trong giai đoạn huấn luyện để qua đó diễn giải kiến trúc mô hình một cách thuận tiện. Và sau đó chúng tôi sẽ trình bày cách để đưa ra gợi ý cho người dùng mới sau.

Với tập dữ liệu được sử dụng huấn luyện gồm U người dùng $u_1, u_2, .., u_U$ và cần xây dựng mô hình đưa ra gợi ý trong tập I sản phẩm $i_1, i_2, ..., i_I$. Bên cạnh đó, dữ liệu tương tác của người dùng với các sản phẩm sẽ được thể hiện bởi một ma trận tương tác $X \in \mathbb{N}^{U \times I}$. Tương tác một người dùng sẽ là một véc-tơ $x_u = [x_{u1}, x_{u2}, ..., x_{uI}]^T \in \mathbb{N}^I$ với $u \in U$.

Với giả định rằng việc một người dùng tương tác với các sản phẩm sẽ đến từ những đặc trưng ẩn của người dùng đó. Việc trích xuất đặc trưng ẩn của ngừoi dùng dựa trên những tương tác với hệ thống trong quá khứ được hi vọng rằng những đặc đặc trưng ẩn sẽ có thể được sử dụng để thể hiện cho người dùng. Sau khi có được đặc trưng ẩn của người dùng thì vì dựa trên mục tiêu rằng đặc trưng sẽ thể hiện tốt được những yếu tố quyết định đến các tương tác của người dùng thì việc huấn luyện mô hình sẽ cần phải đảm bảo rằng đặc trưng ẩn có thê dùng để hình thành tương tác của người dùng.

Mục tiêu của mô hình là nhận đầu vào là tương tác của người dùng, tương tác này được chiếu vào một không gian đặc trưng có số chiều thấp hơn và sau đó lại tái tạo lại dữ liệu ban đầu với mục đích để dự đoán những sản phẩm chưa được tương tác trước đó. Một cách đơn giản là để huấn luyện mô hình auto-encoder là ta sẽ cố gắng tái tạo lại tương tác của người dùng để dựa vào kết quả tương tác được tái tạo để đưa ra tập sản phẩm "tốt" nhất mà người dùng chưa tương tác trước đó. Hoặc ta có thể xem như là một bài toán hồi quy, khi mà dữ liệu đầu vào là một con số thể hiện cho tương tác của người dùng, và mục tiêu của mô hình chính là kết quả của mô hình cũng là một véc tơ số "gần" với dữ liệu tương tác ban đầu. Để áp dụng hướng tiếp cận cơ bản này, chúng tôi sẽ trình bày về kiến trúc của một mô hình auto-encoder để xây dựng hệ thống gợi ý cơ bản. Mô hình được cấu thành từ hai mạng nơ ron tách biệt được gọi là encoder và decoder. Ta có thể xem rằng mạng nơ-ron encoder là một hàm phi tuyến ánh xạ dữ liệu đầu vào ở chiều không gian cao, $x_u \in R^I$ sang một không gian thấp hơn để biểu diễn cho đặc trưng ẩn $z_u \in R^k$ với k là số chiều của không gian đặc trưng ẩn và thường thì k sẽ rất nhỏ so với I. Cụ thể thì ta có:

$$z_u = f(W * b_1)$$

trong đó $f(.)$ là hàm kích hoạt phi tuyến và W, b_1 tương ứng sẽ là trọng số và hệ số "bias" của encoder.

Sau khi có được đặc trưng ẩn, z_u sẽ được lan truyền thẳng thông qua mạng nơ-ron decoder nhằm mục đích xây dựng lại tương tác ban đầu từ z_u

$$x'_u = g(V \times z_u + b_2)$$

Tương tự với encoder, V, b_2 sẽ tương ứng là các trọng số và "bias" của mạng nơ ron và g(.) chính là hàm kích hoạt của mạng nơ-ron.

Để huấn luyện một mô hình auto-encoder để thực hiện gợi ý sản phẩm ta có thể dùng hàm chi phí sau:

$$\min_\phi \sum_{x_u \in X} (||r - h(x_u, \theta)||_2^2)$$

trong đó $h(x_u, \theta)$ là tương tác được tái tạo lại từ dữ liệu đầu vào cũng chính là tương tác của người dùng $u \in U$ và $x_u \in R^I$.

$$h(x_u, \theta) = f(W \times g(V \times x_u + b_1) + b_2)$$

với f(.) và g(.) sẽ là các hàm kích hoạt của mạng nơ-ron và $\theta = W, V, b_1, b_2$ với W, V là trọng số tương ứng của encoder và decoder, b_1, b_2 tương ứng sẽ là hệ số "bias" của decoder và encoder. Với hàm chi phí này thì mô hình sẽ huấn luyện và tìm ra bộ tham số θ^* sao cho với bộ tham số này, thì mô hình sẽ có thể tự động trích xuất được những đặc trưng ẩn có thể được dùng để hình thành nên các tương tác của người dùng.

Tuy nhiên nhiên, để có thể tính toán cũng như cập nhật trọng số cho mô hình trong tình huống mô hình được sử dụng để phát sinh gợi ý cho người dùng. Cụ thể, đầu tiên, khi chúng ta huấn luyện một mạng nơ-ron phi tuyến thì tình trạng "overfitting" sẽ thường xảy ra, gây ảnh hưởng đến độ chính xác của mô hình. Do đó chúng ta cần phải thêm một lượng "regular" cho bộ tham số θ. Thứ hai, chúng ta cần phải xét đến việc rằng dữ liệu đầu vào của chúng ta là $x_u \in R^I$ là một véc-tơ có dạng bag-of-world. Có nghĩa là chỉ có phần sản phẩm là có tương tác, một người dùng chỉ tương tác một lượng ít sản phẩm chứ không phải toàn bộ. Do đó để

những trương tác trong quá khứ của một người dùng có thể được dùng để thể hiện được đặc trưng ẩn của người đó, chúng ta chỉ cập nhật trọng số cho những nơ-ron mà có kết nối đến những sản phẩm mà người dùng đã tương tác. Vậy nên đê huấn luyện mô hình auto-encoder với bài toán xây dựng hệ thống gợi ý sản phẩm ta sẽ huấn luyện mô hình để tìm được bộ tham số cho mô hình, ta sẽ huấn luyện với hàm mục tiêu sau:

$$\min_\phi \sum_{x_u \in X} (||r - h(x_u, \theta)||^2_{\mathcal{O}}) + \frac{\lambda}{2} \times (||W||^2_2 + ||V||2_2)$$

trong đó $||.||^2_{\mathcal{O}}$ có nghĩa là chúng ta chỉ xét đến những tương tác xảy ra, bỏ qua đóng góp của các nơ-ron liên kết với các sản phẩm không có tương tác.

Với kiến trúc trên, mô hình xây dựng cho hệ thống gợi ý với số lượng sản phẩm là I và đặc trưng ẩn với số chiều là k thì số lượng tham số của mô hình sẽ là $2Ik + I + k$.

Sau khi huấn luyện mô hình với hàm mục tiêu trên, ta tìm được được bô tham số θ^*, thì để dự đoán số điểm tương tác của người dùng u với sản phẩm i cụ thể sẽ là:

$$\widehat{x}_{ui} = h(x_{ui}, \theta^*)$$

Đây chính là kiến trúc cơ bản nhất khi ta sử dụng mô hình auto-encoder để xây dựng một thống gợi ý sản phẩm.

3.2.1 Tăng khả năng phát sinh gợi ý cho người dùng bằng kỹ thuật "drop-out"

Như đã thảo luận ở phần trên, thì chúng ta đã xây dựng một kiến trúc cơ bản nhất trong việc xây dựng một hệ thống gợi ý sản phẩm dựa trên mô hình auto-encoder. Tuy nhiên, mô hình trên vẫn còn nhiều hạn chế sau:

- Mặc dù, để phù hợp với bài toán gợi ý sản phẩm thì chúng ta chỉ

tập nhật trọng số có liên kết với các phần tử là tương tác của người trong tập $\mathcal{O}$, là tập tương tác của người dùng, nhưng thực tế thì mô hình vẫn đang làm tác vụ là tái tạo lại tương tác của người dùng từ đặc trưng ẩn phát sinh từ tương tác của người dùng.

- Một hạn chế khác đó là tính trạng overfiting, khi huấn luyện mạng nơ-ron. Với việc thêm 'regular' để huấn luyện nhưng với dữ liệu thưa thì, regular sẽ không thực sự hiệu quả.

Một phương pháp mà chúng tôi sử dụng để có thể giải quyết những vấn đề trên đó là chúng tôi che đi một số tương tác trước khi dữ liệu được truyền thẳng qua mạng encoder. Cụ thể thì chúng tôi áp dụng dropout cho véc-tở đầu vào, droput là một phương pháp "regularization" thường được áp dụng khi huấn luyện các mạng học sâu. Dropout sẽ ép mạng nơ-ron tìm ra được những đặc trưng quan trọng hơn, hay là tìm ra những sản phẩm ảnh hưởng lớn đến đặc trưng ẩn của người dùng.

Với việc sử dụng thêm nhiễu vào dữ liệu tương tác ban đầu của người dùng sẽ mang lại cho mô hình auto-encoder phù hợp hơn và hiệu quả hơn khi huấn luyện với dữ liệu tương tác của người dùng. Điều đầu tiên mang lại cho mô hình khi sử dụng một tầng "dropout" đó là mô hình lúc này ngoài việc trọng số được cập nhật để có thể tái tạo lại dữ liệu của ngươi dùng mà khi này, trong tương tác của người dùng đã thực sự bị che mất một phần vậy nên cả quá trình huấn luyện mô hình cũng sẽ phải học để có thể dự đoán những trương tác bị che trong dữ liệu. Nói cách khác đó là sau khi thực hiện thêm nhiễu, thì mô hình sẽ thực sự là dự đoán những tương tác bị che từ những tương tác có trước đó trong lịch sử của người dùng thay vì chỉ để tái tạo lại tương tác của người dùng trước đó.

Bên cạnh đó, drop out là một cách hiệu quả để ngăn tình trạng "over-fitting" hiệu quả.

Đây cũng chính kiến trúc của một biến thể của auto-encoder đó là denosing auto-encoder đã được trình bày trong phần 2.1.3.

3.3 Mở rộng mô hình "Variational Auto-encoder" cho bài toán gợi ý sản phẩm

Trong lĩnh vực trí tuệ nhân tạo thì dữ liệu đóng vài trò cực kỳ quan trọng. Đặc biệt là lĩnh vực máy học, thì hiểu được dữ liệu sẽ góp phần không ít đến việc xây dựng một mô hình hiệu quả. Do đó, để có thể tận dụng được tính chất của dữ liệu trong bài toán xây dựng gợi ý sản phẩm đó là thưa, có nghĩa là người dùng trong thực tế sẽ chỉ tương tác với một lượng nhỏ số lượng sản phẩm. Với đặc điểm này, cũng chính là lý do mà tác giả trong bài báo mà chúng tôi tìm hiểu đã đề xuất việc sử dụng mô hình variational auto-encoder, một biến thể đặc biệt của auto-encoder cơ bản để xây dựng hệ thống gợi ý sản phẩm.

Ở phần này chúng tôi sẽ trình bày về kiến trúc mô hình đã tìm hiểu cho bài toán gợi ý sản phẩm trong bài báo cũng như là những đề xuất cải thiện cho mô hình VAEs trong tác vụ gợi ý sản phẩm cho người dùng.

3.3.1 Mô hình variational autoencoder cho bài toán gợi ý sản phẩm

Dữ liệu lương tác trong hệ thống gợi ý sản phẩm thường là dữ liệu thưa, có nghĩa là các phần tử trong véc-tơ input đầu vào sẽ đa phần sẽ mang giá trị 0. Ngoài ra, mục tiêu của một hệ thống gợi ý sản phẩm chính là việc tăng thêm số lượng tương tác của người dùng lên hệ thống, đặc biệt là khi người dùng chưa tương tác nhiều. Do đó, trong trường hợp này, nếu áp dụng những phương pháp dựa trên mạng nơ-ron thông thường sẽ dễ dẫn đến tình trạng "overfiting" khi mà dữ liệu đầu vào ít nên mô hình không học được.

Dựa vào hạn chế này, bài báo đã đề xuất việc sử dụng mô hình variation auto-encoder. Tuy là một mô hình mạng nơ-ron nhưng với nền tảng xác suất, cụ thể là phương pháp variational inference, là một phương pháp suy diễn dữ liệu trong lĩnh vực xác suất thống kê. Như đã trình bày ở phần

2.2, đây là một biến thể đặc biệt của auto-encoder cơ bản, với việc đặc trưng ẩn được rút trích là một phân phôi xác suất, mô hình đã mang lại ý nghĩa xác suất cho đặc trưng ẩn. Nói cách khác, đặc trưng giờ đây k chỉ thể hiện cho tương tác của người dùng, ngoài ra, là một phân phối xác suất thì đặc trưng ẩn sẽ thể hiện được nhiều điều hơn về tương tác của người dùng.

Nếu ta sử dụng tất cả tương tác của người dùng để trích xuất được những đặc trưng ẩn, thì có thể sẽ bị ảnh hưởng bởi nhiễu, hay có thể sẽ có những sản phẩm "không quan trọng" trong việc thê hiện đặc trưng của người dùng nên đặc trưng ẩn sẽ mang theo tất cả các thông tin đó, điều này có thể dẫn đến việc overfiting khi dữ liệu thưa, hay tương tác của người dùng còn ít. Những sản phẩm "không quan trọng" có thể xen là những sản phẩm mà không xuất hiện theo mẫu tương tác chung giữa các người dùng, những sản phẩm có thể sẽ là những sản phẩm đặc biệt của những người dùng có sở thích "khác biệt" so với những người dùng còn lại. Vì theo ý tưởng của collaborative, là hướng tiếp cận mà nhóm tìm hiểu để xây dựng hệ thống gợi ý, thì việc đưa ra gợi ý cho một người dùng sẽ dựa trên việc mô hình tìm ra được những mẫu tương tác chung giữa các người dùng và mẫu tương tác chung này sẽ được thể hiện thông qua đặc trưng ẩn. Ngược lại, nếu từ tương tác của người, ta phát sinh một phân bố để thể hiện cho đặc trưng ẩn, thì khi sử dụng đặc trưng ẩn để đưa ra gợi ý, thì điểm "tốt nhất" trong phân bố, cụ thể ở bài báo được đề xuất là giá trị trung bình của phân bố sẽ được lấy để đưa ra gợi ý. Vì là một phân bố, những sản phẩm "ko quan trọng" sẽ ít đóng góp ít hơn ở điểm trung bình. Vì điểm trung bình sẽ là điểm có phân phối xác suất cao nhất, có nghĩa là tập sản phẩm ít xuất hiện trong xu hướng tương tác chung của các người dùng trong hệ thống sẽ đóng góp ít hơn ở điểm trung bình.

3.3.2 Thay đổi hàm lỗi để phù hợp hơn cho bài toán gợi ý sản phẩm

Multinomial likelihood

Beta variational auto-encoder

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] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. "Are we really making much progress? A worrying analysis of recent neural recommendation approaches". In: *Proceedings of the 13th ACM Conference on Recommender Systems*. 2019, pp. 101–109.

[2] 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.

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

[3] *Movie Recommendations : How Netflix does it? | Data Science Workshop | Ivy Pro School*. URL: `https://www.youtube.com/watch?v=7lPOL74eG5I` (visited on 06/17/2021).

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

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

[7] 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.