**xDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems**

**TÓM TẮT**

Trong lĩnh vực hệ thống gợi ý (hệ khuyến nghị), việc kết hợp tương tác giữa các đặc trưng mà ta có thể nhìn thấy (rõ ràng) và những tương tác ẩn (không rõ ràng) đang là một vấn đề thách thức. Trong bài báo này, tác giả giới thiệu mô hình xDeepFM, một phương pháp mới kết hợp cả hai loại tương tác này, nhằm cải thiện hiệu suất của hệ thống gợi ý. Mô hình xDeepFM kết hợp hai kiến trúc chính là mạng DNN và mạng CIN. DNN được sử dụng để học các tương tác đặc trưng ngầm, trong khi CIN được sử dụng để mô hình hóa tương tác đặc trưng rõ ràng. Mô hình xDeepFM được đánh giá và so sánh với các mô hình gợi ý khác trên các bộ dữ liệu gợi ý phổ biến là Criteo, Dianping và Bing News. Kết quả thực nghiệm cho thấy xDeepFM vượt trội với hiệu suất tốt hơn so với các mô hình FM, Wide & Deep và DeepFM truyền thống. Điều này chỉ ra rằng mô hình xDeepFM là một phương pháp tiềm năng trong việc kết hợp cả tương tác đặc trưng rõ ràng và những tương tác đặc trưng ẩn để cải thiện hiệu suất của hệ thống gợi ý trong các ứng dụng thực tế.

**Từ khoá:** Factorization Machines, Neural Network, recommender systems, deep learning, feature interactions

**1 GIỚI THIỆU**

Trong thời đại số hóa hiện nay, hệ thống gợi ý đã trở thành một công cụ quan trọng để giúp người dùng khám phá và tiếp cận thông tin, sản phẩm và nội dung phù hợp với sở thích cá nhân. Các hệ thống gợi ý dựa trên các thuật toán phức tạp để phân tích dữ liệu người dùng và đề xuất các mục tiêu phù hợp nhằm cung cấp trải nghiệm cá nhân hóa và tăng cường sự tương tác người dùng.

Một trong những thách thức quan trọng của hệ thống gợi ý là kết hợp cả tương tác đặc trưng rõ ràng và tương tác ngầm định từ dữ liệu để tạo ra các đề xuất chất lượng và mang tính cá nhân cao. Tương tác đặc trưng rõ ràng là các tương tác dựa trên thông tin đã biết và được khai thác một cách rõ ràng, trong khi tương tác đặc trưng ngầm định là các tương tác ẩn chứa trong dữ liệu mà chúng ta cần tìm hiểu và mô hình hóa.

Có 3 vấn đề chính trong việc chọn các tương tác đặc trưng rõ ràng. Đầu tiên, các đặc trưng tốt luôn đi kèm với chi phí cao. Bởi vì để lựa chọn đúng các đặc trưng tương tác thì yêu cầu phải dành nhiều thời gian làm rõ và tìm ra khuôn mẫu của các đặc trưng. Thứ hai, dữ liệu trên trang web khá lớn do đó việc trích xuất thủ công các tương tác là bất khả thi. Cuối cùng, các tương tác đặc trưng thủ công không thể khái quát hóa được các tương tác chưa nhìn thấy trong dữ liệu. Vì vậy, việc trích xuất tự động các tương tác đặc trưng là điều cần thiết.

Factorization Machine (FM) sẽ ‘nhúng’ các đặc trưng vào một vector ngầm định và sự tương tác giữa các đặc trưng sẽ được biểu diễn dưới dạng tích 2 vector **.** Trong bài báo này, tác giả sử dụng từ ‘bit’ để biểu diễn cho 1 phần tử trong vector ngầm định **.** Mô hình FM truyền thống có thể mở rộng tới bất kì bậc tương tác đặc trưng nào, nhưng có một nhược điểm là không thể chọn sự tương tác cố định, vì vậy sự tương tác này có thể làm giảm hiệu năng của mô hình.

Trong những năm gần đây, mạng DNNs đã thành công ở các lĩnh vực như thị giác máy tính, xử lí ngôn ngữ tự nhiên bởi vì khả năng biểu diễn các đặc trưng của mạng. Do đó, các nhà khoa học đã đề xuất mô hình Factorisation-machine supported Neural Network (FNN) để học sự tương tác giữa các đặc trưng bậc cao, mô hình này sử dụng 1 lớp Factorization machine đã được huấn luyện sẵn trước khi sử dụng mạng DNN. Tiếp đó, mô hình Product-based Neural Network (PNN) giới thiệu 1 lớp Product đặt giữa lớp embedding và lớp DNN, mô hình này không dựa vào lớp FM được huấn luyện sẵn. Tuy nhiên vẫn tồn tại nhược điểm của FNN và PNN là 2 mô hình này tập trung quá nhiều vào sự tương tác giữa các đặc trưng cấp cao và bỏ qua sự tương tác các đặc trưng cấp thấp. Mô hình Wide&Deep và DeepFM đã giải quyết các vấn đề trên bằng việc đề xuất kiến trúc hỗn hợp bao gồm các mạng nông (shallow layer) và mạng sâu (deep layer), với mục tiêu là học được sự tổng quát hóa trong khi vẫn hoạt động tốt với các đặc trưng cũ.

Các mô hình được đề cập ở trên đã tận dụng mạng DNNs để học sự tương tác giữa các đặc trưng. Tuy nhiên, mạng DNNs mô hình hóa ngầm định các đặc trưng bậc cao, số bậc cuối cùng được học bởi mạng là bất kì. Thêm vào đó, mạng DNNs biểu diễn sự tương tác các đặc trưng ở mức bit-wise trong khi mạng FM truyền thống biểu diễn ở mức vector-wise. Vì vậy, trong lĩnh vực hệ khuyến nghị, liệu rằng mạng DNNs có thực sự biểu diễn được sự tương tác giữa các đặc trưng không vẫn là câu hỏi mở.

Trong bài báo này, tác giả đề xuất mạng Neural Network có thể học rõ ràng sự tương tác giữa các đặc trưng ở mức vector-wise. Hướng tiếp cận của tác giả dựa vào mạng Deep&Cross (DCN) với mục tiêu là nắm được số bậc cụ thể của sự tương tác. Thêm vào đó, tác giả đề xuất mạng Compressed Interaction Network (CIN) để thay thế lớp Cross Network trong mạng DCN. Mạng CIN có thể học được sự tương tác rõ ràng giữa các đặc trưng và số bậc của tương tác tăng dần theo độ sâu của mạng. Dựa vào mạng Wide&Deep và DeepFM, tác giả kết hợp mạng CIN, mạng DNN và mạng FM truyền thống thành một mô hình thống nhất và đặt tên là eXtreme Deep Factorization Machine (xDeepFM).

Tóm lại, trong bài báo này tác giả thực hiện:

* Đề xuất một mô hình mới eXtreme Deep Factorization Machine (xDeepFM) có thể học sự tương tác rõ ràng và ngầm định giữa các đặc trưng hiệu quả và không yêu cầu lựa chọn đặc trưng trước.
* Thiết kế mạng Compressed Interaction Network (CIN) để biễu diễn sự tương tác rõ ràng giữa các đặc trưng. Tác giả thấy rằng số bậc của đặc trưng tăng dần theo độ sâu của mạng và sự tương tác giữa các đặc trưng ở mức vector-wise.
* Tác giả tiến hành các thử nghiệm trên ba bộ dữ liệu trong thế giới thực và kết quả chứng minh rằng xDeepFM của tác giả có kết quả tốt hơn so với một số mô hình trước đó.

**2 HƯỚNG TIẾP CẬN**

**2.1 Factorization Machine**

Factorization Machine (FM) [Rendle, 2010], được đề xuất bởi Steffen Rendle năm 2010, là một thuật toán được giám sát có thể được sử dụng để phân loại, hồi quy, và xếp hạng nhiệm vụ. Nó nhanh chóng chú ý và trở thành một phương pháp phổ biến và có tác động để đưa ra dự đoán và khuyến nghị.

Factorization Machine (FM) là một phương pháp kết hợp giữa hồi quy tuyến tính và matrix factorization (MF). Ý tưởng đằng sau phương pháp này là nhằm mục đích mô hình hóa các tương tác giữa các feature. Bằng cách này, nó có khả năng ước tính tất cả các tương tác giữa các feature ngay cả với dữ liệu cực kì thưa thớt.

Với hồi quy tuyến tính ta thường để cập đến công thức sau:

Trong đó:

* là bias term (intercept)
* là trọng số tương ứng với từng feature vector , ở đây giả sử chúng ta có tổng cộng n feature.

Ưu điểm của công thức này là nó có thể tính toán trong thời gian tuyến tính. Tuy nhiên, nhược điểm là nó không thể xử lý được tương tác giữa các feature. Để giải quyết vấn đề này, *Factorization Machine* lấy cảm hứng từ *matrix factorization* và mô hình hóa tương tác giữa các feature bằng cách sử dụng *latent factors.* Mỗi feature *fi* đều có latent factor *vi* tương ứng và tương tác giữa hai feature được mô hình hóa thành . Giả sử chúng ta có kích thước của *latent factor* k, ta có:

Khi đó công thức mới của chúng ta sẽ là:

**2.2 Wide & Deep Learning**

**2.2.1. Giới thiệu**

Một thách thức trong recommender system, tương tự như vấn đề xếp hạng tìm kiếm chung, là đạt được cả *memorization* và *generalization*. *Memorization* có thể được định nghĩa đơn giản là học sự xuất hiện thường xuyên của các item hoặc feature và khai thác mối tương quan có sẵn trong dữ liệu lịch sử. Mặt khác, *generalization* dựa trên tính bắc cầu của mối tương quan và khám phá các kết hợp feature mới chưa bao giờ hoặc hiếm khi xảy ra trong quá khứ. Các đề xuất dựa trên *memorization* thường mang tính thời sự hơn và liên quan trực tiếp đến các mục mà người dùng đã thực hiện hành động. So với *memorization*, *generalization* có xu hướng cải thiện sự đa dạng của các mục được đề xuất.

Wide & Deep Learning framework đạt được cả *memorization* và *generalization* trong một mô hình bằng cách cùng huấn luyện hai thành phần là linear model và neural network.

A diagram of a network

Description automatically generated with low confidence

*Hình 1: Kiến trúc mô hình Wide&Deep*

**2.2.2. The Wide Component**

The wide component là một mô hình tuyến tính tổng quát có dạng:

Trong đó:

* là kết quả dự đoán
* x là vector của *d* features
* w là tham số model
* là bias

Bộ feature bao gồm các feature đầu vào thô và các feature được chuyển đổi. Một trong những phép chuyển đổi quan trọng nhất là cross-product transformation, được định nghĩa như sau:

Trong đó là biến boolean mang giá trị 1 khi feature thứ i là một phần của phép chuyển đổi thứ *k,* nếu không thì mang giá trị 0. Với binary feature, một phép cross-product transformation (e.g., “AND(gender=female, language=en)”) có giá trị là 1 khi và chỉ khi các feature thành phần (“gender=female” và “language=en”) đều là 1, nếu không là 0. Điều này nắm bắt các tương tác giữa các binary feature và thêm tính phi tuyến vào mô hình tuyến tính tổng quát.

**2.2.3. The Deep Component**

The deep component là một feed-forward neural network. Đối với categorical features, input gốc là chuỗi feature (e.g, “language=en”). Những features thưa thớt, high-dimensional categorical features sẽ được chuyển sang một dạng khác thường được gọi là embedding vector. Kích thước của các embedding vector thường theo thứ tự từ O(10) đến O(100). Các embedding vector được khởi tạo ngẫu nhiên và sau đó giá trị sẽ được huấn luyện để giảm thiểu loss function cuối cùng trong suốt quá trình huấn luyện mô hình. Các embedding vector này sau đó được cho vào các hidden layer của một neural network trong forard pass. Cụ thể, mỗi hidden layer thực hiện phép tính như sau:

Trong đó:

* là số layer
* là activation function
* là activation
* là bias
* là model weights tại layer thứ
  + 1. **Joint Training of Wide & Deep Model**

Joint training of a Wide & Deep Model được thực hiện bằng các lan truyền ngược gradients từ output đến cả hai phần wide và deep của model đồng thời sử dụng mini-batch để tối ưu hóa.

Mô hình kết hợp được minh họa trong Hình 1 (giữa). Đối với vấn đề logistic regression, dự đoán của mô hình là:

Trong đó:

* Y là nhãn nhị phân
* là hàm sigmoid
* là cross product transformation của features gốc x
* là bias term
* là vector tất cả trọng số của wide model
* là trọng số ở final activation

**2.3 DEEPFM - COMBINING FM AND NEURAL NETS**

**2.3.1. Giới thiệu**

So với Google “Wide & Deep Learning” framework, là sự kết hợp giữa linear regression và deep neural network thì DeepFM là sự kết hợp giữa factorization machine và deep neural network, nhưng hai phần của model dùng chung inputs, khác với raw features cho deep neural network and engineered features cho linear regression ở Wide & Deep Learning framework của Google.

Mục đích của DeepFM là kết hợp sức mạnh của factorization machine và deep neural network, DeepFM có những ý chính sau:

* DeepFM là sự tích hợp của factorization machine và deep neural network, có thể được train từ đầu đến cuối mà không cần bất kì kĩ thuật xử lý feature nào.
* DeepFM có thể được train một cách hiệu quả vì phần wide và phần deep của nó chia sẻ cùng một đầu vào và cả embedding vector, với lợi ích là học các tương tác feature bậc thấp và bậc cao từ các feature thô và không cần kỹ thuật xử lý feature.

**2.3.2. DeepFM**

Đối với feature i, được sử dụng để cân nhắc tầm quan trọng bậc 1 của nó, latent vector được sử dụng để cân nhắc tác động tương tác của nó với các feature khác, được cung cấp trong phần FM để mô hình hóa các tương tác feature bậc 2, và được cung cấp trong phần deep để mô hình hóa các tương tác feature bậc cao. Tất cả các tham số trong FM (, ) và deep neural netwwork (W(l),b(l)) được train chung cho model:

Trong đó:

* là giá trị dự đoán CTR
* là output của FM component
* là output của deep component

**2.3.3. FM Component**

**A picture containing line, text, plot, diagram

Description automatically generated**

*Hình 2: Kiến trúc mô hình DeepFM*

Như đã giải thích ở phần Factorization machine, ta có:

**2.3.4. Deep Component**

The deep component là feed-forward neural network được sử dụng để tìm hiểu các tương tác feature bậc cao. Tương tự như deep component của Wide & Deep.

A picture containing line, diagram, text, plot

Description automatically generated

*Hình 3: Kiến trúc Deep component*

**3 MÔ HÌNH**

**3.1. Kiến thức sơ bộ**

**3.1.1. Embedding**

Trong lĩnh vực thị giác máy tính hoặc xử lí ngôn ngữ tự nhiên, input thường là dữ liệu ảnh hoặc các từ vựng. Các dạng dữ liệu này có tương quan về mặt không gian và thời gian, do đó mạng DNN có thể được áp dụng trực tiếp lên dạng dữ liệu này.

Tuy nhiên trong lĩnh vực Hệ khuyến nghị, các đặc trưng đầu vào có các đặc điểm sau:

* Rất thưa (Trong bài toán gợi ý sản phẩm, có nhiều sản phẩm mà người dùng chưa từng dùng, do đó các sản phẩm này sẽ có giá trị 0).
* Số chiều rất lớn (Một người dùng có rất nhiều thông tin liên quan).
* Không có tương quan về mặt không gian và thời gian.

Do đó, phương pháp **multi-field categorical** thường được sử dụng rộng rãi. Ví dụ, input [user\_id=s02,gender=male, organization=msra, interests=comedy&rock] sẽ được biến đổi sang số chiều rộng hơn thông qua việc sử dụng **one-hot encoding:**

A picture containing text, font, line, white

Description automatically generated

Sau đó 1 lớp Embedding sẽ được áp dụng lên các vector one-hot này để giảm chiều dữ liệu. Hình ảnh minh họa về lớp Embedding được thể hiện ở hình {x}. Kết quả của lớp Embedding là sự kết hợp của các vector embedding:

Trong đó:

* M: số lượng đặc trưng input
* : vector embedding của 1 đặc trưng
* : số chiều của vector embedding

Dù cho 1 đặc trưng có nhiều giá trị-nhiều chiều, sau khi qua lớp Embedding, kết quả bị nén lại thành số chiều cố định là .

A picture containing text, diagram, screenshot, line

Description automatically generated

*Hình 4: Lớp Embedding. Số chiều trong ví dụ này là 4*

**3.1.2. Implicit Interaction**

Các mô hình trước đó như FNN, Deep Crossing, Wide&Deep đã sử dụng mạng Neural Network lên lớp Embedding để học được sự tương tác giữa các đặc trưng bậc cao. Quá trình thực hiện như sau:

(1) (2)

Trong đó:

* : chiều sâu của mạng
* : hàm kích hoạt
* : output tại layer k

Kiến trúc trực quan của công thức trên khá giống với hình 5, nhưng không có sự xuất hiện của lớp FM hoặc lớp Product. Kiến trúc này mô hình hóa sự tương tác của các đặc trưng ở mức độ bit-wise, có nghĩa là các phần tử trong cùng 1 vector embedding có thể tương tác lẫn nhau.

A diagram of a network

Description automatically generated with low confidence

*Hình 5: Kiến trúc mô hình DeepFM*

PNN và DeepFM đã chỉnh sửa kiến trúc trên. Trước khi áp dụng DNNs lên vector embedding , 2 mô hình này đã thêm 1 lớp để xem xét sự tương tác giữa các đặc trưng bậc 2. Do đó, mô hình này bao gồm sự tương tác giữa các đặc trưng ở mức độ bit-wise và vector-wise. Sự khác biệt lớn nhất giữa PNN và DeepFM là PNN kết nối output tại Product Layer tới DNN, trong khi đó DeepFM kết nối trực tiếp FM Layer tới output như ở hình 5.

**3.1.3. Explicit Interaction**

Ruoxi Wang và các cộng sự đã đề xuất mạng Cross Network (CrossNet), kiến trúc mô hình được thể hiện ở hình 6, với mục tiêu là mô hình hóa sự tương tác giữa các đặc trưng một cách rõ ràng. Không giống như mạng DNN truyền thống, lớp ẩn của mạng CrossNet được tính toán như sau:

Trong đó:

* Trọng số của mô hình
* bias
* output tại layer k

Mạng CrossNet có thể học hiệu quả sự tương tác giữa các đặc trưng (độ phức tạp không đáng kể khi so với DNN). Tuy nhiên mạng CrossNet có 2 nhược điểm sau:

* Output của CrossNet bị giới hạn ở 1 hình thái cố định-mỗi lớp ẩn là 1 phép nhân tuyến tính với
* Sự tương tác giữa các đặc trưng ở mức độ bit-wise

A picture containing text, diagram, line, screenshot

Description automatically generated

*Hình 6: Kiến trúc mô hình CrossNet*

**3.2. Mạng CIN**

Tác giả đề xuất mạng Compressed Interaction Network (CIN) để giải quyết vấn đề của mạng CrossNet với các ưu điểm:

* Sự tương tác giữa các đặc trưng ở mức độ vector-wise thay vì bit-wise.
* Các đặc trưng bậc cao có thể được biểu diễn rõ ràng.
* Độ phức tạp của mạng không tăng theo cấp số nhân theo số bậc của các tương tác đặc trưng.

Bởi vì vector embedding có thể được coi là tương tác ở mức độ vector-wise. Do đó, tác giả biểu diễn output của lớp embedding các đặc trưng ban đầu là một ma trận , với dòng thứ i trong là một vector embedding của một đặc trưng: , trong đó D là số chiều của vector embedding.

Kết quả đầu ra tại layer thứ k của mạng CIN là một ma trận , trong đó là số lượng vector embedding tại layer k và đặt số lượng vector embedding tại layer đầu tiên là

Tại mỗi layer k, được tính dựa trên công thức:

(6)

Trong đó:

* : ma trận trọng số của đặc trưng h
* : Hadamard product

Có thể thấy rằng được tính dựa trên sự ảnh hưởng giữa output tại layer và layer , do đó sự tương tác giữa các đặc trưng này là rõ ràng và số bậc của của ảnh hưởng tăng theo chiều sâu của mạng. Kiến trúc mạng CIN khá giống với mạng RNN, output của layer tiếp theo dựa vào output tại layer trước đó và input hiện tại.

A picture containing diagram, plan, text, technical drawing

Description automatically generated

*Hình 7: Các thành phần và kiến trúc mạng CIN*

Có thể thấy biểu thức (6) khá giống với mạng Convolutional Neural Network (CNN) trong lĩnh vực thị giác máy tính. Như trong hình 7a, tác giả giới thiệu một tensor được tính dựa trên phép outer product giữa layer k và layer đầu tiên dọc theo chiều embedding D. Sau đó, giống với mạng CNN, tác giả coi là một bức ảnh và là một filter. Tiếp theo tác giả sẽ trượt filter dọc theo chiều embedding D như trong hình 7b. Kết quả thu được là 1 vector ẩn , thường được gọi là feature map trong thị giác máy tính và là sự tổng hợp của nhiều feature map. Từ Compressed được tác giả đặt trong mạng CIN có nghĩa là “nén”, vector ban đầu có số chiều là ( sau đó được nén xuống số chiều là .

Kiến trúc tổng quan của mạng CIN được thể hiện ở hình 7c. Output tại các lớp ẩn của mạng CIN đều kết nối đến output cuối cùng. Đầu tiên, tác giả sử dụng sum pooling cho từng feature map đối với từng lớp ẩn:

Do đó, tác giả có 1 vector để tổng hợp các sum pooling tại layer k là , với là số lượng feature map tại layer k.

Các vector pooling tại các layer k sẽ được nối lại với nhau trước khi qua lớp output cuối cùng

Bài toán CTR được tác giả sử dụng là bài toán phân loại nhị phân, vì vậy sẽ đi qua lớp sigmoid và đưa ra kết quả dự đoán:

Trong đó: là trọng số mô hình

**3.3. Mạng xDeepFM**

Như đã đề cập trong mục 3.1.2, mạng DNN truyền thống có thể học sự tương tác giữa các đặc trưng cấp cao một cách ngầm định. Vì mạng CIN và DNN có thể bổ sung cho nhau, nên một cách trực quan để làm cho mô hình mạnh hơn là kết hợp 2 mạng này lại với nhau.

Mô hình này khá giống với Wide&Deep và DeepFM. Kiến trúc mô hình được mô tả ở hình 8. Tác giả đặt tên cho mô hình này là eXtreme Deep Factorization Machine (xDeepFM). Mô hình này có các ưu điểm sau:

* Học được sự tương tác giữa các đặc trưng cấp thấp và cấp cao (kết nối trực tiếp tới output và qua Deep Neural Network)
* Bao gồm sự tương tác các đặc trưng rõ ràng và ngầm định (CIN và DNN)

Kết quả đầu ra của mô hình là:

Trong đó:

* : hàm sigmoid
* a: đặc trưng ban đầu
* : output của mạng CIN
* : trọng số của mô hình

Trong tác vụ CTR, loss function sử dụng binary cross entropy:

Trong đó:

* N: số mẫu dữ liệu huấn luyện

Quá trình huấn luyện là quá trình cực tiểu hóa hàm mất mát sau:

Trong đó:

* : tham số điều chuẩn
* : các tham số huấn luyện

A picture containing diagram, text, line, screenshot

Description automatically generated

*Hình 8: Kiến trúc mạng xDeepFM*

**4 BỘ DỮ LIỆU**

**4.1. Criteo dataset**

### **4.1.1. Tổng quan**

Đây là bộ dữ liệu khá nổi tiếng, được sử dụng để phát triển và đánh giá các hệ thống dự đoán CTR. Đây là bài toán phân loại nhị phân với 2 nhãn là Clicked (1) và Non-clicked (0).

Bộ dữ liệu chứa các thông tin truy cập của Criteo. Mỗi dòng là một quảng cáo hiển thị được Criteo phân phối tới người dùng. Tổng quan về bộ dữ liệu được thể hiện ở hình 9. Mỗi dòng trong bộ dữ liệu được sắp xếp theo thời gian. Các đặc trưng của người dùng bị ẩn để đảm bảo quyền riêng tư của người dùng.

A screenshot of a computer

Description automatically generated with low confidence

*Hình 9: Criteo Sample Dataset*

### **4.1.2. Các đặc trưng trong bộ dữ liệu**

Bộ dữ liệu Criteo được nhóm sử dụng có 100.000 dòng và chứa các thuộc tính sau:

| Thuộc tính | Loại thuộc tính |
| --- | --- |
| Label | Binary |
| I1-I13 | Numeric |
| C1-C26 | Nominal |

*Bảng 1: Các thuộc tính trong bộ dữ liệu Criteo*

### **4.1.3 Khám phá bộ dữ liệu**

Phân phối nhãn của bộ dữ liệu Criteo được thể hiện ở hình 10. Đây là bộ dữ liệu có phân phối nhãn không đều nhau, đa số các nhãn sẽ là 0.

A picture containing text, screenshot, font, diagram

Description automatically generated

*Hình 10: Phân phối nhãn của Criteo*

**4.2. Dianping dataset**

### **4.2.1. Tổng quan**

Dianping.com là trang web đánh giá dành cho người tiêu dùng lớn nhất tại Trung Quốc. Trang web này cung cấp các tính năng đa dạng như đánh giá, check-in, các thông tin về các cửa hàng (bao gồm cả thông tin địa lí và thuộc tính cửa hàng).

Tác giả thu nhập hoạt động check-in của người dùng trong 6 tháng cho hệ thống gợi ý cửa hàng. Tổng quan về bộ dữ liệu được thể hiện ở hình 11 và hình 12. Với thuộc tính người dùng, các thuộc tính nhà hàng và 3 địa điểm người dùng ghé qua gần nhất, mục tiêu của bài toán là dự đoán xác suất người dùng sẽ ghé thăm một nhà hàng cụ thể.

Với mỗi nhà hàng trong hoạt động check-in của người dùng, tác giả sẽ gán 4 nhà hàng trong vòng 3km là mẫu 0 (negative) và nhà hàng được người dùng check-in sẽ là mẫu 1 (positive)

A picture containing text, screenshot, number, parallel

Description automatically generated

*Hình 11: Đánh giá của người dùng*

A screenshot of a computer

Description automatically generated with low confidence

*Hình 12: Thuộc tính của nhà hàng*

### **4.2.2. Các đặc trưng trong bộ dữ liệu**

Bộ dữ liệu Dianping có các thuộc tính sau:

| Thuộc tính | Loại thuộc tính | Ý nghĩa |
| --- | --- | --- |
| Item\_id | Nominal | Mã id cửa hàng |
| User\_id | Nominal | Mã id người dùng |
| User\_rating | Ordinal | Đánh giá người dùng với cửa hàng |
| User\_timestamp | Interval-scaled | Thời gian người dùng đến cửa hàng |
| User\_flavor | Ordinal | Đánh giá người dùng với hương vị |
| User\_environment | Ordinal | Đánh giá người dùng với không gian |
| User\_service | Ordinal | Đánh giá người dùng với dịch vụ |
| User\_cost | Ratio-scaled | Giá tiền sau khi dùng tại cửa hàng |
| User\_stage | Nominal | Thời gian dùng bữa tại cửa hàng (sáng, chiều, tối) |
| User\_waiting | Ratio-scaled | Thời gian đợi |
| Res\_item\_name | Nominal | Tên cửa hàng |
| Res\_city\_name | Nominal | Tên thành phố của cửa hàng |
| Res\_address | Nominal | Địa chỉ cửa hàng |
| Res\_style | Nominal | Phong cách cửa hàng |
| Res\_areas | Nominal | Khu vực của cửa hàng |
| Res\_tel | Nominal | Số điện thoại của cửa hàng |
| Res\_shophour | Nominal | Thời gian hoạt động của cửa hàng |
| Res\_score | Ordinal | Điểm số đánh giá chung của nhà hàng |
| Res\_flavor | Ordinal | Điểm số đánh giá chung về hương vị của nhà hàng |
| Res\_environment | Ordinal | Điểm số đánh giá chung về không gian của nhà hàng |
| Res\_service | Ordinal | Điểm số đánh giá chung về dịch vụ của nhà hàng |
| Res\_share | Ratio-scaled | số lượng người dùng trang web đã nhấp vào nút 'chia sẻ với bạn bè của tôi' cho doanh nghiệp |
| Res\_cost | Ratio-scaled | Giá trung bình của cửa hàng |
| Res\_page | Nominal | Mã nguồn trang website của cửa hàng |
| Res\_longitude | Interval-scaled | Kinh độ của cửa hàng |
| Res\_latitude | Interval-scaled | Vĩ độ của cửa hàng |
| Label | Binary | Nhãn dữ liệu (1: check-in, 0: không check-in) |

*Bảng 2: Các thuộc tính trong bộ dữ liệu Dianping*

### **4.2.3. Khám phá bộ dữ liệu**

Phân phối nhãn của bộ dữ liệu Dianping được thể hiện ở hình 13. Đây là bộ dữ liệu có phân phối nhãn không đều nhau, đa số các nhãn sẽ là 0.

A picture containing text, screenshot, rectangle, font

Description automatically generated

Hình 13: Phân phối nhãn của bộ dữ liệu Dianping

Điểm số đánh giá chung của nhà hàng được thể hiện ở hình 14. Dựa vào biểu đồ có thể thấy đa số người dùng không đánh giá, và điểm đánh giá được nhiều người chọn là 3.5.

A picture containing text, screenshot, rectangle, line

Description automatically generated

*Hình 14: Điểm số đánh giá chung của nhà hàng*

Điểm số đánh giá dịch vụ của nhà hàng được thể hiện ở hình 15, điểm số này được tính toán bởi trang web Dianping và sẽ hiển thị trên trang đầu tiên của cửa hàng. Dựa vào biểu đồ có thể thấy đa số người dùng không đánh giá, và điểm đánh giá nhiều nhất là 15.

A picture containing text, screenshot, diagram, software

Description automatically generated

*Hình 15: Điểm số đánh giá dịch vụ của nhà hàng*

**4.3. Bing News dataset**

### **4.3.1. Tổng quan**

Bing News là một phần của công cụ tìm kiếm Bing của Microsoft. Bộ dữ liệu được thu nhập 5 ngày liên tiếp dựa trên nhật kí hiển thị của trang tin tức. Tổng quan về bộ dữ liệu được thể hiện ở hình 16.

Với thông tin tiêu đề của tin tức và đối tượng được đề cập (entity\_info), mục tiêu là dự đoán khả năng người dùng click vào tin tức đó.

A picture containing text, screenshot, font, line

Description automatically generated

*Hình 16: Bộ dữ liệu Bing News*

### **4.3.2. Các đặc trưng trong bộ dữ liệu**

Bộ dữ liệu Bing News chứa các thuộc tính sau:

| Thuộc tính | Loại thuộc tính | Ý nghĩa |
| --- | --- | --- |
| News\_title | Chuỗi kí tự | Tiêu đề của tin tức |
| Entity\_info | Chuỗi kí tự | Đối tượng được đề cập |
| Label | Binary | Nhãn dữ liệu (1: clicked, 0: non-clicked) |

*Bảng 3: Các thuộc tính trong bộ dữ liệu Bing News*

### **4.3.3. Khám phá bộ dữ liệu**

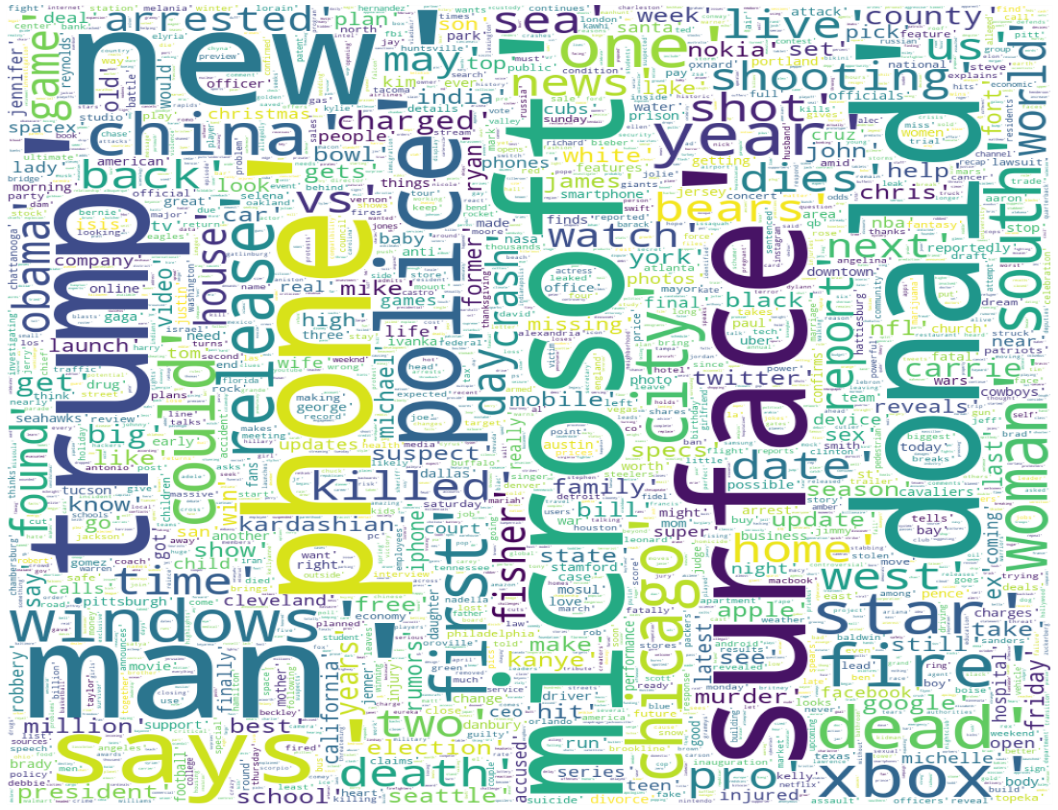
Phân phối nhãn của bộ dữ liệu Bing News được thể hiện ở hình 17. Đây là bộ dữ liệu có phân phối nhãn không đều nhau, đa số các nhãn sẽ là 1.

A picture containing text, screenshot, rectangle, diagram

Description automatically generated

*Hình 17: Phân phối nhãn của bộ dữ liệu Bing News*

Tần xuất xuất hiện của các từ với tiêu đề được người dùng click vào được thể hiện ở hình 18. Có thể thấy các từ được xuất hiện nhiều nhất là: microsoft, donald trump, says, news, surface, ...



*Hình 18: Tần suất xuất hiện các từ người dùng click vào*

**5 KẾT QUẢ THỰC NGHIỆM**

**5.1. Phân chia bộ dữ liệu**

Kích thước bộ dữ liệu được tác giả sử dụng rất lớn, do đó nhóm chỉ thực hiện trên một mẫu nhỏ dữ liệu. Số mẫu dữ liệu được nhóm sử dụng được thể hiện ở bảng 4.

Trong bài báo của tác giả, bộ dữ liệu Criteo và Dianping được chia thành 80% train, 10% validation và 10% test. Đối với bộ dữ liệu Bing News, tác giả thu nhập dữ liệu 5 ngày liên tiếp. Tác giả lấy dữ liệu 3 ngày đầu để train và validation, 2 ngày sau được sử dụng để test.

| Bộ dữ liệu | Paper | Ours |
| --- | --- | --- |
| Criteo | 45M | 0.1M |
| Dianping | 1.2M | 0.2M |
| Bing News | 5M | 0.01M |

*Bảng 4: Bảng so sánh số mẫu dữ liệu được nhóm sử dụng*

**5.2. Độ đo đánh giá**

### **5.2.1 AUC**

**ROC Curve:** là một đồ thị cho biết hiệu năng của mô hình phân loại tại từng ngưỡng phân loại. Để vẽ đường cong ROC (ROC Curve) cần xác định 2 tham số:

* True Positive Rate: độ phủ của mô hình, thường được gọi là Recall và được tính dựa trên công thức sau:
* False Positive Rate: tỉ lệ các mẫu âm tính bị phân loại sai thành dương tính, được tính dựa trên công thức sau:

Đường cong ROC sẽ vẽ TPR và FPR ở các ngưỡng phân loại khác nhau. Hạ thấp ngưỡng phân loại sẽ cho ra nhiều mẫu Positive hơn, do đó làm tăng False Positve và True Positive. Ví dụ về đường cong ROC được thể hiện ở hình 19.

A picture containing text, screenshot, font, line

Description automatically generated

*Hình 19: Đường cong ROC*

Để tính toán các điểm trong đường cong ROC, phải đánh giá mô hình hồi quy logistic nhiều lần với các ngưỡng phân loại khác nhau, nhưng điều này sẽ không hiệu quả. Có một thuật toán Sort có thể cung cấp thông tin này, được gọi là AUC.

**AUC:** là từ viết tắt của “Area Under the ROC Curve”, thuật toán này có thể tính diện tích bên dưới đường cong ROC

AUC cung cấp độ đo đánh giá trên tất cả các ngưỡng phân loại có thể. AUC nằm trong khoảng giá trị [0,1]. Một mô hình có dự đoán sai 100% có AUC là 0.0 và một mô hình có dự đoán chính xác 100% có AUC là 1.0.

AUC đo xác suất một mẫu Positive xếp hạng cao hơn một mẫu Negative được chọn ngẫu nhiên. AUC chỉ tính đến thứ tự của các mẫu được dự đoán và không nhạy cảm với vấn đề mất cân bằng lớp.

### **5.2.2 LogLoss**

Trong bài báo này, tác giả sử dụng thêm Binary Cross Entropy để làm kết quả đánh giá mô hình.

**Binary Cross Entropy:** thường được sử dụng để tối ưu hàm mất mát hoặc đánh giá mô hình trong bài toán phân loại nhị phân. BCE được tính dựa trên công thức sau:

Tác giả dựa nhiều hơn vào BCE vì cần sử dụng xác suất dự đoán để ước tính lợi ích trong việc gợi ý.

**5.3. Kết quả của Tác Giả**

### **5.3.1 Cài đặt**

Tác giả sử dụng Tensorflow để thử nghiệm phương pháp đề xuất. Các siêu tham số của mô hình được lựa chọn bằng cách sử dụng grid-search trên tập dữ liệu validation. Learning rate được chỉnh ở tốc độ 0.001. Tác giả sử dụng Adam làm hàm tối ưu và sử dụng mini-batch size là 4096. Tác giả sử dụng điều chuẩn L2 với cho mạng DNN, DCN, Wide&Deep, DeepFM, xDeepFM và sử dụng dropout 0.5 cho mạng PNN.

Số lượng neuron mỗi layer mặc định là:

* Mạng DNN: 400 neuron
* Mạng CIN: 200 neuron cho bộ dữ liệu Criteo và 100 cho bộ dữ liệu Dianping và Bing News
* Số chiều embedding D cho mô hình: 10

### **5.3.2. So sánh các mạng**

Tác giả thử nghiệm mạng CIN khi hoạt động độc lập. Trước đó:

* Mạng FM học được sự tương tác rõ ràng giữa các đặc trưng bậc 2.
* Mạng DNN học được sự tương tác ngầm định giữa các đặc trưng bậc cao.
* Mạng CrossNet học được sự tương tác ngầm định giữa các đặc trưng bậc cao với số lượng tham số ít hơn.

Vì vậy không có sự đảm bảo về mặt lý thuyết nào về tính hiệu quả của một mạng riêng lẻ so với các mạng khác, do đó nó thực sự phụ thuộc vào tập dữ liệu. Ví dụ: nếu bộ dữ liệu không yêu cầu các tương tác đặc trưng bậc cao, thì mạng FM có thể là tốt nhất. Do đó, tác giả không có bất kỳ kỳ vọng về mạng nào sẽ hoạt động tốt nhất trong thử nghiệm này.

|  | **Criteo** | | | **Dianping** | | | **Bing News** | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| AUC | LogLoss | Depth | AUC | LogLoss | Depth | AUC | LogLoss | Depth |
| FM | 0.7900 | 0.4592 | - | 0.8165 | 0.3558 | - | 0.8223 | 0.2779 | - |
| DNN | 0.7993 | 0.4491 | 2 | 0.8318 | 0.3382 | 3 | 0.8366 | 0.273 | 2 |
| CrossNet | 0.7961 | 0.4508 | 3 | 0.8283 | 0.3404 | 2 | 0.8304 | 0.2765 | 6 |
| CIN | **0.8012** | 0.4493 | 3 | **0.8576** | **0.3225** | 2 | **0.8377** | **0.2662** | 5 |

*Bảng 5: Kết quả của các mạng trên từng bộ dữ liệu*

Bảng 5 cho thấy kết quả của các mạng riêng lẻ trên 3 bộ dữ liệu. Kết quả của mạng CIN vượt trội so với các mạng khác. Một mặt, kết quả chỉ ra rằng đối với các bộ dữ liệu thực tế, các tương tác bậc cao đối với các đặc trưng thưa là điều cần thiết và điều này có thể được xác minh thông qua mạng DNN, CrossNet và CIN vượt trội đáng kể so với FM trên cả ba bộ dữ liệu. Mặt khác, CIN là mạng tốt nhất, chứng minh được tính hiệu quả trong việc mô hình hóa các tương tác rõ ràng giữa các đặc trưng bậc cao.

### **5.3.3. So sánh các mô hình**

xDeepFM là mô hình kết hợp giữa mạng CIN và mạng DNN. Mặc dù CIN và DNN là 2 loại riêng biệt trong việc mô hình hóa các tương tác đặc trưng, do đó tác giả muốn biết liệu có thực sự cần thiết và hiệu quả khi kết hợp chúng lại với nhau để học các sự ảnh hưởng rõ ràng và ngầm định hay không.

Kết quả so sánh giữa các mô hình được thể hiện ở bảng 6. Mô hình LR kém hơn nhiều so với tất cả các mô hình còn lại, điều này chứng tỏ rằng các mô hình factorization-based là cần thiết để học các đặc trưng thưa. Wide&Deep, DCN, DeepFM và xDeepFM tốt hơn đáng kể so với DNN.

Mô hình xDeepFM được tác giả đề xuất có kết quả tốt nhất trên 3 bộ dữ liệu, cho thấy việc kết hợp sự tương tác giữa các đặc trưng bậc thấp và bậc cao là điều cần thiết.

Số layer tốt nhất của mô hình xDeepFM là 3, cho biết rằng các tương tác đặc trưng mà mô hình đã học được nhiều nhất là 4 bậc.

|  | **Criteo** | | | **Dianping** | | | **Bing News** | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| AUC | LogLoss | Depth | AUC | LogLoss | Depth | AUC | LogLoss | Depth |
| LR | 0.7577 | 0.4854 | -,- | 0.8018 | 0.3608 | -,- | 0.7988 | 0.2950 | -,- |
| FM | 0.7900 | 0.4592 | -,- | 0.8165 | 0.3558 | -,- | 0.8223 | 0.2779 | -,- |
| DNN | 0.7993 | 0.4491 | -,2 | 0.8318 | 0.3382 | -,3 | 0.8366 | 0.2730 | -,2 |
| DCN | 0.8026 | 0.4467 | 2,2 | 0.8391 | 0.3379 | 4,3 | 0.8379 | 0.2677 | 2,2 |
| Wide&Deep | 0.8000 | 0.4490 | -,3 | 0.8361 | 0.3364 | -,2 | 0.8377 | 0.2668 | -,2 |
| PNN | 0.8038 | 0.4927 | -,2 | 0.8445 | 0.3424 | -,3 | 0.8321 | 0.2775 | -,3 |
| DeepFM | 0.8025 | 0.4468 | -,2 | 0.8481 | 0.3333 | -,2 | 0.8376 | 0.2671 | -,3 |
| xDeepFM | 0.8052 | 0.4418 | 3.2 | 0.8639 | 0.3156 | 3.3 | 0.8400 | 0.2649 | 3.2 |

*Bảng 6: Kết quả của các mô hình trên từng bộ dữ liệu*

### **5.3.4. Kết quả của Nhóm**

Quy trình chuyển đổi dữ liệu không được tác giả đề cập trong bài báo. Do đó đối với các dữ liệu dạng số trong bộ dữ liệu Criteo và Dianping, nhóm sẽ chuẩn hóa các dữ liệu này về khoảng [0,1] và sử dụng Label Encoder để chuẩn hóa các dữ liệu hạng mục. Còn đối với bộ dữ liệu Bing News, nhóm sẽ ánh xạ từng từ sang tập từ vựng ở thuộc tính “new\_title” và “entity\_info”, các từ không tồn tại trong tập từ vựng sẽ được gán nhãn là “UNK”.

Như đã đề cập trong phần 5.1, bộ dữ liệu được nhóm sử dụng ít hơn nhiều so với tác giả, do đó kết quả mô hình bị giảm rõ rệt trên cả 3 bộ dữ liệu. Độ đo AUC giảm khoảng 4-5% đối với bộ dữ liệu Criteo và Bing News, giảm khoảng 9% đối với bộ dữ liệu Dianping.

|  | **Criteo** | | **Dianping** | | **Bing News** | |
| --- | --- | --- | --- | --- | --- | --- |
|  | AUC | LogLoss | AUC | LogLoss | AUC | LogLoss |
| Paper | 0.8052 | 0.4418 | 0.8639 | 0.3156 | 0.8400 | 0.2649 |
| Ours | 0.7507 | 0.4695 | 0.7764 | 0.4372 | 0.8048 | 0.5308 |

*Bảng 7: So sánh kết quả của nhóm và tác giả*

**6 KẾT LUẬN**

Trong bài báo này, tác giả đã đề xuất mạng Compressed Interaction Network (CIN) với mục tiêu là học được sự tương tác rõ ràng của các đặc trưng. Mạng CIN có 2 ưu điểm riêng biệt, đầu tiên mạng CIN có thể học sự tương tác với số bậc cụ thể và thứ hai là biểu diễn các tương tác này ở mức độ vector-wise.

Dựa theo ý tưởng từ các mô hình trước đó, tác giả kết hợp mạng CIN và mạng DNN thành một mô hình thống nhất và đặt tên là eXtreme Deep Factorization Machine (xDeepFM). Do đó, mô hình xDeepFM có thể tự động học được sự tương tác ngầm định và rõ ràng giữa các đặc trưng bậc cao.

Tác giả cũng tiến hành các thử nghiệm và kết quả chứng minh rằng mô hình xDeepFM của tác giả luôn vượt trội so với các mô hình trước đó trên ba bộ dữ liệu trong thế giới thực là Criteo, Dianping và Bing News. Kết quả này đã chứng minh tiềm năng của việc kết hợp cả tương tác đặc trưng rõ ràng và ngầm định trong việc cải thiện hiệu suất hệ thống gợi ý trong các ứng dụng thực tế.

Nhóm chúng tôi cũng đã thực hiện lại các thử nghiệm với mô hình của tác giả và kết quả đạt được cũng khá khả quan. Tuy nhiên, bởi vì dữ liệu được nhóm chúng tôi sử dụng còn khá ít nên kết quả thấp hơn so với bài báo của tác giả. Trong tương lai, nhóm sẽ thực hiện chia dữ liệu lớn này thành các phần nhỏ để huấn luyện song song để xem xét lại kết quả trên toàn bộ dữ liệu, đồng thời sẽ so sánh mô hình của tác giả với các mô hình sau này và trước đó.

**7 TÀI LIỆU THAM KHẢO**

[1] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric  
Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang  
Chen, et al . 2016. Deep speech 2: End-to-end speech recognition in english and  
mandarin. In International Conference on Machine Learning. 173–182

[2] Mathieu Blondel, Akinori Fujino, Naonori Ueda, and Masakazu Ishihata. 2016.  
Higher-order factorization machines. In Advances in Neural Information Process-  
ing Systems. 3351–3359.

[3] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-  
Seng Chua. 2017. Attentive collaborative filtering: Multimedia recommendation  
with item-and component-level attention. In Proceedings of the 40th International  
ACM SIGIR conference on Research and Development in Information Retrieval. ACM,  
335–344.

[4] Tianqi Chen, Weinan Zhang, Qiuxia Lu, Kailong Chen, Zhao Zheng, and Yong  
Yu. 2012. SVDFeature: a toolkit for feature-based collaborative filtering. Journal  
of Machine Learning Research 13, Dec (2012), 3619–3622.

[5] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra,  
Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al.  
2016. Wide & deep learning for recommender systems. In Proceedings of the 1st  
Workshop on Deep Learning for Recommender Systems. ACM, 7–10.

[6] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau,  
Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase  
representations using RNN encoder-decoder for statistical machine translation.  
arXiv preprint arXiv:1406.1078 (2014).

[7] Xin Dong, Lei Yu, Zhonghuo Wu, Yuxia Sun, Lingfeng Yuan, and Fangxi Zhang.  
2017. A Hybrid Collaborative Filtering Model with Deep Structure for Recom-  
mender Systems. In AAAI. 1309–1315.

[8] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A multi-view deep  
learning approach for cross domain user modeling in recommendation systems. In  
Proceedings of the 24th International Conference on World Wide Web. International  
World Wide Web Conferences Steering Committee, 278–288.

[9] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017.  
Deepfm: A factorization-machine based neural network for CTR prediction. arXiv  
preprint arXiv:1703.04247 (2017).

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual  
learning for image recognition. In Proceedings of the IEEE conference on computer  
vision and pattern recognition. 770–778.

[11] Ruining He and Julian McAuley. 2016. VBPR: Visual Bayesian Personalized  
Ranking from Implicit Feedback. In AAAI. 144–150.

[12] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse  
predictive analytics. In Proceedings of the 40th International ACM SIGIR conference  
on Research and Development in Information Retrieval. ACM, 355–364.

[13] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng  
Chua. 2017. Neural collaborative filtering. In Proceedings of the 26th International  
Conference on World Wide Web. International World Wide Web Conferences  
Steering Committee, 173–182.

[14] Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine  
Atallah, Ralf Herbrich, Stuart Bowers, et al. 2014. Practical lessons from predicting  
clicks on ads at facebook. In Proceedings of the Eighth International Workshop on  
Data Mining for Online Advertising. ACM, 1–9.

[15] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed,  
Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N  
Sainath, et al . 2012. Deep neural networks for acoustic modeling in speech  
recognition: The shared views of four research groups. IEEE Signal Processing  
Magazine 29, 6 (2012), 82–97.

[16] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic opti-  
mization. arXiv preprint arXiv:1412.6980 (2014).

[17] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted  
collaborative filtering model. In Proceedings of the 14th ACM SIGKDD international  
conference on Knowledge discovery and data mining. ACM, 426–434.

[18] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization tech-  
niques for recommender systems. Computer 42, 8 (2009).

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classifica-  
tion with deep convolutional neural networks. In Advances in neural information  
processing systems. 1097–1105.

[20] Joonseok Lee, Seungyeon Kim, Guy Lebanon, and Yoram Singer. 2013. Local  
low-rank matrix approximation. In International Conference on Machine Learning.  
82–90.

[21] Jianxun Lian and Xing Xie. 2016. Cross-Device User Matching Based on Massive  
Browse Logs: The Runner-Up Solution for the 2016 CIKM Cup. arXiv preprint  
arXiv:1610.03928 (2016).

[22] Jianxun Lian, Fuzheng Zhang, Min Hou, Hongwei Wang, Xing Xie, and  
Guangzhong Sun. 2017. Practical Lessons for Job Recommendations in the  
Cold-Start Scenario. In Proceedings of the Recommender Systems Challenge 2017  
(RecSys Challenge ’17). ACM, New York, NY, USA, Article 4, 6 pages. https:  
//doi.org/10.1145/3124791.3124794

[23] Jianxun Lian, Fuzheng Zhang, Xing Xie, and Guangzhong Sun. 2017. CCCFNet:  
a content-boosted collaborative filtering neural network for cross domain rec-  
ommender systems. In Proceedings of the 26th International Conference on World  
Wide Web Companion. International World Wide Web Conferences Steering  
Committee, 817–818.

[24] Jianxun Lian, Fuzheng Zhang, Xing Xie, and Guangzhong Sun. 2017. Restaurant  
Survival Analysis with Heterogeneous Information. In Proceedings of the 26th  
International Conference on World Wide Web Companion. International World  
Wide Web Conferences Steering Committee, 993–1002.

[25] Xiaoliang Ling, Weiwei Deng, Chen Gu, Hucheng Zhou, Cui Li, and Feng Sun.  
2017. Model Ensemble for Click Prediction in Bing Search Ads. In Proceedings of  
the 26th International Conference on World Wide Web Companion. International  
World Wide Web Conferences Steering Committee, 689–698.

[26] Guimei Liu, Tam T Nguyen, Gang Zhao, Wei Zha, Jianbo Yang, Jianneng Cao, Min  
Wu, Peilin Zhao, and Wei Chen. 2016. Repeat buyer prediction for e-commerce.  
In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge  
Discovery and Data Mining. ACM, 155–164.

[27] H Brendan McMahan, Gary Holt, David Sculley, Michael Young, Dietmar Ebner,  
Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, et al . 2013.  
Ad click prediction: a view from the trenches. In Proceedings of the 19th ACM  
SIGKDD international conference on Knowledge discovery and data mining. ACM,  
1222–1230.

[28] Aditya Krishna Menon and Charles Elkan. 2010. A log-linear model with latent  
features for dyadic prediction. In Data Mining (ICDM), 2010 IEEE 10th International  
Conference on. IEEE, 364–373.

[29] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černock `y, and Sanjeev Khu-  
danpur. 2010. Recurrent neural network-based language model. In Eleventh  
Annual Conference of the International Speech Communication Association.

[30] Rong Pan, Yunhong Zhou, Bin Cao, Nathan N Liu, Rajan Lukose, Martin Scholz,  
and Qiang Yang. 2008. One-class collaborative filtering. In Data Mining, 2008.  
ICDM’08. Eighth IEEE International Conference on. IEEE, 502–511.

[31] Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang.  
2016. Product-based neural networks for user response prediction. In Data Mining  
(ICDM), 2016 IEEE 16th International Conference on. IEEE, 1149–1154.

[32] Steffen Rendle. 2010. Factorization machines. In Data Mining (ICDM), 2010 IEEE  
10th International Conference on. IEEE, 995–1000.

[33] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme.  
2009. BPR: Bayesian personalized ranking from implicit feedback. In Proceedings  
of the twenty-fifth conference on uncertainty in artificial intelligence. AUAI Press,  
452–461.

[34] Steffen Rendle and Lars Schmidt-Thieme. 2010. Pairwise interaction tensor  
factorization for personalized tag recommendation. In Proceedings of the third  
ACM international conference on Web search and data mining. ACM, 81–90.

[35] Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting  
clicks: estimating the click-through rate for new ads. In Proceedings of the 16th  
international conference on World Wide Web. ACM, 521–530.

[36] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015.  
Autorec: Autoencoders meet collaborative filtering. In Proceedings of the 24th  
International Conference on World Wide Web. ACM, 111–112.

[37] Ying Shan, T Ryan Hoens, Jian Jiao, Haijing Wang, Dong Yu, and JC Mao. 2016.  
Deep crossing: Web-scale modeling without manually crafted combinatorial  
features. In Proceedings of the 22nd ACM SIGKDD International Conference on  
Knowledge Discovery and Data Mining. ACM, 255–262.

[38] Nathan Srebro, Jason Rennie, and Tommi S Jaakkola. 2005. Maximum-margin  
matrix factorization. In Advances in neural information processing systems. 1329–  
1336.

[39] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning  
for recommender systems. In Proceedings of the 21th ACM SIGKDD International  
Conference on Knowledge Discovery and Data Mining. ACM, 1235–1244.

[40] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & Cross Network  
for Ad Click Predictions. arXiv preprint arXiv:1708.05123 (2017).

[41] Xinxi Wang and Ye Wang. 2014. Improving content-based and hybrid music rec-  
ommendation using deep learning. In Proceedings of the 22nd ACM international  
conference on Multimedia. ACM, 627–636.

[42] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. 2016. Collabora-  
tive denoising auto-encoders for top-n recommender systems. In Proceedings of  
the Ninth ACM International Conference on Web Search and Data Mining. ACM,  
153–162.

[43] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua.  
2017. Attentional Factorization Machines: Learning the Weight of Feature Inter-  
actions via Attention Networks. In Proceedings of the Twenty-Sixth International  
Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August  
19-25, 2017. 3119–3125. <https://doi.org/10.24963/ijcai.2017/435>

[44] Fajie Yuan, Guibing Guo, Joemon M Jose, Long Chen, Haitao Yu, and Weinan  
Zhang. 2016. Lambdafm: learning optimal ranking with factorization machines  
using lambda surrogates. In Proceedings of the 25th ACM International on Confer-  
ence on Information and Knowledge Management. ACM, 227–236.

[45] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma.  
2016. Collaborative knowledge base embedding for recommender systems. In  
Proceedings of the 22nd ACM SIGKDD international conference on knowledge  
discovery and data mining. ACM, 353–362.

[46] Weinan Zhang, Tianming Du, and Jun Wang. 2016. Deep learning over multi-field  
categorical data. In European conference on information retrieval. Springer, 45–57.

[47] Guorui Zhou, Chengru Song, Xiaoqiang Zhu, Xiao Ma, Yanghui Yan, Xingya  
Dai, Han Zhu, Junqi Jin, Han Li, and Kun Gai. 2017. Deep interest network for  
click-through rate prediction. arXiv preprint arXiv:1706.06978 (2017).