دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

رساله ارشد
هوش مصنوعی

عنوان رساله

پیش‌بینی احتمال تعامل کاربران در تبلیغات نمایشی

نگارش
محمدرضا رضائی

استاد راهنما
حمیدرضا ربیعی

زمستان ۱۳۹۹

دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

رساله ارشد

پیش‌بینی احتمال تعامل کاربران در تبلیغات نمایشی

نگارش: محمدرضا رضائی

استاد راهنما:	حمیدرضا ربیعی	امضاء:
داور داخلی:	مهدیه سلیمانی	امضاء:
داور خارجی:	مصطفی صالحی	امضاء:

چکیده

امروزه تبلیغات برخط بخش زیادی از وب‌سایت‌ها و برنامه‌های موبایلی را دربر گرفته است. در این نوع تبلیغات به محض تعامل کاربر با سایت یا برنامه موبایل باید در کسری از ثانیه در مورد اینکه چه تبلیغی به وی نشان داده شود تصمیم گرفته شود[1]. در سامانه‌های تبلیغ برخط، درآمد این سیستم‌ها معمولا پس از کلیک کاربر روی تبلیغ یا تعامل کاربر با تبلیغ صورت می‌گیرد و لذا روش معمول این است که برای انتخاب تبلیغ برای نمایش به کاربر، ابتدا احتمال کلیک یا تعامل کاربر با تبلیغات مختلف را محاسبه کرده و سپس بر اساس این احتمال و مبلغ درآمد به ازای تبلیغات مختلف، یک تبلیغ را به عنوان تبلیغ برنده انتخاب و به کاربر نمایش می‌دهند. لذا یکی از مهم‌ترین مسائل در تبلیغات برخط پیش‌بینی احتمال کلیک کاربر بر روی تبلیغات مختلف است که مورد توجه زیادی در حوزه تحقیقات دانشگاهی قرار گرفته است. محاسبه دقیق این احتمال تعامل، از طرفی باعث نمایش تبلیغات مرتبط‌تر به کاربران و افزایش رضایت آن‌ها خواهد شد و از طرفی دیگر درآمد سیستم‌های تبلیغاتی را افزایش خواهد داد.

تحقیقات قبلی در حوزه پیش‌بینی احتمال کلیک و تعامل، مساله را به یک مساله دسته‌بندی دودویی تبدیل می‌کنند و با استفاده از اطلاعات موجود در تاریخچه که به سه دسته‌ی سمت کاربر، سمت تبلیغ دهنده و سمت نمایش دهنده تقسیم می‌شود، سعی در پیش‌بینی احتمال تعامل دارند. چالش‌هایی نظیر نامتوازن بودن کلاس‌ها، تنک بودن داده‌ها، بعد زیاد و شروع سرد، این مساله را به کلی از مسائل سنتی دسته‌بندی متفاوت می‌کنند. روش‌های موجود در این حوزه را می‌توان به دو دسته روش‌های کم عمق و روش‌های ژرف دسته‌بندی کرد. با توجه به سادگی پیاده‌سازی و قابلیت موازی‌سازی، روش‌های کم عمق در عمل استفاده بیشتری داشته‌اند.

در این پژوهش، با بررسی مساله‌ی پیش‌بینی احتمال نرخ تعامل کاربران با تبلیغات، و همچنین با تاکید بر چالش‌های گفته شده، روش جدیدی برای حل این مساله پیشنهاد می‌دهیم. برای طراحی روش پیشنهادی، از مجموعه‌ی متنوعی از ایده‌های موجود و همچنین جدید بهره گرفته و این مدل را در راستای مقاوم بودن در برابر چالش‌های مساله، طراحی نموده و با بررسی معیارهای ارزیابی نظیر مساحت تحت منحنی، دقت و بازیابی، عملکرد آن را روی مجموعه داده‌های استاندارد می‌آزماییم. با بررسی نتایج آزمایش‌ها، نتیجه می‌گیریم مدل پیشنهادی عملکرد قابل قبولی ارائه کرده و در نتیجه قابل آزمایش در شرایط آنلاین و واقعی است.

کلمات کلیدی: تبلیغات نمایشی، کاربر، احتمال تعامل، بردارهای تعبیه، تعامل بین ویژگی‌ها

[1] استاندارد پذیرفته شده در دنیا حدود ۱۰۰ میلی‌ثانیه است

فهرست مطالب

فهرست تصاویر

فهرست جداول

فصل ۱

مقدمه

انسان برای رفع نیازهای خود به اقتصاد وابسته است. برای توسعه‌ی چرخه‌های اقتصادی، باید عوامل مهمی از قبیل افزایش تولید و گذر از تولید دستی به انبوه و همچنین بازاریابی مناسب را در نظر گرفت. یکی از عوامل دست یافتن به بازاریابی مناسب، انجام تبلیغات صحیح برای محصولات است.

امروزه با گسترش اینترنت، شاهد تاثیرگذاری آن بر اکثر جنبه‌های زندگی بشری، از جمله اقتصاد هستیم. یکی از نمودهای این تاثیرگذاری، ظهور تبلیغات آنلاین در مقابل گونه‌های سنتیِ آن است. مقرون به صرفه بودن، در دسترس بودن در مقیاس جهانی و قابلیت گرفتن بازخورد مستقیم از کاربران مورد نظر از جمله برتری‌های قابل توجه تبلیغات آنلاین است.

تبلیغات آنلاین، به شیوه‌های متنوعی انجام می‌شود.[۱] تعدادی از گونه‌های این نوع تبلیغات، وبسایت‌ها، شبکه‌های اجتماعی[۲]، تبلیغات کلمه کلیدی[۳]، بهینه‌سازی موتورهای جستجو[۴] و تبلیغات نمایشی[۵] هستند.

در تبلیغات نمایشی، استفاده از بنر[۶]های ثابت، انیمیشنی و ویدیویی و نشان دادن آن به کاربر[۷] در کادرهای از پیش تعیین شده داخل وب‌سایت‌ها یا برنامه‌های موبایل به عنوان روشی کارآمد برای جذب مخاطب به کار می‌رود؛ اما انتخاب این که کدام بنر در کدام کادر (کدام صفحه‌ی وب) به کدام کاربر نمایش داده‌شود، چالش قابل توجهی است.

[2] Social Networks
[3] Keyword Advertising
[4] SEO
[5] Display Advertising
[6] Banner
[7] User

۱-۱ معرفی انواع معاملات در تبلیغات نمایشی

از آنجا که درآمد بسیاری از صاحبان صفحات وب، تنها از تبلیغات نمایشی انجام شده در وب‌سایت‌هایشان حاصل می‌شود، انتخاب نحوه‌ی قرارداد با تبلیغ کننده[1]ها اهمیت زیادی برای آن‌ها دارد.[۲] در این بخش به طور مختصر انواع قراردادهای رایج بین تبلیغات کننده‌ها و صاحبان صفحات وب را توضیح می‌دهیم.

- **قراردادهای مستقیم**

در ابتدای ظهور تبلیغات آنلاین نمایشی، تبلیغ کننده با صاحب وب‌سایت قرارداد مستقیم[2] بسته و با انتخاب یک کادر ثابت در وب‌سایت و یک بنر تبلیغاتی مشخص، تا مدت (یا تعداد کلیک) مشخصی با نمایش دادن تبلیغ یکسان به تمامی کاربرانی که از آن صفحه‌ی به خصوص بازدید می‌کردند، تبلیغات خود را نمایش می‌دادند. با وجود این که تعدادی وب‌سایت هنوز از چنین روشی استفاده می‌کنند؛ واضح است که به کار گرفتن آن برای تعداد بالای صفحات و تبلیغات، هزینه و زحمت قابل توجهی را به هر دو طرف معامله تحمیل می‌کند. به دلیل این مشکل، سراغ دسته‌ای از قراردادها می‌رویم که به معاملات برنامه‌ریزی شده[3] معروف‌اند.

- **قراردادهای برنامه‌ریزی شده**

در بقیه‌ی روش‌ها، که جزء شاخه‌ی برنامه‌ریزی شده طبقه بندی می‌شوند، با رعایت کردن یک استاندارد مشترک، میزان هزینه و زحمت مورد نیاز کاهش یافته و فرآیند سریعتر انجام می‌شود. معاملات برنامه‌ریزی شده به دو دسته‌ی معاملات تضمین شده[4] و مزایده‌ی بلادرنگ[5] تقسیم می‌شوند.

  - **قراردادهای تضمین شده**

در این دسته از قراردادها، هزینه و تعداد بنرهایی که باید به کاربران نشان داده شوند، از پیش تعیین می‌شود. نکته‌ی حائز اهمیت در این دسته از قراردادها، اضافه شدن سیستم‌هایی است که به صورت اتوماتیک بخش‌های قابل توجهی از فرآیند نمایش تبلیغ را انجام داده و با حذف دخالت انسانی، هزینه‌ها و زحمات کار را به شدت کاهش می‌دهند. دو دسته‌ی مهم از این قراردادها، دسته‌ی قراردادهای تضمین شده‌ی اتوماتیک[6] و قراردادهای تضمین شده‌ی برنامه‌ریزی شده[7] نامیده می‌شوند.

    * **قراردادهای تضمین شده‌ی اتوماتیک**

همانطور که در بخش قبل گفته شد، در قراردادهای تضمین شده‌ی اتوماتیک، تمرکز بر خودکارسازی[8] فرآیند نمایش تبلیغ است. یکی از مهم‌ترین فواید خودکارسازی نمایش

[1] Advertisier
[2] Direct Deas
[3] Programmatic Deals
[4] Guaranteed Deals
[5] Realtime Bidding (RTB)
[6] Automated Guaranteed Deals
[7] Programmatic Guaranteed Deals
[8] Automation

تبلیغ برای تبلیغ کننده، امکان تبلیغ همزمان در چندین وب‌سایت بدون نیاز به عقد چندین قرارداد است.

* **قراردادهای تضمین شده‌ی برنامه‌ریزی شده**

در این دسته از قراردادها علاوه بر ساده‌سازی‌هایی که در قراردادهای تضمین شده‌ی اتوماتیک انجام می‌شود، امکان تنظیمات جزئی‌تری برای تبلیغ کننده وجود داشته و در نتیجه این دسته از قراردادها بسیار محبوب‌تر از قراردادهای تضمین شده‌ی اتوماتیک هستند. در قراردادهای تضمین شده‌ی برنامه ریزی شده، تبلیغ کننده می‌تواند با اعمال چندین قاعده‌ی محدود کننده، نمایش بنر خود را برای کاربران مختلف فیلتر کرده و عملا بنر تبلیغاتی خود را فقط برای کاربرانی با مشخصات از پیش تعیین شده نمایش دهد. به عنوان مثال فرض کنید یک شرکت می‌تواند فروش کالاهای خود را برای کشورهای خاصی انجام دهد و برای فیلتر کردن کاربران، تنظیماتی را اعمال می‌کند که با دریافت اطلاعات مرورگر، در صورتی که آدرس آی پی[1] کاربر خارج از بازه‌ی سرویس دهی شرکت باشد، از انجام تبلیغ صرف نظر کند. به این ترتیب این شرکت میزان قابل توجهی از هزینه‌های تبلیغاتی خود را از هدر رفت باز می‌دارد.

- **مزایده‌ی بلادرنگ**

تفاوت مزایده‌های بلادرنگ با معاملات تضمین شده، در مشخص کردن قیمت و تعداد دفعات نمایش دادن تبلیغات به کاربران است. در مزایده‌های بلادرنگ، هزینه‌ی هر تبلیغ به طور جداگانه در هنگام درخواست بارگیری صفحه توسط کاربر، توسط یک مزایده[2] بین تبلیغ کنندگان تعیین می‌شود.

* **مزایده‌ی بلادرنگ آزاد**

در مزایده‌های بلادرنگ آزاد[3]، هربار که یک کاربر به یکی از صفحات دارای کادر مناسب برای تبلیغ وارد می‌شود، همه‌ی تبلیغ کنندگان می‌توانند یک قیمت برای نمایش تبلیغ خود به کاربر، پیشنهاد دهند و تبلیغ دارای بالاترین پیشنهاد قیمت، به کاربر نمایش داده می‌شود. امروزه این نوع معامله به دلیل هزینه‌ی پایین برای تبلیغ کنندگان و درآمد بالا برای صاحبان صفحات وب، میزان قابل توجهی از تبلیغات کنندگان و صاحبان صفحات وب در سراسر جهان را به خود جذب کرده است.

* **مزایده‌ی بلادرنگ خصوصی**

در مزایده‌های بلادرنگ خصوصی[4]، تبلیغات کنندگان باید قبل از شروع فرآیند تبلیغ وارد قرارداد شده و با قبول شرایط اولیه‌ای که صاحب صفحات وب پیشنهاد می‌کند، وارد فرآیند مزایده شود.

در این پایان نامه، بر نوع مزایده‌های بلادرنگ آزاد تمرکز خواهیم داشت و جزئیات و چالش‌های مربوط به آن را بررسی خواهیم کرد.

[1] IP Address
[2] Auction
[3] Open Realtime Auction
[4] Private Realtime Auction

۱-۲ اجزا و نحوه‌ی اجرای مزایده‌های بلادرنگ

در عمل، برای انجام مزایده‌های بلادرنگ، به اجزا و نقش‌های متنوعی نیاز است.[۲] در این بخش اصطلاحات استفاده شده در مزایده‌های بلادرنگ و همچنین اجزا و نقش‌های آن را تعریف کرده و توضیح می‌دهیم.

۱-۲-۱ کاربر

تعریف کاربر در مزایده‌های بلادرنگ، با تعریفی که در بخش قبل ذکر شد، تفاوت چندانی ندارد. تنها فرق جزئی در این نکته است که اینجا، تمرکز بیشتر روی مرورگری است که کاربر استفاده می‌کند و اعمالی که در این بخش به کاربر نسبت می‌دهیم، عملا توسط مرورگر کاربر انجام می‌شود و خود کاربر اطلاعی از انجام آن‌ها ندارد.

۱-۲-۲ ناشر

در ادبیات مزایده‌های بلادرنگ، ناشر[1] به وب‌سایتی اشاره می‌کند که در آن امکان انجام تبلیغات وجود دارد و لذا [بخشی از] درآمد این وب‌سایت از تبلیغات است. از ملزومات اجرای فرآیند مزایده‌های بلادرنگ، وجود اسکریپت‌های مربوط به سکوی سمت تامین در این صفحه است.

۱-۲-۳ سکوی سمت تامین

سکوی سمت تامین[2] به بخشی از زیرساخت اطلاق می‌شود که با تعدادی ناشر قرارداد بسته و از طریق تعدادی اسکریپت که در سایت ناشرها تعبیه کرده است، اجرای فرآیند مزایده را ممکن می‌سازد.

این اسکریپت‌ها، برخی اطلاعات از جمله سوابق مرور کاربر در همه‌ی وب‌سایت‌هایی که این اسکریپت در آن‌ها وجود دارد را به سکوی سمت تامین ارسال کرده و در هنگام نیاز به نمایش تبلیغ، اطلاعاتی از جمله موقعیت جغرافیایی، نحوه‌ی اتصال به وب‌سایت (موبایل، تبلت یا کامپیوتر) و حتی نحوه‌ی ورود به وب‌سایت (موتور جستجو، ایمیل تبلیغاتی، لینک توصیه شده از طرف کاربر دیگر و...) را به این سکو ارسال می‌کند؛ لذا سکوی سمت تامین اطلاعات جامعی از این کاربر در اختیار داشته و بر اساس این اطلاعات، تبلیغات مناسب را در اختیار کاربر قرار دهد.

[1] Publisher
[2] Supply Side Platform

۱-۲-۴ سکوی سمت نیاز

سکوی سمت نیاز[1] به بخشی از زیرساخت اطلاق می‌شود که با تعدادی تبلیغ کننده (بازاریاب) ارتباط داشته و عملا شرکت‌کننده‌های اصلی مزایده، آن‌ها هستند. سکوهای سمت نیاز برای هر موقعیت قابل تبلیغ[2] وارد مزایده شده و قیمت پیشنهادی خود را برای انجام تبلیغ ارائه می‌کنند.

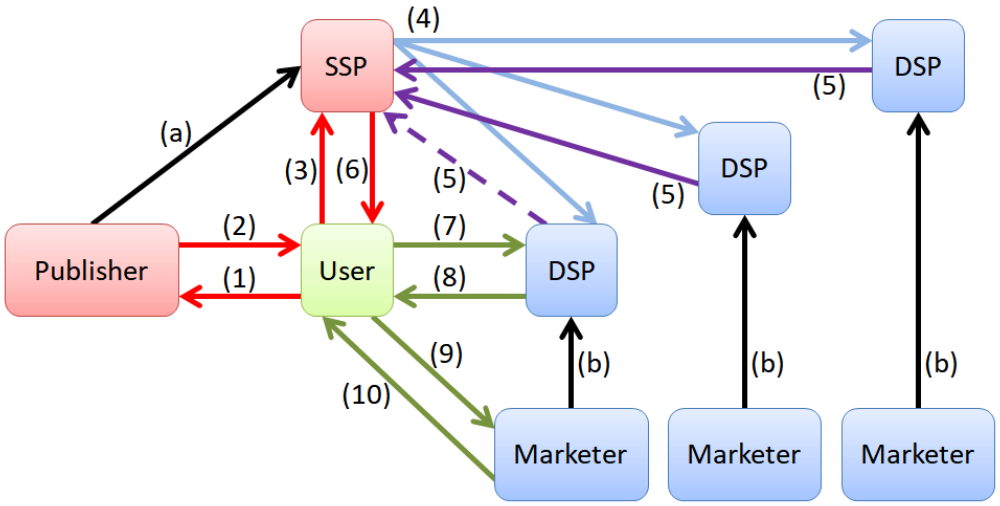
۱-۲-۵ تبلیغ کننده

تبلیغ کننده (بازاریاب[3]) در بخش قبلی به صورت کامل تعریف شده است. آن‌ها برای انجام تبلیغ و بازاریابی کالا یا خدماتی که ارائه می‌دهند، دست به تبلیغ زده و بودجه‌ی قابل توجهی را روانه‌ی زیرساخت‌های تبلیغاتی می‌کنند. بازاریاب‌ها با سکوهای سمت نیاز قرارداد بسته و تبلیغات خود را به آن‌ها ارائه کرده و به ازای تعداد کلیک کاربران روی تبلیغاتشان، به آن‌ها پرداخت می‌کنند. به عنوان مثال، سکوی سمت نیاز در قراردادی تضمین می‌کند تعداد ۱۰۰۰ کلیک بر روی بنر تبلیغاتی یکی از تبلیغ کننده‌ها تامین کرده و در قبال آن، هزینه‌ای دریافت کند.

۱-۲-۶ اجرای فرآیند مزایده‌های بلادرنگ

فرآیند مزایده‌ی بلادرنگ، از کاربر شروع می‌شود. زمانی که کاربر وارد صفحه‌ای متعلق به یک ناشر می‌شود، مرورگر کاربر یک درخواست برای نمایش وب‌سایت ناشر ارسال می‌کند(۱). وب‌سایت ناشر، صفحه‌ی اچ‌تی‌ام‌ال[4] خود را برای کاربر ارسال کرده و همزمان لینک مربوط به اسکریپت سکوی سمت تامین را در اختیار کاربر می‌گذارد(۲). کاربر برای بارگیری صفحه‌ی اچ‌تی‌ام‌ال دریافتی، سراغ تک‌تک منابع رفته و هرکدام را بارگیری می‌کند. برای نمایش اطلاعاتی که در کادر تبلیغ وجود دارد، کاربر یک درخواست اچ‌تی‌تی‌پی[5] به سکوی سمت تامین ارسال می‌کند(۳). در این مرحله، سکوی سمت تامین وارد عمل شده و موقعیت قابل تبلیغ و اطلاعات کاربر را از قبیل سابقه‌ی کاربر، مشخصات و سابقه‌ی سایت ناشر و اطلاعات مربوط به ابعاد کادر تبلیغ به تمامی سکوهای سمت نیاز در دسترس ارسال می‌کند(۴). هر سکوی سمت نیاز با در نظر گرفتن تبلیغ خود، با استفاده از روش‌های مختلف (که نمونه‌های آن در فصل ۲ توضیح داده می‌شود) یک قیمت به عنوان هزینه‌ی پیشنهادی نمایش تبلیغ ارائه می‌دهد. پیشنهادی که بیشترین قیمت را پیشنهاد داده باشد، برنده‌ی مزایده می‌شود. پیشنهاد برنده با خط‌چین نمایش داده شده است(۵). پس از دریافت هزینه‌های پیشنهادی سکوهای سمت نیاز، سکوهای سمت تامین بالاترین قیمت را انتخاب کرده و لینک سکوی سمت نیاز برنده را به کاربر ارسال می‌کند(۶). کاربر با کسب اطلاع از آدرس مشخصات

[1] Demand Side Platform
[2] Impression
[3] Marketer
[4] HTML
[5] HTTP



شکل ۱–۱: فرآیند مزایده‌ی بلادرنگ

سکوی سمت نیاز برنده، برای اطلاع از محل نهایی بنر تبلیغ انتخاب شده، به آن آدرس رجوع می‌کند(۷). سکوی سمت نیاز برنده به درخواست کاربر پاسخ داده و آدرس بنر (که در سرور متعلق به بازاریاب است) را برای کاربر ارسال می‌کند(۸). کاربر به آدرس بنر رجوع می‌کند(۹). سرور بازاریاب بنر تبلیغ را به کاربر ارسال می‌کند(۱۰). مراحل اجرای این فرآیند در شکل ۱–۱ قابل ملاحظه است.

نکته‌ی قابل توجه در فرآیند مزایده‌ی بلادرنگ، تفاوت نوع قراردادهای بسته شده بین سکوهای سمت نیاز با تبلیغ کننده‌ها و سکوهای سمت تامین با ناشران است. سکوهای سمت تامین به ازای **نمایش** هر تبلیغ به ناشران مبلغی پرداخت می‌کنند؛ اما سکوهای سمت تامین به ازای هر **کلیک انجام شده** روی بنرهای تبلیغ کننده‌ها، مبلغی از آن‌ها دریافت می‌کنند؛ بنابراین برای تضمین سوددهی این سیستم، باید تبلیغاتی برای نمایش به کاربران انتخاب شوند که احتمال کلیک شدن روی آن‌ها قابل توجه باشد؛ پس تخمین این احتمال که به نرخ کلیک[1] معروف است، به یک مساله‌ی محوری در این فرآیند تبدیل می‌شود.[۳]

لازم به ذکر است در برخی قراردادهای دیگر، نوع قرارداد بین سکوهای سمت تامین و تبلیغ کنندگان، به جای تضمین تعداد کلیک انجام شده، تضمین تعداد خرید انجام شده از طریق بنر مربوطه است؛ پس به جای تخمین نرخ کلیک، احتمال انجام خرید از طریق تبلیغ نمایش داده شده تخمین زده می‌شود که به نرخ تبدیل[2] معروف است. در عمل می‌توان نرخ تبدیل را ضریبی از نرخ کلیک در نظر گرفت که به دلیل تنک بودن، کار کردن با آن چالش بیشتری دارد. در این پژوهش به دلیل محدودیت در مجموعه‌های داده‌ی انتخاب شده، تنها از نرخ کلیک استفاده می‌کنیم.

[1] Click Through Rate
[2] Convertion Rate

۱-۳ چالش‌ها

در تخمین نرخ کلیک و نرخ تبدیل، چالش‌هایی وجود دارند که کار پژوهش در این موضوع را دچار مشکل می‌کنند. در این بخش به اختصار در مورد این چالش‌ها بحث می‌کنیم.

- چالش عدم توازن شدید کلاس‌ها[1]

هنگام دسته‌بندی دودویی در مساله‌ای که داده‌ها به صورت نامتوازن هستند، با چالش جدی عدم توازن کلاس‌ها روبرو هستیم.[۴] در تبلیغات نمایشی، در بیشتر موارد کاربر روی تبلیغ کلیک نمی‌کند و یا پس از کلیک، بازدید کاربر از صفحه‌ی مقصد به خرید (تبدیل) منتهی نمی‌شود و این شرایط باعث می‌شود این مساله نیز جزء مسائل مواجه با چالش عدم توازن شدید کلاس‌ها باشد.

- چالش ابعاد بالا[2]

به دلیل وجود تعداد ابعاد ورودی بسیار بالا، رویارویی با این مساله با الگوریتم‌های ساده‌ی یادگیری ممکن نیست. این مشکل با نام دیگر نفرین ابعاد[3] نیز معروف است. نفرین ابعاد باعث می‌شود تعداد پارامترهای مدل بیشتر شده و در نتیجه فرآیند یادگیری آن دچار مشکلات متنوعی شود.[۵]

- چالش شروع سرد[4]

وقتی یک تبلیغ جدید برای نمایش اضافه می‌شود، سکوهای سمت نیاز هیچ اطلاعاتی در مورد آن و کاربرهایی که احتمالا به آن تبلیغ علاقه نشان دهند، ندارند؛ لذا تعداد زیادی از موقعیت‌های قابل تبلیغ و در نتیجه میزان قابل توجهی هزینه صرف شناسایی تبلیغ جدید می‌شود. از طرفی، کاربر جدیدی که شروع به بازدید از صفحات مربوط به ناشرین می‌کند، از طرف سکوهای سمت تامین مورد نظر شناخته شده نیست؛ پس وقت و هزینه‌ی زیادی صرف شناختن سلایق این کاربر جدید می‌شود. این مشکل در ادبیات سیستم‌های پیشنهاد دهنده[5] به نام شروع سرد معروف است.[۶، ۷]

- چالش سرعت آموزش

بسیاری از شرکت‌هایی که خدمات مربوط به مزایده‌های بلادرنگ را ارائه می‌دهند، به دلیل تغییرات روزانه‌ی زیاد در مجموعه‌های داده، عمل آموزش مدل‌هایشان را در فواصل زمانی کوتاه (مثلا هر روز) تکرار می‌کنند. پس مدل‌هایی که آموزش آن‌ها زمان‌بر باشد، قابل استفاده در عمل نخواهند بود؛ لذا علاوه بر چالش‌هایی که ذکر شد، مدل ارائه شده باید توازنی بین عملکرد مناسب و سرعت آموزش ایجاد کند.

[1] High class imbalance challenge
[2] High dimentionality challenge
[3] Curse of dimentionality
[4] Cold start challenge
[5] Recommender systems

۱-۴ هدف پژوهش

در فرآیند مزایده‌های بلادرنگ، تنها نکته‌ای که در آن اجماع عمومی وجود ندارد، روشی است که با آن نرخ کلیک یا نرخ تبدیل تخمین زده شده و هزینه‌ی پرداختی به هر موقعیت قابل تبلیغ بر مبنای آن محاسبه و پیشنهاد می‌شود؛ لذا هدف کلی این پژوهش، ارائه‌ی یک مدل یادگیری ماشین برای تخمین نرخ کلیک است.

۱-۵ پرسش‌های اساسی پژوهش

برای رسیدن به هدف کلی این پژوهش که ارائه‌ی یک راهکار جدید برای تخمین نرخ کلیک است، باید مشخص شود که **چه راهکاری برای مواجهه با چالش‌های موجود، مناسب بوده و می‌تواند با وجود همه‌ی این چالش‌ها تخمین قابل قبولی از نرخ کلیک ارائه دهد؟** بنابراین، پرسش‌های زیر پیش‌رویمان خواهد بود:

۱. روش‌های موجود برای تخمین نرخ کلیک در تبلیغات نمایشی، کدامند؟

۲. هریک از چالش‌های مهم تخمین نرخ کلیک، چه تاثیری بر عملکرد روش‌ها می‌گذارند؟

۳. روش مناسبی که با این چالش‌ها رویارو شود، باید چه ویژگی‌هایی داشته باشد؟

۱-۶ ساختار رساله

در فصل دوم این رساله، پس از معرفی برخی از پیش‌نیازها، روش‌های پیشین را معرفی، دسته‌بندی و مقایسه کرده و در مورد مزایا و معایب هرکدام در رویارویی با چالش‌های مربوط به مساله می‌اندیشیم. در فصل سوم، با توجه به چالش‌ها و کاستی‌های روش‌های پیشین، مدل پیشنهادی خود را گام به گام طراحی کرده و با ارائه‌ی دلایل شهودی و ریاضی، ایده‌های ارائه شده را توجیه می‌کنیم؛ سپس مدل پیشنهادی را فرموله‌بندی کرده و پیش‌نیازهای لازم برای آموزش آن در چارچوب گرادیان کاهشی را ارائه می‌نماییم. با توجه به اکتشافی بودن فرآیند طراحی مدل پیشنهادی، بدون تثبیت گام‌های ابتدایی، یافتن گام‌های بعدی ممکن نخواهد بود؛ لذا با برداشتن هر گام، چگونگی اجرای گام بعدی خودنمایی خواهد کرد. در فصل چهارم، پس از معرفی مجموعه‌های داده و معیارهای ارزیابی استفاده شده، آزمایش‌های گوناگونی را طراحی و اجرا کرده و بر اساس نتایج این آزمایش‌ها، مدل پیشنهادی را از ابعاد مختلف سنجیده و سپس آن را با روش‌های پیشین مقایسه می‌کنیم. در فصل پنجم این رساله، از پژوهش انجام شده نتیجه گیری کرده و گام‌هایی را برای ادامه‌ی پژوهش در این مسیر معرفی می‌نماییم.

فصل ۲

پژوهش‌های پیشین

در این فصل پژوهش‌های پیشین در حوزه‌ی پیش‌بینی نرخ کلیک را بررسی و طبقه بندی کرده و نقاط قوت و ضعف آن‌ها را بررسی می‌کنیم. این بررسی را از روش‌های کلاسیک یادگیری ماشین آغاز کرده و سپس با معرفی خانواده‌ای از مدل‌ها به نام ماشین فاکتورگیری و مدل‌های مقتبس از آن، این بررسی را ادامه می‌دهیم؛ سپس به سراغ مدل‌های ژرف رفته و پس از آن، با مقایسه‌ی نهایی این مدل‌ها و بررسی مزایا و معایب هریک از آن‌ها، این فصل را به پایان می‌بریم.

۲-۱ روش‌های کلاسیک

همان‌طور که در فصل قبل بیان کردیم، مساله‌ی پیش‌بینی نرخ کلیک را می‌توان یک مساله‌ی دسته بندی[۲] که از مسائل پایه‌ای یادگیری ماشین است، در نظر گرفته و لذا از روش‌های موجود در ادبیات یادگیری ماشین، برای حل این مساله کمک گرفت.

اولین تلاش‌ها برای حل مساله‌ی پیش‌بینی نرخ کلیک، به استفاده از روش‌های کلاسیک یادگیری ماشین انجامید. هرچند چالش‌هایی که در فصل قبل معرفی کردیم، عملکرد این روش‌ها را محدود و نتایج آن‌ها را تحت تاثیر قرار می‌دادند؛ اما به دلیل نبود روش جایگزین، این روش‌ها در بسیاری از موارد به عنوان تنها روش‌های ممکن در نظر گرفته شده و برای حل مساله‌ی پیش‌بینی نرخ کلیک به کار بسته می‌شدند.

در این بخش به بررسی برخی از این پژوهش‌ها که برخی از آن‌ها قدمت زیادی دارند، می‌پردازیم. ابتدا استفاده از ماشین‌های بردار پشتیبان برای پیش‌بینی نرخ کلیک را بررسی می‌کنیم؛ سپس روش‌های دیگر این دسته از قبیل رگرسیون تکه‌ای خطی و یک مدل رگرسیون بیزی را معرفی می‌کنیم.

[2] Classification

۲-۱-۱ ماشین‌های بردار پشتیبان

در ادبیات یادگیری ماشین کلاسیک، ماشین‌های بردار پشتیبان[۸] سابقه‌ی پژوهشی برجسته و مهمی دارند. ماشین‌های بردار پشتیبان بر اساس در نظر گرفتن ارتباط خطی بین ورودی و خروجی، مساله‌ی رگرسیون را حل می‌کنند. یادگیری پارامترهای ماشین بردار پشتیبان به دلیل استفاده از روش‌های برنامه‌ریزی درجه دوم[۱] و بهره بردن از فرم دوگان[۲] بسیار سریع است. پس از اتمام فرآیند آموزش، مدل ماشین بردار پشتیبان، خروجی مساله را به صورت یک رابطه‌ی خطی ارائه می‌دهد:

$$\hat{y}(x) = w_0 + \sum_{i=1}^{n} w_i x_i, \qquad w_0 \in \mathbb{R}, \quad w \in \mathbb{R}^n \tag{2.1}$$

در این رابطه x ورودی، $\hat{y}$ خروجی، n تعداد ابعاد ورودی و w_0 و w پارامترهای مدل هستند که در فرآیند آموزش تخمین زده می‌شوند. همانطور که از این رابطه مشخص است، عدم پشتیبانی ماشین‌های بردار پشتیبان از ارتباط‌های غیر خطی بین ورودی و خروجی باعث سادگی بیش از حد این مدل می‌شود. در ادبیات یادگیری ماشین کلاسیک، برای حل این مشکل، نسخه‌ی کرنل دار این ماشین‌ها استفاده می‌شود. در ماشین‌های بردار پشتیبان با کرنل چندجمله‌ای درجه دوم، به عبارت بالا یک جمله‌ی دیگر اضافه می‌شود تا پیچیدگی کافی برای حل مساله را به مدل اضافه کند. رابطه‌ی پیش‌بینی ماشین بردار پشتیبان با کرنل چندجمله‌ای درجه دوم به صورت زیر است:

$$\hat{y}(x) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} w'_{i,j} x_i x_j, \qquad w_0 \in \mathbb{R}, \quad w \in \mathbb{R}^n, \quad w' \in \mathbb{R}^{n \times n} \tag{2.2}$$

که w' پارامترهایی هستند که به این مدل اضافه شده‌اند. می‌توان جمله‌ی آخر این عبارت را به تاثیر حضور همزمان دو ویژگی مختلف x_i و x_j در خروجی مدل تعبیر کرد.

همانطور که انتظار می‌رود، این مدل دچار ایراداتی اساسی در طراحی آن است. در صورتی که به پارامترهای این مدل توجه کنیم، متوجه می‌شویم که تعداد پارامترهای این مدل بسیار زیاد است؛ پس برای تکمیل فرآیند یادگیری برای این تعداد پارامتر، نیاز به تعداد بسیار زیادی داده وجود دارد که چنین تعدادی از داده‌ها در دسترس نیست. علاوه بر این، در صورتی که به تفسیر جمله‌ی دوم این عبارت توجه کنیم، متوجه می‌شویم که هرکدام از درایه‌های ماتریس w' تنها زمانی استفاده (و لذا آموزش داده) می‌شوند که هر دو ویژگی مربوطه حاضر باشند. این در حالی است که می‌دانیم بسیاری از جفت ویژگی‌های مجموعه‌های داده در مساله‌ی پیش‌بینی نرخ کلیک، تعداد دفعات بسیار کمی در کنار هم رخ داده و در بسیاری از حالات، هرگز به صورت

[1] Quadratic Programming
[2] Dual Form

همزمان رخ نمی‌دهند. این مشکلات توان یادگیری این مدل را به شدت تهدید کرده و لذا در بسیاری از شرایط، نتایج قابل قبولی ارائه نمی‌دهند.

به دلیل همه‌ی مشکلات گفته شده، ماشین‌های بردار پشتیبان نقش کمتری در پژوهش‌های امروزی در اکثر مساله‌ها، خصوصا مساله‌ی پیش‌بینی نرخ کلیک ایفا می‌کنند.

مدل تکه‌ای خطی[۹]

در ادامه‌ی بررسی روش‌های کلاسیک یادگیری ماشین برای حل مساله‌ی پیش‌بینی نرخ کلیک و رویارویی با چالش ابعاد بالا و غیرخطی بودن روابط بین ویژگی‌ها و خروجی، به بررسی مدل تکه‌ای خطی می‌پردازیم. این مدل قبل از انتشار در مقالات پژوهشی، به مدت قابل توجهی در شرکت علی‌بابا[1] به عنوان روش اصلی حل مساله‌ی پیش‌بینی نرخ کلیک استفاده شده است.

از آن‌جا که جزئیات مساله‌ی مورد بررسی، نیاز به انعطاف غیر خطی را ایجاب می‌کند، لذا محققین شرکت علی‌بابا برای یافتن یک مدل غیرخطی مناسب، تمرکز خود را بر ترکیب مدل‌های خطی به شیوه‌ای که بتوانند در کنار هم عملکرد غیرخطی داشته باشند؛ قرار دادند؛پس یک مدل ساده و عمومی از ترکیب مدل‌های خطی معرفی کردند. در این مدل، نیمی از پارامترها برای تفکیک فضای داده به بخش‌هایی که در هر کدام یک یا ترکیبی از چند مدل جزئی در آن عملکرد قابل قبولی داشته باشند؛ و نیمه‌ی دیگر پارامترها را برای آموزش مدل‌های جزئی در آن بخش‌ها اختصاص داده شده است. رابطه‌ی ریاضی این مدل کلی به صورت زیر است:

$$y = g(\sum_{j=1}^{m} \sigma(u_j^T x)\eta(w_j^T x)) \tag{2.3}$$

که در آن، η تابع تصمیم‌گیری مدل‌های جزئی است. η می‌تواند یک تابع توزیع احتمال دودویی مثل تابع سیگموید[2] باشد. همچنین تابع σ می‌تواند یک تابع وزن دهی چند کلاسه باشد. در ساده‌ترین حالت، تابع سافت مکس[3] می‌تواند این نقش را ایفا کند. بردارهای u_j و w_j پارامترهای مدل هستند و زیر نویس j نشان‌دهنده‌ی شماره‌ی مدلی است که به آن تعلق دارند. ابرپارامتر m تعداد مدل‌های جزئی را تعیین می‌کند که به دلیل جلوگیری از پیچیدگی بیش از حد مدل، اکثرا مقداری نزدیک به ۱۲ دارد. همچنین تابع g یک تابع نرمال ساز احتمال بوده و تنها نقش آن تبدیل تابع به وجود آمده به یک تابع توزیع احتمال معتبر است.

این مدل می‌تواند به وسیله‌ی تابع خطایی نظیر قرینه‌ی درستنمایی[4] و به وسیله‌ی روش‌های گرادیان کاهشی [۱۰] آموزش یابد.

همچنین واضح است که در حالت کلی، و با افزایش تعداد مدل‌های جزئی، این ساختار توانایی مدل کردن

[1] Alibaba
[2] Sigmoid
[3] Softmax
[4] Negative likelihood

هر تابعی را دارد؛ در نتیجه مشکل پیچیدگی بیش از حد مدل، محققین را وادار به افزودن جملات تنظیم به تابع خطای مدل می‌کند. در این تحقیق از دو جمله‌ی خطای زیر استفاده می‌شود:

$$||\theta||_1 = \sum_{i=1}^{d} \sum_{j=1}^{2m} |\theta_{ij}| \tag{2.4}$$

$$||\theta||_{2,1} = \sum_{i=1}^{d} \sqrt{\sum_{j=1}^{2m} \theta_{ij}^2} \tag{2.5}$$

که در آن d تعداد ابعاد داده‌ها و $\theta_{-,j}$ شامل u_j و w_j است.

تنظیم نوع اول برای کاهش کلی تعداد پارامترهای غیر صفر و تنظیم نوع دوم باعث فشردگی میزان پارامترها به منظور کسب واریانس کمتر تعریف شده‌است؛ اما اضافه شدن این دو جمله، باعث می‌شود سطح خطا در فضای پارامترها، سطحی غیر محدب[۱] و غیر نرم[۲] باشد؛ در نتیجه استفاده از روش‌های کاهش گرادیان یا بیشینه‌سازی امید ریاضی[۳] منطقی نیست. برای رفع این اشکال، محققین به روشی مشابه کواسی-نیوتون با حافظه‌ی محدود[۴][۱۱] روی آورده و مدل را بدین طریق آموزش می‌دهند. همچنین در این پژوهش تعدادی تکنیک برای کاهش مصرف حافظه و زمان آموزش ارائه شده که این مدل را برای استفاده در صنعت مناسب می‌سازد.

از مزایای این مدل می‌توان به قابلیت تغییر قسمت‌هایی از مدل و انعطاف پذیری آن، پارامترهای تنک و تفسیر پذیری مناسب اشاره کرد. همچنین از معایب این روش می‌توان به تعداد پارامتر بالا، کندی در زمان آموزش و تفاوت نسبتا جزئی نتایج آن با نتایج روش‌های خطی مثل رگرسیون لجستیک اشاره نمود.

مدل بیزی[۱۲]

در پژوهشی دیگر، محققین شرکت مایکروسافت، برای سیستم جستجوی حمایت شده[۵]ی بینگ[۶]، یک متد پیش‌بینی نرخ کلیک ارائه داده‌اند. خروجی این پژوهش از سال ۲۰۰۹ در مقیاس بالا در جستجوی حمایت شده‌ی بینگ به کار بسته می‌شد.

در این پژوهش، از تابع پرابیت[۷] (تابع تجمعی احتمال توزیع گاوسی)، برای نگاشت[۸] از محور حقیقی، به توزیع احتمال استفاده می‌شود. به همین دلیل به این دسته روش‌ها، رگرسیون پرابیت[۹] گفته می‌شود. دلیل

[1] Convex
[2] Smooth
[3] Expectation Maximization
[4] LBFGS
[5] Sponsored search
[6] Bing
[7] Probit
[8] Mapping
[9] Probit Regression

این نوع نامگذاری، تقابل این دسته از روش‌ها با رگرسیون‌های لجستیک است. همانطور که گفته شد، در رگرسیون لجستیک، از تابع سیگموید برای این نگاشت استفاده می‌شود.

در این روش، با فرض گاوسی و مستقل بودن احتمال پیشین هر یک از پارامترهای مدل، مساله را به صورت یک مساله‌ی رگرسیون خطی در نظر می‌گیریم:

$$p(w) = \prod_i N(w_i|\mu_i, \sigma_i^2) \tag{2.6}$$

حال با استفاده از دو متغیر نهفته[1]ی s و t کار را پیش‌می بریم. متغیر تصادفی s به صورت ضرب داخلی بردار ورودی‌ها در بردار وزن‌ها تعریف شده و به صورت قطعی از روی ورودی‌ها و وزن‌ها قابل مقایسه است. متغیر تصادفی t یک متغیر تصادفی گاوسی با میانگینی برابر با مقدار s و واریانسی مشخص تعریف می‌شود. همچنین، خروجی این مدل (y) به وسیله‌ی یک تابع آستانه مثل تابع علامت روی متغیر t به دست می‌آید.

$$s = w^T x \tag{2.7}$$

$$t \sim N(s, \sigma^2) \tag{2.8}$$

$$y = sign(t) \tag{2.9}$$

سپس به کمک دو متغیر تصادفی تعریف شده، توزیع احتمال شرطی خروجی نسبت به ورودی را این‌گونه فاکتورگیری می‌کنیم:

$$p(y, t, s, w|x) = p(y|t)p(t|s)p(s|x, w)p(w) \tag{2.10}$$

به دلیل غیر قابل محاسبه بودن توزیع پسین برای وزن‌ها، استفاده از این روابط برای محاسبه‌ی مستقیم مقادیر وزن‌ها ممکن نیست؛ پس با استفاده از الگوریتم‌های پیام‌رسانی[2] و تخمین توزیع پسین با توزیع گاوسی، مقادیر وزن‌ها قابل آموزش می‌شوند.

در این پژوهش، اندازه‌ی گام به روز رسانی مقادیر پارامترها را در طول زمان کاهش داده و بدین طریق، آموزش مدل را تسریع می‌کنند. همچنین فرآیند اکتشاف[3] و بهره برداری[4] نیز، بدین وسیله مدل می‌شود که برای نمونه‌هایی با اطمینان بالا (واریانس پایین) عمل بهره برداری و برای نمونه‌هایی با اطمینان پایین (واریانس

[1] Latent
[2] Message passing
[3] Expolration
[4] Exploitation

بالا) عمل اکتشاف انجام داده می‌شود؛ به همین دلیل این روش نیز مانند بقیه‌ی روش‌ها، از مشکل شروع سرد رنج می‌برد.

نتایج عمده‌ی روش‌هایی که تا اینجا معرفی کردیم، به دلیل وجود چالش‌هایی که در فصل قبل مطرح شد، چندان قابل قبول نیستند؛ لذا از سال ۲۰۱۰ به بعد، توجه بخش عمده‌ای از پژوهشگران به سمت روش‌هایی تحت عنوان خانواده‌ی ماشین‌های فاکتورگیری جلب شد.

۲-۱-۲ ماشین‌های فاکتورگیری

در این بخش به بررسی پژوهش‌های خانواده‌ی ماشین‌های فاکتورگیری می‌پردازیم. ایده‌ی اصلی استفاده از ماشین‌های فاکتورگیری، استفاده از شیوه‌ی به خصوصی از تنظیم است که باعث می‌شود مدل، قابلیت یادگیری خواص ترکیبی بین ویژگی‌های مختلف و متعدد ورودی را با تعداد محدودی پارامتر داشته باشد. در ادبیات ماشین‌های فاکتورگیری، به این خواص ترکیبی، تعامل[1] بین ویژگی‌ها گفته می‌شود. در این بخش چند پژوهش در حوزه‌ی ماشین‌های فاکتورگیری از جمله پژوهشی که اولین بار از این ایده برای پیش‌بینی نرخ کلیک استفاده کرده است را بررسی می‌کنیم.

ایده‌ی فیلدها و شیوه‌ی نگرش به داده‌ها در ماشین‌های فاکتورگیری

در همه‌ی پژوهش‌های این دسته، نگرش خاصی به داده‌ها وجود دارد که در این بخش آن را معرفی می‌کنیم. در اغلب مجموعه‌های داده‌ی موجود در ادبیات تخمین نرخ کلیک و همچنین سیستم‌های پیشنهاد دهنده، همه یا اکثر ویژگی‌ها به صورت دسته‌ای[2] هستند. مدل‌های یادگیری ماشین برای برخورد مناسب با این نوع ویژگی‌ها، از روش‌های مختلفی از جمله کدگذاری یک از k[3] استفاده می‌کنند.

در روش کدگذاری ۱ از k، ابتدا همه‌ی مقادیر مختلف این ویژگی دسته‌ای لیست شده، سپس به هر کدام یک شماره یا اندیس تخصیص داده می‌شود؛ سپس برای نمایش دادن حالتی که ویژگی دسته‌ای مقدار nام را داشته باشد، برداری به اندازه‌ی k (تعداد حالات ویژگی دسته‌ای) ایجاد شده و همه‌ی مقادیر آن (بجز خانه‌ی اندیس nام) صفر قرار داده می‌شود و در خانه‌ی اندیس nام، مقدار ۱ قرار داده می‌شود؛ پس در هر حالت، تنها یکی از درایه‌های این بردار برابر یک بوده و بقیه‌ی درایه‌ها مقدار صفر دارند؛ به همین دلیل به این بردار، بردار تک داغ[4] هم گفته می‌شود.

در روش‌های ماشین فاکتورگیری، به هر یک از ویژگی‌های دسته‌ای و بردارهای مربوط به آن‌ها، یک فیلد[5] گفته می‌شود. همچنین به هر یک از درایه‌های این بردارها، یک ویژگی باینری[6] گفته می‌شود. در این مدل‌ها

[1] Interaction
[2] Categorical
[3] One of k coding
[4] One hot vector
[5] Field
[6] Binary feature

پس از کدگذاری همه‌ی فیلدهای موجود در داده‌ها، بردارهای تک داغ ساخته شده را به هم چسبانده و یک بردار چند داغ[1] ساخته می‌شود. این بردار به صورت مستقیم به عنوان ورودی مدل‌های ماشین فاکتورگیری استفاده می‌شود. در اغلب مجموعه‌های داده‌ی در دسترس، تعداد فیلدها (f) بین ۱۰ تا ۵۰ بوده و تعداد ویژگی‌های باینری (n) بین چند ده هزار تا چند ده میلیون است؛ لذا ورودی ماشین‌های فاکتورگیری، بردارهایی به طول چند میلیون هستند که تنها چند ده درایه‌ی غیر صفر دارند.

ماشین‌های فاکتورگیری ساده[۱۳]

خانواده‌ی بزرگی از مدل‌هایی که برای محاسبه‌ی نرخ کلیک استفاده می‌شوند، ماشین‌های فاکتورگیری[2] و نسخه‌های پیشرفته‌ی آن‌ها هستند. تحقیقات بسیاری با پیاده‌سازی و پیشنهاد انواع جدید این خانواده، مساله‌ی پیش‌بینی نرخ کلیک را حل کرده و بهترین نتایج توسط همین تحقیقات ارائه شده‌اند.

ایده‌ی اصلی ماشین‌های فاکتورگیری، همان‌طور که از نام آن‌ها مشخص است، عمل فاکتورگیری ماتریسی است. عمل فاکتورگیری زمانی استفاده می‌شود که نیاز به تخمین زدن یک ماتریس وجود داشته باشد، اما به دلیل ابعاد بالای این ماتریس، قابلیت یادگیری همه‌ی درایه‌های آن برای مدل موجود نباشد. مثلا ماشین بردار پشتیبان با کرنل چندجمله‌ای درجه دوم که آن را در بخش‌های قبل معرفی کردیم، ماتریس w' که مشخص کننده‌ی وزن جمله‌های مرتبه دوم است، دقیقا همین شرایط را داراست؛ پس در پژوهشی که اولین بار ماشین‌های فاکتورگیری را معرفی کرد، سراغ همین ماتریس رفته و عمل فاکتورگیری را روی آن انجام دادند. در ماشین فاکتورگیری، به جای این که فرض کنیم همه‌ی درایه‌های این ماتریس پارامترهای مستقل و قابل یادگیری هستند، این ماتریس را حاصل ضرب یک ماتریس با ابعاد کمتر در ترانهاده‌ی خودش فرض کرده و لذا رتبه‌ی ماتریس w' را کاهش می‌دهیم:

$$w' = v.v^T, \quad v \in \mathbb{R}^{n \times k} \tag{2.11}$$

که در آن k بعد تعبیه بوده و مقدار کمی (حدود ۱۰) دارد؛ پس ماتریس w' از روی ماتریس v ساخته شده و در نتیجه مشکلات ذکر شده در ماشین بردار پشتیبان با کرنل چندجمله‌ای درجه دوم در آن وجود ندارد. عبارت کامل رابطه‌ی ماشین‌های فاکتورگیری به این صورت است:

$$\hat{y}(x) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} w'_{i,j} x_i x_j, \quad w'_{i,j} = \sum_{l=1}^{k} v_{i,l} v_{j,l},\ v_i \in \mathbb{R}^k \tag{2.12}$$

تعبیر دیگری که می‌توانیم از این روابط داشته باشیم، عملکرد مناسب ماشین‌های فاکتورگیری را بهتر نمایان

[1] Multi hot vector
[2] Factorization Machines

می‌کند. می‌توانیم ماتریس v را به شکل یک جدول تعبیه[1] در نظر بگیریم؛ در نتیجه به ازای هر فیلد، تنها یکی از سطرهای این جدول انتخاب می‌شود. (بقیه‌ی سطرها به دلیل این‌که x_i مربوطه صفر است، تاثیری در خروجی ندارند.) در نهایت، حاصل ضرب داخلی بردارهای تعبیه‌ی همه‌ی فیلدها دو به دو محاسبه شده و نتایج آن با نتایج جمله‌ی خطی جمع می‌شود. حاصل هر یک از این ضرب‌های داخلی، به نام تعامل بین دو ویژگی نیز شناخته می‌شود. در نهایت با اعمال تابع سیگموید، عدد حاصل به توزیع احتمال کلیک تبدیل می‌شود.

همان طور که گفته شد، در ماشین‌های فاکتورگیری علاوه بر ارتباط خطی بین خروجی و همه‌ی ابعاد ورودی، تاثیر تعامل بین ابعاد ورودی نیز در خروجی در نظر گرفته می‌شود؛ لذا پیچیدگی ماشین‌های فاکتورگیری از مدل‌های رگرسیون خطی مثل ماشین‌های بردار پشتیبان یا رگرسیون لجستیک بیشتر است و قادر به مدل کردن خانواده‌ی بزرگتری از توابع هستند.

یکی از مهمترین فواید عدم استقلال درایه‌های ماتریس w' از یکدیگر، در زمان مواجهه با داده‌های تنک[2] مشخص می‌شود. خصوصا در مساله‌ی پیش‌بینی نرخ کلیک که تعداد ابعاد داده بسیار زیاد بوده ولی اکثر ویژگی‌های داده به ندرت فعال (غیر صفر) هستند. اگر در این‌گونه مسائل همه‌ی ضرایب تعامل بین ویژگی‌ها را مستقل در نظر بگیریم، به تعداد بسیار زیاد و گاها غیر قابل دسترس داده نیاز خواهیم داشت. در مقابل، هنگام استفاده از ماشین‌های فاکتورگیری، به دلیل کاهش تعداد پارامترهای قابل یادگیری، با استفاده از تعداد داده‌ی کمتر، نتایج تعمیم‌پذیرتری قابل دستیابی هستند.

علاوه بر این، در صورتی که در داده‌های آموزشی، یک جفت ویژگی به صورت همزمان رخ نداده باشند، یادگیری وزن مربوط به آن‌ها توسط ماشین بردار پشتیبان با کرنل چندجمله‌ای درجه دوم غیر ممکن است. در حالی که در ماشین‌های فاکتورگیری، در صورتی که این دو ویژگی به تعداد قابل قبول به صورت مجزا مشاهده شوند، بردارهای تعبیه‌ی مربوط به آن‌ها توسط ماشین فاکتورگیری یاد گرفته شده و لذا محاسبه‌ی تعامل این دو ویژگی با وجود این که قبلا با هم مشاهده نشده‌اند، ممکن خواهد بود. این مزیت ماشین‌های فاکتورگیری قابلیت تعمیم آن‌ها را افزایش داده و آن‌ها را تا حدودی در مقابل چالش شروع سرد مقاوم می‌کند.

ماشین‌های فاکتورگیری ساده، عملکرد قابل توجهی روی مجموعه‌های داده‌ی مربوط به نرخ کلیک ارائه کرده و در صنعت نیز مورد استفاده قرار گرفتند؛ اما به دلیل سادگی زیاد، تعمیم آن‌ها از جهات مختلف در دستور کار پژوهشگران قرار گرفت و روش‌های متعددی برای تعمیم آن‌ها معرفی شدند. در ادامه به بررسی برخی از این روش‌ها می‌پردازیم.

ماشین‌های فاکتورگیری آگاه از فیلد[۱۴، ۱۵]

ماشین‌های فاکتورگیری ساده، برای محاسبه‌ی تعامل بین دو ویژگی، از عمل ضرب داخلی بین بردار تعبیه‌ی این دو ویژگی استفاده می‌کنند. در نتیجه برای محاسبه‌ی تعامل یک ویژگی از فیلد اول، با یک ویژگی از فیلدهای دوم یا سوم، از بردار تعبیه‌ی یکسانی استفاده شود. محققینی که ماشین فاکتورگیری آگاه از فیلد را

[1] Embedding Table
[2] Sparse

معرفی کردند، ادعا می‌کنند تعامل بین فیلدهای اول و دوم، کاملا از تعامل بین فیلدهای اول و سوم مجزا بوده و می‌توان برای آن‌ها از بردارهای تعبیه‌ی متفاوت استفاده کرد.

این ادعای این پژوهش را می‌توان به صورت دیگر نیز بیان کرد. فرض کنید فضای تعبیه‌ی A برای ویژگی‌های فیلد اول و فضای تعبیه‌ی B و C به ترتیب برای ویژگی‌های فیلد دوم و سوم باشند. در صورتی که پارامترهای موجود در A برای محاسبه‌ی تعامل با بردارهای B یاد گرفته شوند، یعنی فضای A به طریقی ایجاد شده است که تفاوت‌های مربوط به ویژگی‌های فیلد دوم را در نظر گرفته است ولی تفاوت‌های مربوط به ویژگی‌های فیلد سوم از آن حذف شده است؛ پس تعامل محاسبه شده بین A و C نمی‌تواند تمامی اطلاعات ممکن را دارا باشد. در نتیجه لازم است برای هر فیلد، به تعداد $f-1$ فضای تعبیه در نظر گرفته و تعامل بین ویژگی‌های هر جفت فیلد را، در فضای مربوط به آن جفت فیلد محاسبه کنیم.

رابطه‌ی پیش‌بینی نهایی ماشین آگاه از فیلد، به صورت زیر است:

$$\hat{y}_{FFM}(x) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} x_i x_j < v_{i,F_j}, v_{j,F_i} > \qquad (2.13)$$

که در آن، v_{i,F_j} بردار تعبیه‌ی ویژگی iام در مواجهه با ویژگی‌های فیلد مربوط به ویژگی jام بوده و عملگر $< . >$ ضرب داخلی بین دو بردار را محاسبه می‌کند.

همان طور که واضح است که این تغییر باعث افزایش بسیار زیاد تعداد پارامترهای این مدل می‌شود؛ در نتیجه ماشین‌های فاکتورگیری آگاه از فیلد به دلیل تعداد پارامترهای بالا، در مقابل چالش‌هایی از قبیل شروع سرد و سرعت آموزش، چندان موفق نیستند.

ماشین‌های فاکتورگیری با فیلدهای وزن‌دار[۱۶]

در ماشین‌های فاکتورگیری آگاه از فیلد، از آن‌جا که برای هر جفت فیلد، یک دسته بردار تعبیه شده در نظر گرفته می‌شود؛ تعداد پارامترهای مدل بسیار زیاد بوده و این امر باعث بروز مشکلاتی از جمله افزایش زمان آموزش و همچنین بیشتر شدن خطر بیش برازش می‌شود؛ پس محققین به دنبال یافتن راهی برای کاهش تعداد پارامترها با حفظ پیچیدگی مشکل گشته و در نتیجه ماشین‌های فاکتورگیری با فیلدهای وزن‌دار معرفی شدند.

در ماشین‌های فاکتورگیری با فیلدهای وزن‌دار، به این نکته که میانگین میزان تعامل بین جفت‌های مختلف از فیلدها، بسیار متفاوت است؛ توجه ویژه‌ای شده است. به عنوان مثال، اکثر تعاملات بین ویژگی‌های فیلد تبلیغ کننده و فیلد ناشر، میزان چشم‌گیری دارند؛ در حالی که تعاملات بین ویژگی‌های فیلد ساعت و فیلد روز هفته، میزان قابل توجهی ندارند. که این تفاوت با توجه به مفهوم این فیلدها، کاملا منطقی به نظر می‌رسد؛ اما در ماشین‌های فاکتورگیری آگاه از فیلد، چنین تفاوتی مدل نمی‌شود؛ لذا محققین در ماشین‌های فاکتورگیری با فیلدهای وزن‌دار، به آن توجه کرده و این تفاوت را به صورت صریح وارد محاسبات کردند.

رابطه‌ی پیش‌بینی نهایی ماشین فاکتورگیری با فیلد وزن‌دار، به صورت زیر است:

$$\hat{y}_{FwFM}(x) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} x_i x_j < v_i, v_j > r_{F_i,F_j} \tag{2.14}$$

در این رابطه، r_{F_i,F_j} نقش مدل کردن قدرت کلی تعاملات بین فیلد i ام و j ام را ایفا می‌کند. علاوه بر این، یک تفاوت دیگر بین ماشین‌های آگاه از فیلد و ماشین‌های با فیلدهای وزن‌دار وجود دارد. این تفاوت به تعداد بردارهای تعبیه شده‌ی مربوط به هر ویژگی باز می‌گردد. در ماشین‌های فاکتورگیری آگاه از فیلد، برای هر ویژگی، به تعداد فیلدهای دیگر بردار تعبیه شده استفاده می‌شود؛ ولی در ماشین‌های فاکتورگیری با فیلدهای وزن‌دار، برای هر ویژگی، تنها یک بردار تعبیه‌شده استفاده می‌شود و تفاوت قدرت کلی تعاملات بین فیلدها توسط وزن‌های فیلدها (r) مدل می‌شود.

لذا ماشین‌های فاکتورگیری با فیلدهای وزن‌دار، می‌توانند با تعداد پارامترهای بسیار کمتر، عملکرد نسبتا یکسانی با ماشین‌های فاکتورگیری آگاه از فیلد کسب کنند. در صورتی که تعداد پارامترهای استفاده شده در دو مدل یکسان در نظر گرفته شود، عملکرد ماشین‌های با فیلد وزن دار، به صورت محسوسی بهتر می‌شود.

پژوهشگران این مدل با محاسبه‌ی همبستگی بین وزن‌های آموخته شده برای فیلدها (r) با اطلاعات مشترک[1] بین هر زوج فیلد و احتمال کلیک (خروجی مدل)، موفقیت آن را نسبت به مدل‌های پیشین تایید کردند.

با وجود مزایای گفته شده، ماشین‌های فاکتورگیری با فیلدهای وزن‌دار به دلیل سادگی، توان مدل‌سازی محدودی دارند؛ پس محققین به دنبال راهکارهای دیگر برای حل مساله‌ی تخمین نرخ کلیک گشته و پیشرفت‌های دیگری را کسب کردند.

ماشین‌های فاکتورگیری تنک

محققین، پس از بررسی نمونه‌های مختلفی از ماشین‌های فاکتورگیری، متوجه شدند در اکثر نسخه‌های استفاده شده از این خانواده مدل، تعداد پارامترهای آموخته شده بسیار زیاد بوده و به همین دلیل، خطای این مدل‌ها همچنان قابل توجه است؛ لذا اقدام به بررسی راه‌هایی کردند که بتوان به کمک آن‌ها، تنک بودن مدل را تضمین کرده و در نتیجه به خطای کمتر و تفسیر پذیری بیشتری دست یابند. یکی از این اقدامات، ماشین‌های فاکتورگیری تنک است. برای درک بهتر این مدل، بهتر است ابتدا ماشین‌های فاکتورگیری بیزی را بررسی کنیم.

- ماشین‌های فاکتورگیری بیزی[۱۷]

در ادبیات سیستم‌های پیشنهاد دهنده، بسیاری از مدل‌ها به دلیل حجم بالای محاسبات، پاسخگو نیستند؛ در نتیجه تحقیقات زیادی در این زمینه برای یافتن مدل‌هایی با پیچیدگی محاسباتی کمتر اختصاص یافته است. یکی از این تحقیقات، ماشین‌های فاکتورگیری بیزی است. چون آموزش ماشین‌های فاکتورگیری ساده، به پیچیدگی محاسباتی بالایی نیاز دارد؛ همچنین مقدار k بهینه،

[1] Mutual information

جز با آزمون و خطا قابل محاسبه نیست؛ برای آموزش یک مدل مناسب از خانواده‌ی ماشین‌های فاکتورگیری، به زمان محاسبه‌ی بسیار طولانی نیاز است.

این در حالی است که می‌توان عمل فاکتورگیری را، به جای روش‌های مبتنی بر گرادیان، به وسیله‌ی نمونه برداری گیبس[1] انجام داد. همچنین در این روش‌ها، می‌توان با فرض توزیع پیشین برای هر یک از پارامترها، عمل تنظیم را در این مدل‌ها بهبود بخشید؛ پس ماشین‌های فاکتورگیری بیزی، با استفاده از توزیع پیشین برای پارامترهای مدل و همچنین استفاده از نمونه برداری گیبس، با کاهش چشمگیر پیچیدگی محاسباتی و همچنین حفظ عملکرد نهایی (بهبود جزئی) ارائه شدند.

در ماشین‌های فاکتورگیری بیزی، برای همه‌ی پارامترهای قابل یادگیری مدل، توزیع پیشین گاوسی با پارامترهای غیر ثابت در نظر گرفته می‌شود. این پارامترهای غیر ثابت را، ابرپارامترهای مدل می‌نامیم. همچنین برای این ابرپارامترها، توزیع پیشین در نظر گرفته و پارامترهای این توزیع‌های پیشین را، ابر پیشین[2] می‌نامیم. ابر پیشین‌ها عملا توزیع پیشین برای پارامترهای توزیع پیشینِ پارامترهای مدل هستند. به این تکنیک، ابر پیشین‌های سلسله مراتبی[3] گفته می‌شود. از فواید استفاده از این تکنیک، می‌توان به عدم نیاز به جستجوی توری[4] و همچنین تنظیم بیشتر مدل اشاره کرد. به عنوان میانگین توزیع گاوسی پارامترها، یک متغیر تصادفی با توزیع گاوسی و به عنوان عکس واریانس توزیع پارامترها، یک متغیر تصادفی با توزیع گاما در نظر گرفته می‌شود.

به دلیل پیچیدگی بیش از حد، محاسبه‌ی درستنمایی برای خروجی این مدل، قابل انجام نیست؛ پس از طریق نمونه برداری گیبس، پارامترها و هایپر پارامترهای مدل آموخته می‌شوند. به دلیل پیاده سازی خاص، آموزش این مدل به محاسبات خطی نسبت به k نیاز داشته و به مراتب سریع‌تر از ماشین‌های فاکتورگیری عادی است. این مدل علاوه بر سرعت، از پیچیدگی بیشتری نسبت به ماشین‌های فاکتورگیری عادی برخوردار بوده و در نتیجه در دنیای واقعی قابلیت استفاده‌ی بیشتری دارند.

زمانی که ماشین‌های فاکتورگیری بیزی، در ادبیات پیش‌بینی نرخ کلیک به کار گرفته شدند، محققین دریافتند تعداد زیادی از پارامترهای این مدل، مقادیر غیر صفر به خود گرفته و این اتفاق باعث عدم تفسیر پذیری و همچنین عدم تطابق خروجی این مدل با خروجی مورد انتظار از آن می‌شود. همچنین همانطور که گفته شد، در ماشین فاکتورگیری بیزی، ابر پیشین گاوسی برای میانگین‌ها و ابر پیشین گاما برای عکس واریانس‌ها در نظر گرفته می‌شود؛ اما توزیع گاوسی، به دلیل محدودیت و تنک بودن شدید داده‌های پیش‌بینی نرخ کلیک، برای این مسائل چندان مناسب نیست. محققین دریافتند در صورت استفاده از توزیع لاپلاس برای میانگین، به دلیل احتمال بیشتر صفر بودن و همچنین داشتن دنباله‌ی بزرگتر، امکان تطابق بیشتر با داده‌های تنک این مسائل افزایش می‌یابد.

در ماشین‌های فاکتورگیری تنک[۱۸]، با در نظر گرفتن این که تنها حدود ۰٫۱۵ درصد از مقادیر ویژگی‌های مجموعه‌های داده‌ی مورد استفاده غیر صفر هستند، فرض توزیع پیشین گاوسی را برای پارامترهای مدل رد

[1] Gibbs Sampling
[2] Hyperprior
[3] Hierarchical hyperpriors
[4] Grid search

کرده و به جای آن، از توزیع لاپلاس استفاده می‌کنند. توزیع لاپلاس، دارای دنباله‌ی سنگین‌تری نسبت به توزیع گاوسی می‌باشد، ولی احتمال تولید صفر توسط این توزیع، به مراتب بیشتر از توزیع گاوسی است.

به دلیل ناهموار[1] بودن توزیع لاپلاس، استنباط بیزی در مورد ماشین‌های فاکتورگیری تنک غیر قابل انجام است؛ لذا آن را به وسیله‌ی مخلوط مقیاس‌شده[2]ی چگالی توزیع‌های گاوسی و نمایی در نظر گرفته و سپس، با استفاده از زنجیره‌ی مارکوف مونت کارلو[3] نسبت به استنباط روی آن اقدام می‌کنند.

یکی از فواید استفاده از مدل بیزی، این است که به جای پیش‌بینی صرف مقدار نرخ کلیک، برای آن چگالی توزیع محاسبه می‌شود. با استفاده از این چگالی توزیع، می‌توان مواقعی که مدل با اطمینان تصمیم می‌گیرد و مواقعی که مدل اطمینان خاصی ندارد را از هم تمییز داده و از این تمایز، در تصمیم گیری بین اکتشاف یا استفاده[4] بهره جست. به عبارت دیگر، مدل بیزی امکان رویارویی بهتر با چالش شروع سرد را فراهم می‌سازد.

ماشین فاکتورگیری با توجه[۱۹]

در سال‌های اخیر، استفاده از مفهوم توجه[5] در شبکه‌های عصبی، باعث پیشرفت قابل توجهی در نتایج آن‌ها شده و به همین دلیل، در بسیاری از وظایف یادگیری ماشین، از پردازش زبان طبیعی گرفته تا پردازش تصاویر، به صورت گسترده مورد استفاده قرار گرفتند. از طرفی در مساله‌ی پیش‌بینی نرخ کلیک، نیاز به اعمال تمایز میان ویژگی‌های مرتبه بالاتر از نظر میزان اهمیت احساس می‌شد؛ پس پژوهشگران در یک پژوهش، اقدام به استفاده از این مفهوم و ترکیب آن با ماشین‌های فاکتورگیری کرده و نتایج قابل قبولی نیز گرفتند. در این بخش، به معرفی مدل ماشین فاکتورگیری با توجه پرداخته و جزئیات آن را بررسی می‌کنیم.

طبق مشاهدات قبلی، برخی از ویژگی‌های مرتبه دوم در ماشین‌های فاکتورگیری، از برخی دیگر اهمیت بسیار بیشتری داشته و برخی از آن‌ها تقریبا هیچ ارتباطی با متغیر هدف ندارند؛ لذا در مدل ماشین فاکتورگیری ساده، که تمایزی بین این دو دسته وجود ندارد، امکان کم توجهی به ویژگی‌های مرتبه دوم مهم و توجه بیش از حد به ویژگی‌های مرتبه دوم نه چندان مهم (نویز) وجود دارد. این امر باعث تشدید مشکل بیش‌برازش در این مدل‌ها می‌شود. همچنین به دلیل تعداد بالای این ویژگی‌ها، بررسی و ایجاد تمایز بین آن‌ها به صورت دستی ممکن نیست؛ در نتیجه این نیاز احساس می‌شود که این تفاوت‌ها به صورت خودکار و از روی داده‌ها استخراج شوند. در ماشین‌های فاکتورگیری با فیلدهای وزن‌دار، برای حل این مشکل از وزن‌دهی به تعامل بین فیلدها استفاده می‌شد؛ اما این برای مقابله با نویز و بیش‌برازش کافی نیست و در نتیجه در ماشین فاکتورگیری با توجه از مکانیزم توجه برای این امر استفاده می‌شود.

ماشین‌های فاکتورگیری با توجه، دو تفاوت عمده با ماشین‌های فاکتورگیری ساده دارند: ۱- استفاده از ضرب درایه به درایه به جای ضرب نقطه‌ای برای استخراج ویژگی‌های مرتبه دوم؛ ۲- استفاده از ماژول توجه برای

[1] Non-smooth
[2] Scale mixture
[3] Markov Chain Monte Carlo
[4] Explore / Exploit
[5] Attention

ایجاد تمایز بین ویژگی‌های مرتبه دوم. در این بخش این دو تمایز را توضیح می‌دهیم.

در ماشین فاکتورگیری با توجه، ابتدا بردارهای تعبیه‌شده‌ی ویژگی‌های مرتبه دوم طبق فرمول زیر محاسبه می‌شوند:

$$\mathcal{E}_{i,j} = (v_i \odot v_j)x_i x_j \tag{2.15}$$

که در آن عملگر $\odot$ نشان‌دهنده‌ی ضرب درایه به درایه است. مقادیر توجه، از طریق اعمال یک شبکه‌ی عصبی تک لایه روی این بردارهای تعبیه‌شده محاسبه می‌شوند:

$$a_{i,j} = Softmax_{i,j}\{\mathbf{h}^T ReLU(\mathbf{W}\mathcal{E}_{i,j} + \mathbf{b})\} \tag{2.16}$$

در که در آن عملگر $Softmax_{i,j}\{.\}$ بین همه‌ی جملات دارای i و j مختلف اعمال می‌شود؛ در نتیجه مجموع $a_{i,j}$ ها همیشه برابر ۱ است.

سپس این بردارها با استفاده از مکانیزم توجه با هم ترکیب شده و خروجی نهایی ماشین فاکتورگیری با توجه، با اضافه شدن جملات مربوط به رگرسیون خطی، به این صورت تشکیل می‌شود:

$$\hat{y}_{AFM}(x) = w_0 + \sum_{i=1}^{n} w_i x_i + \mathbf{P}^T \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} a_{i,j}\mathcal{E}_{i,j} \tag{2.17}$$

همان طور که از روابط اخیر مشخص است، شیوه‌ی محاسبه‌ی تعامل در این مدل با روش‌های ماشین فاکتورگیری متفاوت بوده و به جای محاسبه‌ی تعامل‌های تک بعدی، ابتدا برای هر جفت فیلد، یک بردار تعامل محاسبه شده و سپس از طریق یک ماتریس، بردارها به فضای تک بعدی خروجی نگاشت می‌شوند. این تفاوت باعث افزایش پیچیدگی این روش و در نتیجه پیشرفت عملکرد در زمان رویارویی با داده‌های حجیم می‌شود؛ اما در مقابل در مواجهه با داده‌های تنک یا شرایط شروع سرد، ممکن است این روش دچار مشکل شده و از بیش‌برازش رنج ببرد.

در نهایت، این مدل بر اساس میانگین مربعات خطا و از طریق روش گرادیان کاهشی تصادفی، بهینه‌سازی شده و از تکنیک حذف تصادفی[1][۲۰] برای تنظیم پارامترهای پیش‌بینی و تنظیم مرتبه دوم[2][۲۱] برای پارامترهای مکانیزم توجه استفاده می‌شود.

[1] dropout
[2] L2-Regularization

۲-۱-۳ روش‌های ژرف

با پیشرفت یادگیری ژرف، امروزه بهترین نتایج در بسیاری از مسائل در زمینه‌ی یادگیری ماشین، توسط مدل‌های ژرف کسب می‌شود. به دلیل قابلیت به کار گیری این مدل‌ها در بسیاری از مسائل و همچنین کسب نتایج قابل قبول این دسته از مدل‌ها، استفاده از آن‌ها در زمینه‌ی تبلیغات نمایشی نیز در حال افزایش است.[۲۲] در این بخش به بررسی چند نمونه از پژوهش‌هایی که از روش‌های یادگیری ژرف در ادبیات پیش‌بینی نرخ کلیک استفاده کرده‌اند می‌پردازیم.

مدل ژرف پیش‌بینی نرخ کلیک[۲۳]

مدل ژرف پیش‌بینی نرخ کلیک یکی از مدل‌هایی است که از تکنیک‌های یادگیری ژرف بر روی مساله‌ی پیش‌بینی نرخ کلیک استفاده کرده است. در این مدل، ویژگی‌های ورودی به دو دسته‌ی ویژگی‌های بصری تصویر بنر و و ویژگی‌های پایه تقسیم می‌شوند.

ویژگی‌های بصری تصویر حاوی مقادیر روشنایی پیکسل‌ها و ویژگی‌های پایه حاوی اطلاعاتی مثل: محل نمایش تبلیغ، کمپین تبلیغ، گروه مخاطب تبلیغ، گروه تبلیغ و مشخصات پایه‌ی کاربر (مانند سن و جنسیت) است. در این پژوهش، ویژگی‌های بصری توسط یک شبکه‌ی عصبی کانوولوشنی و ویژگی‌های پایه توسط یک شبکه عصبی تماما متصل[1] کد می‌شود؛ سپس ویژگی‌های کد شده به وسیله‌ی یک شبکه عصبی تماما متصل دیگر پردازش شده و از آن نرخ کلیک یا احتمال کلیک کاربر بر روی این بنر، به دست می‌آید.

در فرآیند آموزش این مدل، از الگوریتم گرادیان کاهشی برای کمینه کردن مقدار خطای لگاریتمی بهره جسته می‌شود. در کنار تابع هزینه، از تنظیم مرتبه دوم برای بهبود تعمیم پذیری این مدل استفاده می‌شود.

همانطور که اشاره شد، مدل ژرف پیش‌بینی نرخ کلیک شامل سه بخش است:

- شبکه‌ی کانوولوشنی

همانطور که از نام آن مشخص است، شبکه‌ی کانوولوشنی یک شبکه عصبی کانوولوشنی ژرف است. معماری این شبکه از شبکه‌ی معروف رز نت[2][۲۴] الهام گرفته شده و شامل ۱۷ لایه‌ی کانوولوشنی می‌باشد.

لایه‌ی اول این شبکه‌ی کانوولوشنی دارای کرنل‌های ۵ در ۵ و بقیه لایه‌های این شبکه از کرنل‌های ۳ در ۳ تشکیل شده‌اند. این بخش از شبکه قبل از آموزش کلی شبکه، توسط تصاویر بنرها و دسته‌ی بنرها (به عنوان برچسب) پیش‌آموزش[3] می‌شود. برای این منظور از دو لایه‌ی تماما متصل اضافی در انتهای این شبکه استفاده می‌شود که ویژگی‌های استخراج شده توسط لایه‌های کانوولوشنی را به برچسب (دسته‌ی بنر) تبدیل کند. این دو لایه پس از اتمام پیش‌آموزش حذف می‌شوند.

[1] Fully Connected Neural Network
[2] ResNet
[3] Pretrain

- شبکه‌ی پایه

این بخش از شبکه، شامل تنها یک لایه‌ی تماما متصل بوده و برای کاهش ابعاد بردار ویژگی‌های ساده به کار می‌رود. این لایه دارای ۱۲۸ نورون با تابع فعالساز واحد خطی یکسو کننده (رلو)[1][۲۵] بوده و فضای تنک بردار ویژگی‌های ساده را به یک بردار چگال[2] تبدیل می‌کند. می‌توان گفت عملکرد این لایه همانند استفاده از بردارهای تعبیه[3][۲۶] برای تبدیل ویژگی‌های دسته‌ای به بردارهای چگال در روش‌هایی که پیش‌تر معرفی کردیم است.

- شبکه‌ی ترکیبی

خروجی شبکه‌های پایه و کانوولوشنی پس از چسبانده شدن به هم و عبور آن از یک لایه‌ی نرمال‌سازی دسته‌ای[4][۲۷]، به عنوان ورودی شبکه‌ی ترکیبی استفاده می‌شوند. این شبکه دارای دو لایه با ۲۵۶ نورون و یک لایه با تنها یک نورون می‌باشد. خروجی لایه‌های اول به وسیله‌ی تابع فعال‌ساز رلو و لایه‌ی سوم با استفاده از تابع فعال‌ساز سیگموید به فضای غیر خطی منتقل می‌شوند.

برای کاهش زمان آموزش این مدل، دو تکنیک استفاده می‌شوند. اول استفاده از یک پیاده‌سازی سریع برای لایه‌ی تماما متصل تنک است. به دلیل استفاده از کدگذاری ۱ از k و همچنین درهمسازی ویژگی‌ها[5]، دارای تعداد زیادی ویژگی است که در هر نمونه، غالب آن‌ها برابر صفر هستند. استفاده از این دانش در پیاده‌سازی لایه‌ی تماما متصل اول در شبکه‌ی پایه باعث بهبود چشمگیر در سرعت آموزش مدل می‌شود.

تکنیک دیگر استفاده شده در این پژوهش، نمونه برداری مناسب برای بهره‌گیری بیشتر از حافظه می‌باشد. در مجموعه‌داده‌های استفاده شده در این پژوهش، تعداد زیادی تصویر یکسان وجود دارد؛ پس می‌توان با استفاده از این دانش، نمونه برداری قبل از انجام هر گام از الگوریتم گرادیان کاهشی را به نحوی تغییر داد که تعداد محدودی تصویر یکسان در داخل دسته آموزش[6] قرار گیرند؛ در نتیجه محاسبه‌ی مشتقات آن‌ها به سادگی و با صرف حداقل حافظه‌ی گرافیکی قابل انجام خواهد بود.

ماشین فاکتورگیری ژرف[۲۸، ۲۹]

در ماشین‌های فاکتورگیری ساده یا با توجه، اهمیت خاصی به تعامل‌های مرتبه پایین داده می‌شود؛ در نتیجه مدل به سمت استفاده از تعامل‌های مرتبه پایین تشویق می‌شود و در نتیجه نوعی بایاس در طراحی این خانواده از مدل‌ها وجود دارد؛ اما ممکن است با در نظر نگرفتن این بایاس، تعاملات سطح بالای مناسب و مفیدی از داده‌ها کشف کنیم.

در مقابل ماشین‌های فاکتورگیری، که توانایی آن‌ها در مدل کردن مناسب تعاملات مرتبه پایین است، مدل‌های ژرف از جمله خانواده‌ی شبکه‌های عصبی چند لایه، توانایی بالایی برای مدل کردن تعاملات مرتبه بالا

[1] ReLU
[2] Dense
[3] Embedding
[4] Batch Normalization
[5] Feature Hashing
[6] Batch

دارند؛ اما به دلیل عدم توجه به تعاملات مرتبه پایین، در مساله‌ی پیش‌بینی نرخ کلیک کاربرد چندانی ندارند. ماشین‌های فاکتورگیری ژرف، ادغامی از این دو خانواده بوده و با ترکیب هر دو مدل، مدلی با انعطاف بیشتر و بایاس کمتر روی مرتبه‌ی تعامل‌ها ارائه می‌دهد.

در این مدل، دو بخش اصلی وجود دارد:

- **بخش ماشین فاکتورگیری**

این بخش تفاوتی با ماشین فاکتورگیری ساده ندارد. ابتدا ورودی‌هایش که همان ویژگی‌های تنک مساله هستند را به بردارهای تعبیه شده تبدیل کرده و سپس با اعمال ضرب داخلی بین این بردارها، تعامل‌های را محاسبه کرده و همچنین جمله‌ی خطی را به آن اضافه کرده و خروجی مورد نظر را از روی این مجموع ایجاد می‌کند.

- **بخش ژرف**

در این بخش از یک شبکه عصبی عادی استفاده می‌شود. ورودی‌های بخش ژرف، همان بردارهای تعبیه شده‌ی بخش ماشین فاکتورگیری هستند. توابع فعالیت در این بخش اکثرا رلو یا $tanh$ (تانژانت هایپربولیک) بوده و همه‌ی لایه‌های آن از نوع تماما متصل تشکیل شده‌اند.

در ماشین فاکتورگیری ژرف، از این ایده استفاده شده است که بردارهای تعبیه شده در ماشین‌فاکتورگیری، ویژگی‌های مناسبی ایجاد می‌کنند و به دلیل تنک نبودن و اندازه‌ی کمتر نسبت به ورودی‌های اصلی مساله‌ی پیش‌بینی نرخ کلیک، برای استفاده به عنوان ورودی یک شبکه عصبی ژرف کاملا مناسب هستند.

برای ترکیب این دو مدل، علاوه بر استفاده از ویژگی‌های مشترک، خروجی‌های آن‌ها نیز باهم جمع شده و به خاطر ماهیت مساله، که تخمین نرخ کلیک است، از مجموع خروجی‌های آن‌ها تابع سیگموید گرفته می‌شود. خروجی تابع سیگموید بین صفر و یک بوده و دقیقا مشابه توزیع احتمال یا نرخ کلیک است.

این مدل با استفاده از خطای لگاریتمی[1] و روش گرادیان کاهشی تصادفی[2] آموزش داده می‌شود.

مدل وسیع و ژرف[۳۰]

محققین شرکت گوگل[3]، شبکه‌ی وسیع و ژرف را برای توصیه‌ی اپلیکیشن‌ها در بازار اپلیکیشن گوگل پلی[4] توسعه داده و پژوهش خود را در سال ۲۰۱۶ منتشر کردند. به دلیل شباهت بالای کاربرد پیش‌بینی نرخ کلیک روی اپلیکیشن‌ها و پیش‌بینی نرخ کلیک روی تبلیغ‌ها، این مدل را مختصرا در این بخش معرفی می‌کنیم.

در مدل وسیع و ژرف، سه بخش اصلی وجود دارد:

[1] Log Loss
[2] Stochastic Gradient Descent
[3] Google
[4] Google Play Application Store

- مهندسی ویژگی‌ها

محققین در این پژوهش، ابتدا تعدادی از ویژگی‌های موجود در مجموعه‌های داده را حذف کرده و سپس ویژگی‌های سطح دوم را از روی بعضی از ویژگی‌های باقی مانده استخراج کردند. هر یک از ویژگی‌های مرتبه دوم، به صورت اشتراک بین دو ویژگی مرتبه اول تعریف شده و می‌توان آن را معادل تعامل بین دو ویژگی در ماشین‌های فاکتورگیری در نظر گرفت. این ویژگی‌ها پس از تبدیل به ویژگی‌های دسته‌ای یا دودویی، به کمک عمل تعبیه، به بردارهای چگال تعبیه تبدیل شده و در بخش‌های بعدی این مدل استفاده می‌شوند.

- بخش وسیع

در بخش وسیع، همه‌ی ویژگی‌های استخراج شده در بخش قبل کنار هم چسبانده شده و توسط یک تبدیل خطی، به فضای تک بعدی خروجی نگاشت می‌شوند.

- بخش ژرف

در بخش ژرف، بردارهای تعبیه شده به هم چسبانده شده و توسط یک شبکه‌ی عصبی چند لایه به فضای تک بعدی خروجی منتقل می‌شوند.

خروجی نهایی مدل وسیع و ژرف، از ترکیب خطی خروجی‌های بخش‌های وسیع و ژرف تشکیل شده و توسط خطای لگاریتمی آموزش داده می‌شود.

این مدل در رویارویی با چالش‌هایی از قبیل سرعت آزمایش، عملکرد قابل قبولی داشته و می‌تواند در کسری از ثانیه، اپلیکیشن‌های مختلف را برای نمایش به کاربران رتبه‌بندی کند؛ اما به دلیل نیاز به مهندسی ویژگی‌ها و همچنین تعداد بسیار بالای پارامترها، در مساله‌ی پیش‌بینی نرخ کلیک در تبلیغات نمایشی، قابل استفاده نیست؛ اما رویکرد ترکیب یک بخش ژرف و یک بخش غیر ژرف به طوری که ویژگی‌های سطح پایین و سطح بالا توسط این دو بخش به صورت مجزا آموخته شوند، در بسیاری از پژوهش‌های حوزه‌ی پیش‌بینی نرخ کلیک در تبلیغات نمایشی (مثل ماشین فاکتورگیری ژرف یا خودکدگذار پشته شده‌ی دارای توجه) به کار بسته شده است.

خودکدگذار پشته شده‌ی دارای توجه[۳۱]

شبکه‌ی عصبی خودکدگذار[1][۳۲]، یک روش یادگیری ماشین بدون نظارت است که از دو لایه‌ی شبکه‌ی عصبی تشکیل شده است. لایه‌ی اول، داده‌های ورودی را به فضای نهان[2] نگاشت کرده و لایه‌ی دوم، آن‌ها را به فضای ورودی باز می‌گرداند. شبکه‌ی خودکدگذار به این طریق آموزش داده می‌شود که فاصله‌ی اقلیدسی داده‌های ورودی و خروجی حداقل باشد. در نتیجه یک شبکه‌ی خودکدگذار ایده‌آل می‌تواند ورودی‌های

[1] Auto Encoder
[2] Latent Space

خود را بازسازی کند. در صورتی که لایه‌های این شبکه را به صورت مجزا در نظر بگیریم، لایه‌ی اول عمل کدگذاری[1] را انجام داده و لایه‌ی دوم عمل کدگشایی[2] را بر عهده می‌گیرد.

در ادبیات یادگیری ماشین، کاربردهای متنوعی برای شبکه‌های خودکدگذار ارائه شده که یکی از آن‌ها برای استخراج ویژگی بدون نیاز به داده‌های برچسب گذاری شده است. اگر پس از آموزش دادن یک خودکدگذار، صرفا از بخش کدگذار آن استفاده کرده و داده‌های کد شده را، به ورودی یک خودکدگذار دیگر بدهیم و این فرآیند را چندین بار انجام دهیم، یک خودکدگذار پشته شده[3] به وجود می‌آید. خودکدگذار پشته شده را می‌توان به صورت مرحله به مرحله یا به صورت یکجا آموزش داد. در صورتی که خطای بازسازی خودکدگذار پشته شده کم باشد، می‌توان نتیجه گرفت که ویژگی‌های استخراج شده در لایه‌ی میانی (پس از کدگذاری) حاوی اکثر اطلاعات مهم داده‌های ورودی بوده و به همین دلیل بخش کدگشا قادر به بازسازی داده‌های ورودی شده است؛ پس می‌توان به جای اطلاعات اصلی، از ویژگی‌های استخراج شده در لایه‌ی میانی (که از تعداد ابعاد کمتری برخوردار است) استفاده کرده و در نتیجه از ویژگی‌های سطح بالا و چگال مناسب بهره جست.

خودکدگذار پشته شده‌ی دارای توجه، مدلی است که برای پیش‌بینی نرخ کلیک ارائه شده و به نوعی ترکیبی از ماشین فاکتورگیری با توجه و خودکدگذار پشته شده است. این مدل از دو بخش تشکیل شده است:

- بخش ماشین فاکتورگیری با توجه

  ماشین فاکتورگیری با توجه، همانطور که قبلا بحث شد، یک مدل با پیچیدگی قابل توجه برای پیش‌بینی نرخ کلیک در تبلیغات نمایشی به شمار می‌رود. این بخش می‌تواند از ویژگی‌های مرتبه اول و دوم استفاده کرده و همچنین به کمک ساختار توجه، توازن را در میان ویژگی‌های مرتبه دوم رعایت کند.

- بخش خودکدگذار پشته شده

  خودکدگذار پشته شده همانطور که گفته شد، می‌تواند ویژگی‌های سطح بالا و فشرده استخراج کند. در این بخش، ابتدا ویژگی‌های تنک را به بردارهای تعبیه‌شده تبدیل کرده و سپس آن‌ها را کدگذاری و سپس کدگشایی می‌کنیم.

در فرآیند آموزش، ویژگی‌های لایه‌ی میانی بخش خودکدگذار پشته شده و ویژگی‌های مرتبه اول و دوم (که خروجی ماشین فاکتورگیری با توجه هستند) را به هم چسبانده و سپس توسط یک شبکه‌ی عصبی تک لایه، آن‌ها را به فضای تک بعدی خروجی نگاشت می‌کنیم.

برای آموزش خودکدگذار پشته شده‌ی باتوجه، خطای مدلسازی (خطای لگاریتمی) را با خطای بازسازی خودکدگذار جمع کرده و سپس از الگوریتم گرادیان کاهشی برای آموزش سراسری مدل استفاده می‌کنیم.

مدل خودکدگذار پشته شده‌ی با توجه به دلیل استفاده از ویژگی‌های سطح بالا در کنار ویژگی‌های سطح پایین، بر روی مجموعه‌های داده‌ی با حجم بالا، عملکرد بهتری از بسیاری از مدل‌های دیگر ارائه می‌دهد.

[1] Encoding
[2] Decoding
[3] Stacked Auto Encoder

همچنین به دلیل استفاده‌ی چندگانه از بردارهای تعبیه شده، سرعت یادگیری اولیه‌ی این مدل بهتر از سایر روش‌های مبنی بر بردارهای تعبیه است.

در این بخش، تعدادی از روش‌هایی که در ادبیات پیش‌بینی نرخ کلیک استفاده شده‌اند را معرفی و بررسی کردیم. خلاصه‌ای از مدل‌های ذکر شده و همچنین مقایسه‌ی کلی مزایا و معایب آن‌ها در جدول ۲-۱ نمایش داده شده است.

جدول ۲-۱: خلاصه‌ی روش‌های اصلی مطالعه شده

نام مدل	نقاط قوت	نقاط ضعف	سال و مرجع
ماشین بردار پشتیبان (کرنل چند جمله‌ای)	سرعت انجام بالا	تعداد پارامترهای بسیار بالا	۱۹۹۲[۸]
مدل تکه‌ای خطی	انعطاف‌پذیری پارامترهای تنک تفسیرپذیری مناسب	تعداد پارامتر زیاد آموزش کند تنظیم سخت ابرپارامترها	۲۰۱۷[۹]
مدل رگرسیون بیزی	امکان برقراری تعادل بین اکتشاف و بهره‌برداری	انعطاف پذیری کم نیاز به داده‌های زیاد	۲۰۰۹[۱۲]
ماشین فاکتورگیری	مدل‌سازی تعامل‌های مرتبه دوم تعداد کم پارامترهای مستقل	بی‌توجهی به روابط کلی بین فیلدها خطر بیش‌برازش	۲۰۱۰[۱۳]
ماشین فاکتورگیری آگاه از فیلد	مدلسازی تفاوت بین فیلدها	تعداد بالای پارامترها احتمال بالای بیش‌برازش	۲۰۱۶[۱۴، ۱۵]
ماشین فاکتورگیری با فیلدهای وزن‌دار	کنترل تعداد پارامترها مدل‌سازی تفاوت کلی فیلدها	توان مدل‌سازی محدود عدم مدل‌سازی تعامل‌های مرتبه بالا	۲۰۱۸[۱۶]
ماشین فاکتورگیری بیزی	امکان برقراری تعادل بین اکتشاف و بهره‌برداری	استنباط غیر قابل محاسبه پیش‌فرض نامناسب گاوسی	۲۰۱۱[۱۷]
ماشین فاکتورگیری تنک	تفسیر پذیری بالا تنک بودن مدل	استنباط غیر قابل محاسبه استفاده از تخمین برای محاسبه‌ی توزیع	۲۰۱۶[۱۸]
ماشین فاکتورگیری با توجه	افزایش پیچیدگی مدل افزایش تفسیرپذیری مدل	احتمال بیش‌برازش نیاز به داده‌های زیاد	۲۰۱۷[۱۹]
مدل ژرف پیش‌بینی نرخ کلیک	توانایی مدل تعاملات مرتبه بالا تعمیم پذیری مناسب	نیاز به تصویر بنر تبلیغ امکان بیش‌برازش به دلیل کمبود داده عدم مواجهه با چالش شروع سرد	۲۰۱۶[۲۳]
ماشین فاکتورگیری ژرف	مدل‌سازی تعاملات مرتبه بالا عدم وجود بایاس در مرتبه تعاملات	تعداد زیاد ابرپارامتر تفسیرپذیری پایین	۲۰۱۷[۲۸، ۲۹]
مدل وسیع و ژرف	پیاده‌سازی سریع توان مدل‌سازی مناسب	نیاز به مهندسی ویژگی‌ها تعداد بالای پارامترها	۲۰۱۶[۳۰]
خودکدگذار پشته شده‌ی با توجه	توانایی مدل‌سازی مناسب اشتراک بالای پارامترها	تعداد زیاد پارامترها احتمال بیش‌برازش	۲۰۱۸[۳۱]

فصل ۳

روش پیشنهادی

در فصل قبل، روش‌های حل مساله‌ی پیش‌بینی نرخ کلیک را دسته‌بندی کرده و تعدادی از پژوهش‌های مهم هر دسته را بررسی و مقایسه کرده و با بیان مزایا و کاستی‌های هر کدام، دید مناسبی از دشواری‌ها و چالش‌های این مساله کسب کردیم.

در این فصل، با در نظر گرفتن چالش‌های مساله‌ی پیش‌بینی نرخ کلیک و همچنین با توجه به ایرادات یا کاستی‌های مشترک روش‌های پیشین، اقدام به طراحی یک مدل جدید، برای حل این مساله می‌نماییم. برای طراحی این مدل جدید، اقدام به معرفی ایده‌های جدید و همچنین بهره‌گیری از برخی ایده‌های موجود در ادبیات یادگیری ماشین کرده و در هر گام، با توجه به چالش‌های ذاتی مساله و همچنین محدودیت‌های ناشی از گام‌های قبلی، روش پیشنهادی را توسعه می‌دهیم.

۳-۱ تعبیه‌ی ویژگی‌ها

از آن‌جا که استفاده از بردارهای تعبیه شده، امری ضروری برای بهره‌گیری از ویژگی‌های دسته‌ای موجود در مجموعه‌های داده‌ی پیش‌بینی نرخ کلیک به شمار می‌رود، طراحی مدل پیشنهادی را از همین بخش آغاز می‌نماییم.

در فصل قبل با مطالعه‌ی تعداد قابل توجهی از روش‌های پیشین، مشاهده کردیم که همه‌ی این پژوهش‌ها، در یک اصل مشترک هستند. همه‌ی این روش‌ها، با استفاده از ترفند تعبیه، ویژگی‌های دسته‌ای ورودی را به بردارهای چگال قابل یادگیری تبدیل کرده و سپس این بردارها را برای استفاده در بقیه‌ی قسمت‌های مدل، به کار می‌بندند. نکته‌ی دیگرِ قابل توجه و مشترک در همه‌ی این روش‌ها، استفاده از بردارهای تعبیه با بعد یکسان برای ویژگی‌های همه‌ی فیلدها است.

می‌توانیم استفاده از بردارهای هم‌بعد را به این صورت تعبیر کنیم که در این مدل‌ها، برای هر فیلد یک فضای k بعدی در نظر گرفته شده و تمامی ویژگی‌ها (حالت‌ها)ی این فیلد، به عنوان نقاطی در این فضای k بعدی

جای می‌گیرند. به عنوان مثال، در صورتی که فیلد F_a دارای ۳ حالت مختلف و فیلد F_b دارای ۱۰۰۰ حالت مختلف باشند، در فضای تعبیه‌ی فیلد اول (E_a) سه نقطه (یا سه بردار k بعدی) و همچنین در فضای تعبیه‌ی فیلد دوم (E_b) هزار نقطه (یا بردار k بعدی) حضور خواهند داشت؛ پس جایگیری نقاط در فضای E_b نسبت به جایگیری نقاط در فضای E_a شرایط فشرده‌تری دارد.

با ملاحظه‌ی نکته‌ی فوق، این سوال به وجود می‌آید که **آیا تعبیه‌ی ویژگی‌های همه‌ی فیلدها در فضای دارای ابعاد یکسان (که همه‌ی روش‌های پیشین در انجام آن اتفاق دارند)، بهترین تصمیم ممکن است؟** برای پاسخ به این سوال، می‌توانیم از دو روش مختلف استفاده کنیم. روش اول، استفاده از نگرش مرسوم در نظریه‌ی اطلاعات[1] برای اندازه‌گیری اطلاعات موجود در این بردارها و روش دوم، بررسی شهودی این مساله، با توجه به مفاهیم مرسوم در ادبیات یادگیری ماشین و یادگیری ژرف[2] است.

۳-۱-۱ بررسی ابعاد بردارهای تعبیه به کمک نظریه‌ی اطلاعات

در نظریه‌ی اطلاعات[۳۳]، آنتروپی[3] یک ویژگی دسته‌ای (فیلد)، به صورت زیر محاسبه می‌شود:

$$H(F) = -\sum_{i=1}^{|F|} p_i log_2(p_i) \tag{3.1}$$

که در آن $|F|$ تعداد دسته‌ها و p_i احتمال وقوع حالت iام این ویژگی هستند.

در صورتی که این ویژگی دسته‌ای را در فضای k بعدی تعبیه کنیم و هریک از عناصر موجود در بردارهای تعبیه، دارای s بیت باشند، می‌توانیم آنتروپی بردار تعبیه‌شده را محاسبه کنیم:

$$H(E) = -\sum_{i=1}^{2^{ks}} p_i log_2(p_i) \tag{3.2}$$

که در آن p_i احتمال یک بودن بیت iام این بردار است.

با مقایسه‌ی دو رابطه‌ی ۱.۳ و ۲.۳ می‌توانیم میزان اطلاعات موجود در آن فیلد را، با میزان اطلاعات قابل بیان توسط بردار تعبیه شده مقایسه کنیم.

در پژوهش [۳۴] با فرض هم احتمال بودن توزیع حالت‌های ویژگی دسته‌ای و همچنین هم احتمال بودن توزیع بیت‌های بردار تعبیه شده، مقایسه‌ی فوق را انجام داده و در نتیجه به رابطه‌ی زیر رسیدند:

$$log_2(|F|) = k.s \tag{3.3}$$

[1] Information Theory
[2] Deep Learning
[3] Entropy

می‌توان این رابطه را به این صورت تعبیر کرد که برای تناسب اطلاعات موجود در ویژگی دسته‌ای و بردار تعبیه شده‌ی مربوطه، باید بعد تخصیص داده شده به آن بردار با لگاریتم کاردینالیتی مجموعه‌ی حالات مختلف انتخاب آن متناسب باشد؛ پس فیلدی که کاردینالیتی بالاتری داشته باشد، باید در فضای دارای ابعاد بیشتر تعبیه شود.

با در نظر گرفتن این نکته که مورد استفاده‌ی اصلی بردارهای تعبیه شده در مدل‌های یادگیری ماشین و یادگیری ژرف است، می‌توان رابطه‌ی بالا را نقد کرد. در رابطه‌ی گفته شده، فرض شده است که همه‌ی اطلاعات موجود در فیلد باید در بردارهای تعبیه شده موجود باشد. همچنین فرض شده است که از همه‌ی بیت‌های بردارهای تعبیه شده برای ذخیره‌ی این اطلاعات استفاده می‌شود. این در حالی است که در یادگیری ماشین و یادگیری ژرف، هیچ یک از این دو فرض صحت ندارند. مدل‌های یادگیری ماشین و یادگیری ژرف، به جای همه‌ی اطلاعات موجود در ویژگی دسته‌ای، تنها به اطلاعات مشترک این ویژگی با متغیر هدف (خروجی) نیاز داشته و همچنین، از بردارهای تعبیه شده این انتظار می‌رود که به جای فشرده‌سازی حداکثری، دارای فواصل کم بین ویژگی‌های مشابه و فواصل زیاد بین ویژگی‌های متفاوت باشد. در نتیجه مدل‌های گفته شده، بتوانند از اطلاعات موجود در این بردارها به صورت مطلوب استفاده کنند.

با وجود این فرض‌های اشتباه و فرض ساده‌کننده‌ی توزیع یکنواخت، این روابط تنها با افزودن چند ضریب قابل اصلاح است. فرض می‌کنیم اطلاعات مشترک بین متغیر هدف و هریک از فیلدها، به صورت ضریب ثابتی (μ) از اطلاعات موجود در آن فیلد باشد. همچنین، فرض می‌کنیم هر چند (δ) بردار تعبیه، به دلیل شباهت مفهوم مربوطه، در محل یکسانی از فضای تعبیه جا بگیرند.

$$I(y, F_i) = H(F_i) \times \mu = log_2(\frac{|F_i|}{\delta}) \times \mu \tag{3.4}$$

همچنین فرض می‌کنیم برای مطلوب بودن فضای تعبیه، انتظار می‌رود تنها از کسر ثابتی (κ) از ظرفیت بیت‌های موجود در بردار تعبیه استفاده شود.

$$H(E) = k.s.\kappa \tag{3.5}$$

حال با برابر قرار دادن روابط ۴.۳ و ۵.۳، به این رابطه می‌رسیم:

$$k = log_2(\frac{|F_i|}{\delta}) \times \frac{\mu}{s \times \kappa} \tag{3.6}$$

که می‌توان آن را به صورت زیر هم نوشت:

$$k = \omega. \ln(|F_i|) + \epsilon \tag{3.7}$$

که در آن، با معرفی پارامترهای ω و ϵ همه‌ی ضرایب ثابت را یک جا جمع می‌کنیم. حال از طریق رابطه‌ی فوق، می‌توانیم ابعاد مناسب برای تعبیه‌ی هر فیلد را محاسبه کنیم.

۳-۱-۲ بررسی ابعاد بردارهای تعبیه به کمک مفاهیم شهودی یادگیری ماشین و یادگیری ژرف

در بخش قبل، به کمک مفاهیم نظریه‌ی اطلاعات، رابطه‌ای برای تخصیص مناسب بعد به فیلدها ارائه کنیم. در این بخش، با بهره‌گیری از شهود و همچنین برخی از مفاهیم مورد استفاده در ادبیات یادگیری ماشین و یادگیری ژرف، نسبت به توجیه، نقد و اصلاح رابطه‌ی ارائه شده اقدام می‌کنیم.

فرض کنید یک ویژگی دسته‌ای توسط بردارهایی تعبیه می‌شود و یک مدل شبکه عصبی، با استفاده از اطلاعات موجود در این بردارها، نسبت به تخمین یک متغیر هدف اقدام می‌کند. چون واحدهای سازنده‌ی شبکه‌های عصبی، نورون‌های خطی هستند، در شرایطی که بیش‌برازش شدید موجود نباشد، این شبکه مرز تصمیم‌گیری نرمی خواهد داشت. به این معنا که نقاطی‌که در کنار هم تعبیه شده‌اند، به احتمال بسیار زیاد به یک کلاس تخصیص داده خواهند شد.

در صورتی که بعد تعبیه‌ی این مدل را افزایش دهیم، این نقاط می‌توانند از هم دورتر شده و لذا چگالی نقاط در این فضا کاهش می‌یابد. با کاهش چگالی نقاط، مدل قادر خواهد بود این نقاط را با دقت بیشتری از هم جدا کرده و لذا در صورت زیاده روی در افزایش بعد تعبیه، شباهت بین این نقاط توسط مدل قابل درک نخواهد بود. این پدیده می‌تواند یکی از شکل‌های بیش‌برازش را ایجاد کند.

در مقابل، اگر بعد تعبیه‌ی این مدل را کاهش دهیم، این نقاط به هم نزدیک‌تر شده و لذا چگالی نقاط در این فضا افزایش می‌یابد. با افزایش چگالی نقاط، مدل توانایی جداسازی این نقاط از هم را از دست می‌دهد. در نتیجه توان مدل‌سازی مدل کاهش یافته و عملا کیفیت عملکرد مدل افت خواهد کرد.

از مثال بالا، می‌توانیم این مفهوم را برداشت کنیم که برای دسترسی به بهترین عملکرد ممکن، چگالی نقاط در فضای تعبیه باید مقدار معینی داشته باشد. برای درک بهتر این مفهوم، می‌توانیم تعریف فیزیکی چگالی را در نظر گرفته و سعی کنیم رابطه‌ای برای بعد تعبیه به دست آوریم.

از آن‌جا که تعریف فیزیکی چگالی، از تقسیم تعداد ذرات به حجم محاسبه می‌شود، ولی فضای تعبیه حجم بی‌نهایت دارد، مجبوریم این تعریف را تا حدودی تغییر دهیم. اعمال تکنیک‌های تنظیم بر پارامترهای تعبیه، باعث محدود شدن محل هندسی بردارهای تعبیه شده می‌شوند و لذا می‌توانیم فرض کنیم همه‌ی پارامترهای تعبیه، در بازه‌ی $(-\frac{L}{۲}, \frac{L}{۲})$ محدود خواهند بود؛ پس اگر بعد تعبیه را k و تعداد نقاطی که در این فضا تعبیه می‌شوند را n در نظر بگیریم، می‌توانیم چگالی متوسط این نقاط را محاسبه کنیم:

$$density(E) = \frac{n}{L^k} \tag{3.8}$$

حال اگر در یک مدل که بیش از یک ویژگی دسته‌ای ورودی دارد، مقدار چگالی را برای فضای تعبیه‌ی همه‌ی فیلدها یکسان در نظر بگیریم:

$$\frac{|F_i|}{L^{k_i}} = \frac{|F_j|}{L^{k_j}} = c.t.e. \tag{3.9}$$

با لگاریتم گرفتن از رابطه‌ی فوق می‌توانیم:

$$\ln(|F_i|) - k_i \ln(L) = \ln(|F_j|) - k_j \ln(L) = c \tag{3.10}$$

که در آن c یک عدد ثابت بوده و خواهیم داشت:

$$\forall i : k_i = \frac{\ln(|F_i|) - c}{ln(L)} \tag{3.11}$$

با تغییر دادن پارامترها، می‌توان این رابطه را به شکل زیر در آورد:

$$\forall i : k_i = \omega \times \ln(|F_i|) + \epsilon \tag{3.12}$$

که رابطه‌ی اخیر، کاملا بر رابطه‌ی به دست آمده به کمک مفاهیم نظریه‌ی اطلاعات مطابقت دارد.

رابطه‌ی به دست آمده، نسبت به کاردینالیتی فیلد، صعودی است. به این معنا که فیلد دارای دسته‌های بیشتر، لزوما در فضای دارای بعدهای بالاتر تعبیه خواهد شد؛ پس در صورتی که مدلی از این رابطه برای تخصیص پارامترهای تعبیه به فیلدهای ورودی استفاده کند، همیشه تعداد پارامترهای بیشتری به فیلدهایی که کاردینالیتی بالاتر دارند در نظر می‌گیرد. اگر بخواهیم یک مثال افراطی از این مساله را مطرح کنیم، می‌توانیم فیلدهای شناسه[1] را در نظر بگیریم. فیلد شناسه، به ویژگی‌هایی گفته می‌شود که در هر رکورد از مجموعه‌ی داده، یک مقدار متفاوت به خود گرفته و لذا هرگز در مجموعه‌ی داده تکرار نمی‌شوند. هر چند چنین ویژگی‌هایی از نظر نظریه‌ی اطلاعات، دارای آنتروپی و اطلاعات زیادی هستند، اما واضح است که به دلیل عدم تکرار (یا تکرار بسیار کم) آن‌ها در مجموعه‌ی داده، یادگیری از آن‌ها را غیر ممکن ساخته و لذا تخصیص پارامترهای زیاد به این ویژگی‌ها، باعث هدر رفتن قدرت محاسباتی و همچنین افزایش خطر بیش‌برازش می‌شود.

برای اصلاح این مشکل، می‌توانیم رابطه‌ی فوق را به شکل زیر تغییر دهیم:

$$\forall i : k_i = \omega \times \ln(|F_i|) \times \frac{|Dataset| - |F_i|}{|Dataset|} + \epsilon \tag{3.13}$$

که در آن $|Dataset|$ تعداد رکوردهای موجود در مجموعه‌ی داده است. همان‌طور که مشخص است، کسر $\frac{|Dataset|-|F_i|}{|Dataset|}$ در صورتی که $|F_i|$ نسبت به تعداد رکوردهای مجموعه‌ی داده، مقدار کمی داشته باشد، تقریبا برابر یک بوده و تاثیر چندانی روی نتیجه‌ی رابطه نمی‌گذارد؛ اما در صورتی که $|F_i|$ نسبت به $|Dataset|$ قابل مقایسه باشد، این کسر میزان کمتر از یک به خود گرفته و لذا بعد تعبیه را برای این ویژگی‌ها کاهش می‌دهد. به عبارت دیگر، برای ویژگی‌هایی که تعداد تکرار موجودیت‌های آن‌ها در مجموعه‌ی داده کم باشد، بعد تعبیه را کمی کاهش می‌دهیم. در حالت افراطی ویژگی‌های شناسه، که در آن‌ها $|F_i|$ تقریبا با $|Dataset|$

[1] ID

برابر است، میزان این کسر تقریبا برابر صفر شده و لذا بعد تعبیه برای این ویژگی‌ها به حداقل کاهش می‌یابد.

با توجه به نکات مطرح شده، در این پژوهش بعد تعبیه‌ی هریک از ویژگی‌های دسته‌ای، از رابطه‌ی نهایی زیر محاسبه خواهد شد:

$$\forall_{1\leq i\leq f} : Dim(F_i) = \omega \times \ln(|F_i|) \times \frac{|Dataset| - |F_i|}{|Dataset|} + \epsilon \tag{3.14}$$

که در آن، f تعداد فیلدهای ورودی است. همچنین، پارامترهای مربوط به تعبیه‌ی فیلدهای مدل را به صورت زیر تعریف می‌کنیم:

$$\forall_{1\leq i\leq f} : \mathbf{E}_i \in \mathbb{R}^{|F_i|\times Dim(|F_i|)} \tag{3.15}$$

حال اگر x_i اندیس ویژگی فعال در فیلد iام باشد، بردارهای تعبیه شده‌ی مدل به این صورت تعریف می‌شوند:

$$\forall_{1\leq i\leq f} : e_i = \mathbf{E}_i^{x_i} \in \mathbb{R}^{Dim(|F_i|)} \tag{3.16}$$

در این بخش از دو زاویه‌ی متفاوت به مساله‌ی محاسبه‌ی بعد تعبیه برای فیلدهای ورودی نگریسته و به یک نتیجه‌ی یکسان رسیدیم. نتایج هر دو بررسی، پرسشی را که در ابتدای این بخش مطرح کرده بودیم را رد کرده و لذا برخلاف همه‌ی روش‌های پیشین، در این پژوهش فیلدهای مختلف ورودی را در فضاهایی با ابعاد متفاوت تعبیه کرده و مدل محاسباتی خود را، بر مبنای این بردارهای تعبیه شده، طراحی می‌نماییم.

۳-۲ محاسبه‌ی تعامل

در ادبیات پیش‌بینی نرخ کلیک، مفهوم تعامل، به ویژگی‌های درجه دوم (یا بیشتر)ی اشاره می‌کند که نشان دهنده‌ی تاثیر رخداد همزمان دو (یا چند) ویژگی باینری بر تصمیمات مدل هستند. به عبارت دیگر، تمامی اطلاعاتی که مدل از رخداد همزمان دو ویژگی باینری نیاز دارد، باید از طریق تعامل بین این دو ویژگی تامین شود. بدون در نظر گرفتن مفهوم تعامل، اکثر روش‌های موجود در ادبیات پیش‌بینی نرخ کلیک، به یک مدل خطی و ساده کاهش یافته و این مساله، اهمیت بالای این مفهوم را می‌رساند.

در اکثر روش‌های معرفی شده‌ی پیشین که از مفهوم تعامل برای افزایش قابلیت مدل‌سازی استفاده می‌کنند، برای محاسبه‌ی میزان تعامل از مکانیزم‌های بسیار ساده‌ای نظیر ضرب داخلی (در روش‌های مبتنی بر ماشین فاکتورگیری ساده)، یا ضرب درایه به درایه و سپس ترکیب خطی از نتایج حاصل از آن (در روش‌های مبتنی بر ماشین فاکتورگیری با توجه) استفاده می‌کنند. در نتیجه همه‌ی این مدل‌ها از نیاز به استفاده از بردارهای تعبیه‌ی هم بعد برای همه‌ی فیلدها (که در بخش قبل نشان دادیم ویژگی مناسبی نیست) رنج می‌برند.

به دلیل محدودیت عملگرهای ضرب داخلی و ضرب درایه به درایه به استفاده از بردارهای تعبیه‌ی هم بعد،

در این پژوهش نمی‌توانیم به صورت مستقیم از این عملگرها برای محاسبه‌ی میزان تعامل بین ویژگی‌های فیلدهای مختلف بهره ببریم؛ در نتیجه باید راه دیگری برای پیاده‌سازی مفهوم تعامل بیابیم. در این بخش دو روش ممکن برای محاسبه‌ی مقادیر تعامل را معرفی می‌کنیم.

۳-۲-۱ نگاشت خطی بردارهای تعبیه به فضای هم‌بعد

در پژوهش [۳۵]، یک روش ساده برای مقابله با این مشکل معرفی شده است. می‌دانیم ضرب ماتریسی بردارهای یک فضای k_{in} بعدی، در یک ماتریس با ابعاد $k_{in} \times k_{out}$، یک نگاشت خطی بین فضای k_{in} بعدی گفته شده و یک فضای k_{out} بعدی جدید است. یعنی اگر n بردار از فضای اول را در سطرهای ماتریس X قرار دهیم و همچنین ماتریس $W_{k_{in} \times k_{out}}$ را از سمت راست در X ضرب کنیم، حاصل این عمل نگاشت این بردارها در یک فضای جدید k_{out} بعدی خواهد بود:

$$Y_{n \times k_{out}} = X_{n \times k_{in}} W_{k_{in} \times k_{out}} \tag{3.17}$$

در صورتی که $k_{in} < k_{out}$ باشد، نقاط در فضای Y تنها به یک زیرفضا (منیفولد) از این فضای k_{out} بعدی محدود شده و از تمام پیچیدگی موجود در این فضا استفاده نخواهد شد. همچنین در صورتی که $k_{in} > k_{out}$ باشد، نقاط در فضای Y به صورت فشرده‌تری حضور داشته و می‌توان گفت میزانی از اطلاعات نهفته در این نقاط، از دست خواهد رفت.

در پژوهش فوق، پیشنهاد شده است که پس از ایجاد بردارهای تعبیه در فضاهای با ابعاد مختلف، با استفاده از تعدادی تبدیل ماتریسی خطی، همه‌ی این فضاها را به فضای k بعدی مشترک نگاشت کنیم؛ سپس مقادیر تعامل را مانند ماشین‌های فاکتورگیری، به کمک عملگر ضرب داخلی محاسبه کنیم. چون ابعاد ماتریس‌های گفته شده و در نتیجه تعداد پارامترهای آن‌ها در مقایسه با تعداد پارامترهای جدول‌های تعبیه ناچیز خواهد بود، لذا به سادگی می‌توانیم این پارامترها را به کمک روش‌های گرادیان کاهشی بیاموزیم.

۳-۲-۲ محاسبه‌ی تعامل به کمک شبکه‌ی عصبی

در پژوهش [۳۶] که در ادبیات سیستم‌های پیشنهاد دهنده انجام شده است، برای محاسبه‌ی تعامل بین دو ویژگی کاربر و کالا، که به مساله‌ی فیلتر کردن مشترک[1] معروف است، از ایده‌ی متفاوتی استفاده شده است. لازم به ذکر است این پژوهش، چندین روش مختلف و ترکیب آن‌ها را معرفی کرده است، در صورتی که در این پژوهش، تنها به یکی از این روش‌ها رجوع کرده و از ایده‌ی موجود در آن بهره می‌جوییم.

برای محاسبه‌ی تعامل بین دو بردار تعبیه شده، لزومی بر استفاده از عملگر ضرب داخلی وجود ندارد، بلکه می‌توان از یک شبکه‌ی عصبی چند لایه[2] بهره جست. مهمترین ویژگی شبکه‌های عصبی چند لایه، توانایی

[1] Collaborative Filtering
[2] Multi Layer Perceptron

تخمین همه‌ی توابع است. یعنی در صورتی که یک شبکه‌ی عصبی چند لایه، به تعداد کافی نورون داشته باشد و همچنین با مقدار کافی داده آموزش داده شود، می‌تواند روابط موجود بین این داده‌ها را با میزان خطای دلخواه[1] فرا گرفته و تخمین بزند؛ لذا به شبکه‌های عصبی چند لایه، تخمین زننده‌ی سراسری[2] نیز گفته می‌شود.

در پژوهش فوق، پس از تعبیه‌ی دو ویژگی موجود در مجموعه‌ی داده، بردارهای تعبیه شده را به هم چسبانده و سپس از یک شبکه‌ی عصبی چند لایه برای محاسبه‌ی تعامل استفاده می‌شود. به دلیل سادگی و تطبیق پذیری شبکه‌های عصبی چندلایه، در این پژوهش نیز از این شبکه‌ها برای محاسبه‌ی تعامل بین ویژگی‌ها بهره خواهیم جست.

۳-۲-۳ تعامل‌های چندبعدی به جای تعامل‌های چندگانه

یکی از مزایای ماشین‌های فاکتورگیری، سادگی پیاده‌سازی تعامل‌های چندگانه است. تعامل‌های چندگانه، به محاسبه‌ی مفهوم تعامل بیشتر از دو ویژگی به صورت همزمان اشاره می‌کند. در ماشین‌های فاکتورگیری، به دلیل محاسبه‌ی تعامل به صورت ضرب داخلی، به سادگی می‌توان عمل محاسبه‌ی تعامل را به بیش از دو ویژگی تعمیم داد. به عنوان مثال، رابطه‌ی زیر تعامل میان سه ویژگی در یک ماشین فاکتورگیری (مرتبه سوم) را نشان می‌دهد:

$$I_{i,j,l} = \sum_{m=1}^{k} e_{i_m} e_{j_m} e_{l_m} \tag{3.18}$$

که در آن k بعد تعبیه‌ی مشترک همه‌ی فیلدها است. به این ترتیب، ماشین‌های فاکتورگیری ساده و بسیاری از مشتقات آن، به سادگی قابلیت محاسبه‌ی تعامل چندگانه را دارا هستند. تعامل چندگانه قابلیت مدل‌سازی را افزایش داده و البته خطر بیش‌برازش را افزایش می‌دهد.

در این پژوهش، می‌توانیم تعامل چندگانه را به سادگی با به هم چسباندن بیش از دو بردار تعبیه شده و تخصیص یک شبکه‌ی عصبی به چند تایی مرتب فیلدهای انتخاب شده پیاده‌سازی کنیم؛ اما انجام این کار باعث افزایش بی‌رویه‌ی پیچیدگی مدل و کاهش مقیاس پذیری روش پیشنهادی خواهد شد.

ایده‌ای که برای رویارویی با این مشکل در این پژوهش معرفی می‌کنیم، استفاده از تعامل‌های چندبعدی است. همه‌ی روش‌های پیشین به دلیل محدودیت‌های ساختاری، مفهوم تعامل را به یک مفهوم تک بعدی که رابطه‌ی آن با احتمال کلیک خطی است، تقلیل داده‌اند. این در حالی است که می‌توانیم مفهوم تعامل را به صورت زیر تعریف کرده و تعمیم دهیم:

تعریف ۱. تعامل بین دو فیلد، بردار نهفته‌ای است که تمامی اطلاعاتی که در زوج مرتب آن دو فیلد وجود دارد و برای تخمین نرخ کلیک مورد نیاز است را به صورت فشرده نمایش می‌دهد.

[1] Arbitrary
[2] Global Approximator

تعریف فوق دو تفاوت عمده با تعریف تعامل در خانواده‌ی ماشین‌های فاکتورگیری دارد:

۱. **چندبعدی بودن**

تعامل بین دو فیلد می‌تواند به جای تک بعدی بودن، چند بعدی باشد و در نتیجه رفتاری مانند بردارهای نهان در شبکه‌های عصبی داشته باشد. به این معنی که فضای چندبعدی ایجاد شده توسط تعامل بین دو فیلد، می‌تواند حاوی اطلاعاتی باشد که بخش‌های دیگر مدل، آن را به صورت یک ویژگی سطح بالا دریافت کرده و لذا قادر به استخراج میزان بیشتری اطلاعات از این بردار نهان چندبعدی خواهند بود.

۲. **رابطه‌ی غیر خطی**

در ماشین‌های فاکتورگیری، فرض شده است که مجموع همه‌ی تعامل بین فیلدهای مختلف، با افزوده شدن به جملات خطی رگرسیون، به صورت مستقیم احتمال کلیک را تخمین می‌زنند. این در حالی است که با در نظر گرفتن مفهوم تعامل به عنوان ویژگی‌های نهان در یک مدل ژرف، می‌توان روابط پیچیده‌تری نسبت به رابطه‌ی خطی بین تعامل بین فیلدها و احتمال کلیک کشف نمود؛ پس لزومی ندارد که از رابطه‌ی بین احتمال کلیک و تعامل‌های بین ویژگی‌ها را به یک رابطه‌ی خطی تقلیل دهیم.

با در نظر گرفتن تفاوت‌های گفته شده، می‌توان ساختار پیشنهادی را ارائه کرد، ولی پیش از معرفی نهایی ساختار پیشنهادی، پرسشی که ممکن است در این مرحله به ذهن برسد را مطرح کرده و پاسخ می‌دهیم.

- **پرسش**

چرا به جای تعامل چندبعدی، با افزایش تعداد لایه‌ها در شبکه‌های تعامل، از تعامل تک بعدی استفاده نکنیم؟ این گونه به نظر می‌رسد که در صورتی که تعداد لایه‌های شبکه‌های تعامل را افزایش دهیم، مدل می‌تواند تعامل‌های چندبعدی گفته شده را در یکی از لایه‌های نهان داخل همین شبکه‌های تعامل فرا گرفته و سپس با استخراج اطلاعات مفید آن، تعامل را به صورت تک بعدی به بخش‌های دیگر مدل انتقال دهد؛ در نتیجه معرفی تعامل چندبعدی به نظر بی‌دلیل می‌رسد.

- **پاسخ**

با توجه به تعریف بالا برای مفهوم تعامل، این مفهوم مربوط به اطلاعات مشترکی است که بین ویژگی‌های دو فیلد وجود دارند؛ پس در نظر گرفتن تنگنا[1]ی تک بعدی، باعث محدودیت شده و ممکن است این اطلاعات مشترک برای عبور از این تنگنا فیلتر شده و بخش مهمی از این اطلاعات از دست برود.

دلیل دیگر استفاده از تعامل‌های چندبعدی، به اشکالی که قبلا معرفی کردیم یعنی عدم مقیاس پذیری مدل پیشنهادی در صورت استفاده از تعامل‌های چندگانه باز می‌گردد. در صورتی که اطلاعات مهمی در تعامل سه فیلد یا بیشتر وجود داشته باشد، در تعامل‌های تک بعدی این اطلاعات در تنگنای فوق حذف شده و مدل قادر به استخراج اطلاعات مربوط به تعامل چندگانه نخواهد بود.

[1] Bottleneck

این در حالی است که اگر اجازه دهیم بردارهای تعامل، چند بعدی باشند، مدل می‌تواند از کنار هم قرار دادن تعامل‌های دوگانه، تعامل‌های مرتبه‌ی بالاتر را به صورت ضمنی[1] محاسبه کرده و از آن برای پیش‌بینی نرخ کلیک بهره ببرد. عملا با در نظر گرفتن تعامل‌های چندبعدی، نیاز به استفاده از تعامل‌های چندگانه حذف شده و لذا مقیاس پذیری مدل افزایش می‌یابد.

با استدلال‌های گفته شده، روش محاسبه‌ی بردارهای تعامل بین فیلدهای ورودی تکمیل شده و لذا در این بخش، با اشاره به برخی جزئیات، این بخش مهم از روش پیشنهادی را جمع بندی می‌کنیم.

در بخش قبل بردارهای تعبیه شده‌ی مدل را تعریف کردیم. اگر در مجموعه‌ی داده، f فیلد داشته باشیم، بردارهای تعبیه‌ی فیلدها را با $\{e_1, e_2, \ldots, e_f\}$ نمایش می‌دهیم. چون تعامل بین ویژگی‌های هر دو فیلد محاسبه می‌شود، نیاز به $\frac{f(f-1)}{2}$ شبکه‌ی عصبی تعامل خواهیم داشت. برای سادگی نامگذاری، این شبکه‌ها را به صورت زیر نامگذاری می‌کنیم:

$$\forall_{1\leq i<j\leq f}, InteractionNet_{i,j} : \mathbb{R}^{Dim(|F_i|)+Dim(|F_j|)} \rightarrow \mathbb{R}^{Dim_{Int}} \tag{3.19}$$

شبکه‌ی $InteractionNet_{i,j}$ چندلایه بوده و تعداد نورون‌های هر لایه، به صورت خطی کاهش می‌یابد تا از $Dim(|F_i|) + Dim(|F_j|)$ بعد به Dim_{Int} بعد برسد. تعداد لایه‌های همه‌ی این شبکه‌ها برابر $Depth_{Interaction}$ است. در فصل بعد با انجام آزمایش‌هایی، تعداد لایه‌ها و همچنین بعد بردارهای تعامل مناسب را به دست خواهیم آورد.

تابع فعال‌ساز همه‌ی لایه‌های این شبکه‌ها (بجز لایه‌ی آخر) را واحد خطی یکسو کننده‌ی نشت کننده[2][۳۷] در نظر می‌گیریم. دلیل استفاده از این تابع، انتقال بهتر گرادیان به لایه‌های پایین‌تر است. در لایه‌ی آخر این شبکه‌ها، برای استفاده در بخش‌های دیگر مدل، از هیچ تابع فعال‌سازی استفاده نمی‌کنیم. در این پژوهش مقادیر بردارهای تعامل را به شکل زیر نامگذاری و محاسبه می‌کنیم:

$$\forall_{1\leq i<j\leq f}, I_{i,j} = InteractionNet_{i,j}(e_i : e_j) \tag{3.20}$$

۳-۳ استفاده از بردارهای تعبیه و تعامل برای تخمین نرخ کلیک

در بخش‌های قبل، شیوه‌ی تعبیه‌ی ویژگی‌ها و همچنین نحوه‌ی محاسبه‌ی تعامل بین بردارهای تعبیه شده در روش پیشنهادی را معرفی کردیم. در این قسمت تنها بخش باقی مانده‌ی مدل را معرفی می‌کنیم. این بخش شبکه‌ی سر[3] نام دارد و مسئول استفاده از همه‌ی ویژگی‌هایی که تا اینجا تعریف کردیم و پیش‌بینی نرخ کلیک به کمک این ویژگی‌ها است.

[1] Implicit
[2] LeakyReLU
[3] Head Network

در تعدادی از پژوهش‌های پیشین که از مدل‌های ژرف استفاده کرده‌اند، برای محاسبه‌ی نرخ کلیک از دو دسته ویژگی مهم استفاده می‌شود:

۱. **ویژگی‌های مرتبه پایین**

ویژگی‌های مرتبه پایین در مدل‌های ژرف مبتنی بر ماشین فاکتورگیری، شامل جمله‌ی بایاس، جملات خطی و همچنین تعامل‌های مرتبه دوم است. همانطور که مشخص است، این ویژگی‌ها نقش اساسی در شکل دهی به تابع تصمیم‌گیری مدل‌ها دارند. این ویژگی‌ها به دلیل سادگی در محاسبه و همچنین نقش ساده و مشخص در پیش‌بینی نرخ کلیک، به سادگی نیز آموزش یافته و به همین دلیل با تعداد داده‌های کم نیز قابل یادگیری هستند.

۲. **ویژگی‌های مرتبه بالا** با گسترش روش‌های ژرف، محققین متوجه توانایی بالای این روش‌ها برای استخراج ویژگی‌های نهان[1] و استفاده از آن‌ها یا استفاده از سایر ویژگی‌های مرتبه بالا شدند. مدل‌های ژرف، در صورتی که داده‌های کافی در اختیار داشته باشند، قادر خواهند بود ویژگی‌های نهان مفیدی ساخته و آن‌ها را برای محاسبه‌ی متغیر هدف به کار ببرند؛ در نتیجه بسیاری از پژوهش‌های پیشین برای پیش‌بینی نرخ کلیک، از این مزیت بهره جسته‌اند.

ویژگی‌های مرتبه بالا در مدل‌های پیش‌بینی نرخ کلیک، شامل تعامل‌های مرتبه بالا بین بردارهای تعبیه و همچنین ویژگی‌های نهان که در برخی پژوهش‌ها به آن‌ها تعامل‌های ضمنی[2] نیز گفته می‌شود، هستند. استدلال این نامگذاری، این نکته است که مقادیر تعامل، به صورت صریح[3] فرموله‌بندی و محاسبه می‌شوند. در حالی که مدل‌های ژرف، می‌توانند ویژگی‌های نهانی محاسبه‌کنند که عملا تفاوتی با مقادیر تعامل بین ویژگی‌ها ندارند، اما فرموله بندی صریحی برای آن‌ها وجود ندارد؛ در نتیجه مدل‌های ژرف بر حسب نیاز، این ویژگی‌ها را استخراج کرده و از آن‌ها استفاده می‌کنند؛ لذا می‌توان این ویژگی‌ها را نسخه‌ی غیر صریح یا ضمنی (و همچنین پیچیده‌تر) مفهوم تعامل در نظر گرفت.

ویژگی‌های مرتبه بالا برای یادگیری، به داده‌های بیشتری نیاز داشته و شامل اطلاعات بیشتری هستند؛ اما آموزش آن‌ها علاوه بر محاسبات بیشتر، نیاز به طراحی دقیق‌تر و چالش‌های مختلف، به مراقبت ویژه در مقابل خطر بیش‌برازش نیاز دارند.

همانطور که گفته شد، در بسیاری از پژوهش‌های ژرف پیشین، از هر دو دسته‌ی این ویژگی‌ها استفاده می‌شود. دسته‌ی اول، شکل کلی تابع تصمیم‌گیری را ترسیم کرده و دسته‌ی دوم، به مدل کمک می‌کنند که این تابع را به طرز دقیق‌تری شکل داده و انعطاف کافی برای مدل‌سازی را به آن بیافزاید.

در این پژوهش نیز، از همین شیوه بهره جسته و از دو دسته ویژگی مختلف برای استفاده‌ی شبکه‌ی سر استفاده می‌کنیم. انتظار داریم این عمل هم در شرایط شروع سرد و هم در مقابل مشکل بیش‌برازش باعث بهبود کلی عملکرد مدل شود؛ پس ورودی شبکه‌ی سر، شامل دو بخش است:

[1] Latent Features
[2] Implicit Interactions
[3] Explicit

- **بردارهای تعبیه**

شبکه‌ی سر، برای پیش‌بینی نرخ کلیک، نیاز به ویژگی‌های مرتبه پایین دارد. به دلیل عدم استفاده از جملات رگرسیون خطی، تنها ویژگی‌های مرتبه پایینی که در اختیار داریم، خود بردارهای تعبیه است؛ بنابراین همه‌ی بردارهای تعبیه را به هم چسبانده و به عنوان ورودی اول شبکه‌ی سر استفاده می‌کنیم.

- **بردارهای تعامل**

همچنین، شبکه‌ی سر برای استخراج ویژگی‌های نهان و تخمین دقیق مرز تصمیم‌گیری، نیاز به ویژگی‌های مرتبه بالا دارد. برای تامین این ویژگی‌های مرتبه‌ی بالا، همه‌ی بردارهای تعامل که توسط شبکه‌های تعامل محاسبه شده‌اند را به هم چسبانده و به عنوان ورودی دوم شبکه‌ی سر استفاده می‌کنیم.

شبکه‌ی سر، یک شبکه‌ی عصبی چند لایه است که بجز لایه‌ی آخر، تعداد نورون‌های همه‌ی لایه‌های آن ثابت بوده و در آن از واحدهای خطی یکسوکننده‌ی نشت کننده به عنوان تابع فعال‌ساز استفاده می‌کنیم. تعداد لایه‌های این شبکه را با $Depth_{HeadNet}$ و تعداد نورون‌های هر لایه را با $Width_{HeadNet}$ نشان می‌دهیم. لایه‌ی آخر این شبکه برای محاسبه‌ی احتمال کلیک، دارای تنها یک نورون بوده و از تابع فعال‌ساز سیگموید برای آن استفاده می‌کنیم.

$$\hat{y} = HeadNet(e_1 : e_2 : \cdots : e_f : I_{1,2} : I_{1,3} : \cdots : I_{f-1,f}) \tag{3.21}$$

چون مساله‌ی پیش‌بینی احتمال کلیک، جزو مسائل دسته‌بندی دو کلاسه است، پس می‌توانیم مدل پیشنهادی را با تابع هزینه‌ی خطای لگاریتمی آموزش دهیم. خطای لگاریتمی از طریق رابطه‌ی زیر محاسبه می‌شود:

$$LogLoss(y, \hat{y}) = -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y}) \tag{3.22}$$

همانطور که از رابطه‌ی خطای لگاریتمی مشخص است، این تابع برای نمونه‌های دو دسته، وزن یکسان در نظر می‌گیرد؛ اما در نظر گرفتن وزن یکسان برای هر دو دسته، در شرایطی که عدم توازن بین دسته‌ها وجود دارد، می‌تواند باعث شود مرز تصمیم گیری به سمت دسته‌ی اقلیت[1] حرکت کرده و در نتیجه عملکرد مدل را برای نمونه‌های این دسته تضعیف کند. برای مقابله با این مشکل، روش‌های متعددی ارائه شده است. در این پژوهش، از وزن‌دهی تابع خطا استفاده می‌کنیم. وزن‌دهی تابع خطا به این صورت است که خطای نمونه‌های کلاس اکثریت را در یک ضریب کوچک و خطای نمونه‌های دسته‌ی اقلیت را در یک ضریب بیشتر ضرب می‌کنیم؛ در نتیجه میزان تاثیر دو کلاس بر تابع خطا یکسان شده و در نتیجه مشکل عدم توازن بین کلاس‌ها تا حدود زیادی حل می‌شود. خطای لگاریتمی در صورتی که از وزن دهی استفاده کنیم، به شکل زیر در می‌آید:

[1] Minority Class

$$LogLoss(y, \hat{y}) = -y \log(\hat{y}) W_{Click} - (1 - y) \log(1 - \hat{y})(1 - W_{Click}) \tag{3.23}$$

که در آن W_{Click} نسبت تعداد نمونه‌های کلیک نشده در کل مجموعه‌ی داده است.

تمامی قسمت‌های روش پیشنهادی، مشتق پذیر هستند. پس می‌توانیم همه‌ی این قسمت‌ها را سر تا سر[1] با روش‌های گرادیان کاهشی دسته‌ای[2] آموزش دهیم. برای انجام این کار، از روش آدام[3] استفاده می‌کنیم.

۳-۴ جمع‌بندی روش پیشنهادی

در جدول ۳-۱ با جمع‌بندی ایده‌های استفاده شده در روش پیشنهادی و مزایای آن‌ها در مقابل چالش‌های مساله و همچنین معایب یا چالش‌های احتمالی هر کدام، این فصل را به پایان می‌بریم.

[1] End To End
[2] Mini-Batch Gradient Descent
[3] ADAptive Moment estimation

جدول ۳-۱: خلاصه‌ی ایده‌های استفاده شده در روش پیشنهادی

تکنیک مورد استفاده	چالش مورد نظر	مزایا در مقابل این چالش	معایب و محدودیت‌ها
تعبیه در ابعاد متفاوت	ابعاد بالا	کاهش پارامترهای غیر ضروری و جلوگیری از خطر بیش‌برازش	استفاده از عملگرهای معمول برای محاسبه‌ی تعامل امکان پذیر نیست
	شروع سرد	افزایش سرعت یادگیری به کمک تعبیه‌ی موثر	
	سرعت آموزش	پارامترهای کمتر و افزایش سرعت آموزش	نیاز به مشخص کردن رابطه‌ای برای ابعاد بردارهای تعبیه‌ی هر فیلد
	سرعت اجرا	محاسبات کمتر و افزایش سرعت پیش‌بینی	
محاسبه‌ی تعامل به کمک شبکه‌های عصبی چند لایه	مدل‌سازی بهتر	استخراج تعامل‌ها با پیچیدگی بیشتر	افزایش جزئی خطر بیش‌برازش
	شروع سرد	بهره گیری از فضای چگال بردارهای تعبیه برای محاسبه‌ی تعامل	لزوم استفاده از تنظیم
	سرعت اجرا	وجود پیاده‌سازی‌های سریع برای شبکه‌های عصبی چند لایه	
تعامل‌های چند بعدی	مدل‌سازی بهتر	وجود اطلاعات بیشتر در بردارهای تعامل	افزایش جزئی خطر بیش‌برازش
	مدل‌سازی بهتر	عدم نیاز به تعامل‌های چندگانه	
ترکیب بردارهای تعبیه و تعامل	مدل‌سازی بهتر	وجود اطلاعات مفید در ویژگی‌های مرتبه پایین و مرتبه بالا	افزایش جزئی خطر بیش‌برازش
	شروع سرد	حضور ویژگی‌های مرتبه پایین در صورت عدم حضور ویژگی‌های مرتبه بالا	
	سرعت آموزش	رسیدن گرادیان از مسیرهای متعدد به بردارهای تعبیه	
وزن دهی تابع خطا	عدم توازن بین دسته‌ها	جلوگیری از بایاس شدن مدل به سمت دسته‌ی اکثریت	کاهش جزئی سرعت همگرایی

فصل ۴

یافته‌های پژوهش

در این بخش ابتدا مجموعه‌های داده‌ی مورد استفاده را معرفی کرده و مختصرا در مورد خصوصیات آن‌ها بحث می‌کنیم؛ سپس برخی از معیارهای ارزیابی مهم در این حوزه را معرفی کرده و دلایل انتخاب این معیارها را شرح می‌دهیم. پس از آن، روش‌های مورد استفاده در این پژوهش برای تنظیم پارامترها را توضیح می‌داده و با انجام آزمایش‌هایی، بهترین مقادیر را برای ابرپارامترهای مدل به دست می‌آوریم؛ سپس با طراحی و اجرای چندین آزمایش، عملکرد روش پیشنهادی را با برخی از روش‌های پیشین مقایسه کرده و به برخی سوالات احتمالی پاسخ می‌دهیم.

۴-۱ مجموعه‌های داده

در این بخش به معرفی و بررسی مجموعه داده‌های مورد استفاده در این پژوهش می‌پردازیم. لازم به ذکر است پیش‌پردازش‌های مختلفی که به خاطر محدودیت‌های سخت‌افزاری اعمال می‌کنیم، باعث می‌شود نتوانیم نتایج به دست آمده را با نتایج گزارش شده توسط پژوهش‌های پیشین مقایسه کنیم. پس به کمک پیاده‌سازی‌های موجود از این پژوهش‌ها، عملکرد آن‌ها را روی مجموعه‌های داده‌ی ایجاد شده محاسبه خواهیم کرد.

۴-۱-۱ آوت‌برین

همانطور که گفته شد، در سال ۲۰۱۶ شرکت آوت‌برین با برگزاری یک چالش در سایت ککل [2]، مجموعه‌داده‌ی خود را منتشر کرد. در این مجموعه داده، هر بار که کاربری به صفحه‌ی سایت یک ناشر مراجعه کرده است، ۲ الی ۱۲ بنر تبلیغاتی به وی نمایش داده شده، که کاربر روی یکی از آن‌ها کلیک کرده است. میانگین تعداد

[2] Kaggle

تبلیغ در این مجموعه‌ی داده، ۱۶.۵ تبلیغ در هر مراجعه است.

یکی از ویژگی‌های مجموعه‌داده‌ی آوت‌برین، وجود اطلاعات جانبی متنوع در مورد صفحاتی است که تبلیغات در آن‌ها به نمایش گذاشته شده‌اند. این صفحات طبق یک طبقه‌بندی موضوعی، به ۹۷ دسته تقسیم شده‌اند. اطلاعاتی نیز در مورد ذکر شدن نام برخی موجودیت‌ها در هر صفحه و میزان اطمینان در مورد آن فراهم شده است. اطلاعات متنوعی نیز از نویسنده، ناشر و زمان انتشار هر صفحه وجود دارد. همچنین اطلاعات مربوط به تبلیغ کننده و کمپین تبلیغاتی برای هر تبلیغ نیز موجود است.

در این مجموعه داده، اطلاعات حجیمی نیز در مورد مشاهده‌ی صفحات مختلف توسط کاربران ارائه شده است. این اطلاعات شامل زمان دقیق مراجعه، پلتفرم (کامپیوتر، موبایل یا تبلت)، محل جغرافیایی و منبع ترافیک (مستقیم، جستجو یا شبکه‌های اجتماعی) هر بازدید هستند. این اطلاعات به دلیل حجم بالا و تعداد زیاد (نزدیک به ۲ میلیارد) بازدید از صفحات مختلف وب بسیار حجیم هستند. این مجموعه‌داده، شامل اطلاعات جمع آوری شده در طول دو هفته (۱۴ روز) از بازدیدها، نمایش تبلیغ‌ها و کلیک‌ها در تعدادی سایت پر بازدید است.

در این مجموعه‌داده، همه‌ی اطلاعات به صورت ناشناس شده[1] ارائه شده و حتی نام سایت‌ها، نوع دسته‌های موضوعی صفحات نیز ذکر نشده و اطلاعات آن به صورت شناسه‌ی گمنام در اختیار محققین قرار گرفته است. تنها ویژگی غیر ناشناس در این مجموعه‌ی داده، موقعیت جغرافیایی کاربران است که البته برای حفظ حریم خصوصی کاربران، به سطح کشور یا استان / ایالت محدود شده است.

آوت‌برین پیش‌پردازش شده

همانطور که گفته شد، تعداد بسیار زیاد ویژگی‌ها و داده‌ها، باعث بروز محدودیت‌های سخت‌افزاری متعددی در انجام آزمایش روی این مجموعه‌ی داده می‌شود؛ به همین دلیل در این پژوهش با حذف تعداد زیادی از این ویژگی‌ها، نسخه‌ی سبک‌تری از این مجموعه‌ی داده استخراج کرده و از آن به عنوان یک مجموعه‌داده‌ی کوچک برای انجام مقایسه‌ها بهره می‌جوییم.

در اولین قدم، تعداد داده‌های موجود در این مجموعه‌ی داده را به کمک روش‌های نمونه برداری، به ۸۷ میلیون کاهش می‌دهیم؛ سپس ویژگی‌هایی از این مجموعه‌ی داده که دسته‌ای نیستند را حذف می‌کنیم. همچنین، تعدادی از ویژگی‌های دسته‌ای که تعداد دسته‌های بسیار زیادی دارند را از این مجموعه‌ی داده حذف می‌نماییم. تعداد ویژگی‌های دسته‌ای باقی مانده در این مجموعه‌ی داده ۱۲ بوده و این ویژگی‌ها شامل موارد: شناسه‌ی کمپین تبلیغاتی، شناسه‌ی تبلیغ کننده، پلتفرم، موقعیت جغرافیایی، شناسه‌ی صفحه، شناسه‌ی ناشر، شناسه‌ی موضوع صفحه، شناسه‌ی دسته‌ی صفحه، شناسه‌ی صفحه‌ی منبع، شناسه‌ی ناشر صفحه‌ی منبع، شناسه‌ی موضوع صفحه‌ی منبع و شناسه‌ی دسته‌ی صفحه‌ی منبع هستند.

مجموع تعداد ویژگی‌های باینری استخراج شده از این مجموعه‌ی داده ۵۳۷۲۷ است. لازم به ذکر است سبک بودن این مجموعه داده، به دلیل تعداد کم داده‌ها نیست؛ بلکه این مجموعه‌ی داده به این دلیل سبک خوانده می‌شود که تعداد ویژگی‌های آن بسیار کمتر از سایر مجموعه‌های داده است.

[1] Anonimized

حدود ۱۹ درصد از داده‌های این مجموعه در دسته‌ی کلیک شده و بقیه‌ی داده‌ها در دسته‌ی کلیک نشده طبقه بندی شده‌اند.

۴-۱-۲ کرایتیو

یکی از شرکت‌های فعال در حوزه‌ی تبلیغات نمایشی آنلاین، کرایتیو[1] است. این شرکت با استفاده از مزایده‌های بلادرنگ تبلیغات مشتریان خود (سکوی نیاز) را بین مشتریان دیگر خود (سکوی تامین) توزیع می‌کند. در سال ۲۰۱۴ این شرکت اطلاعات مربوط به ۷ روز از ترافیک خود را در قالب یک چالش در سایت کگل منتشر کرد.

این مجموعه‌داده، از ۱۳ ویژگی (ناشناس) عددی، که طبق اعلام خود شرکت اکثر این ویژگی‌ها از نوع تعداد هستند؛ و ۲۶ ویژگی ناشناس دسته‌ای، که به صورت درهم‌سازی شده[2] ارائه شده اند، تشکیل شده است. این مجموعه‌داده، شامل تعدادی مقادیر گم شده[3] بوده و مانند مجموعه‌داده‌ی آوت‌برین، اطلاعات آن به صورت ناشناس ارائه شده‌اند.

این مجموعه‌ی داده شامل بیش از ۴۵ میلیون رکورد بوده که کاربران در ۲۶ درصد از این نمونه‌ها روی بنر تبلیغاتی کلیک کرده‌اند. با وجود کمتر بودن تعداد داده‌ها در این مجموعه‌ی داده و شدت کمتر عدم توازن بین کلاس‌ها، تعداد ویژگی‌های زیاد و همچنین تنک بودن بسیاری از این ویژگی‌ها باعث می‌شوند این مجموعه‌ی داده یک چالش واقعی برای روش‌های پیش‌بینی نرخ کلیک به شمار رود.

مجموع تعداد ویژگی‌های باینری استخراج شده از بخش دسته‌ای این مجموعه‌ی داده، به بیش از ۳۳ میلیون می‌رسد؛ بنابراین استفاده از همه‌ی این ویژگی‌ها محدودیت‌های سخت‌افزاری زیادی را به وجود می‌آورد. به همین دلیل، مجموعه‌های داده‌ی کرایتیو-۲۲، کرایتیو-۲۱ و کرایتیو-۲۰ را از این مجموعه‌ی داده استخراج کرده و تمامی آزمایش‌های مربوطه را با این سه مجموعه‌ی داده انجام می‌دهیم.

کرایتیو-۲۲

با حذف ۴ ویژگی دسته‌ای که بیشترین کاردینالیتی را دارند، از مجموعه داده‌ی کرایتیو و همچنین حذف همه‌ی ویژگی‌های عددی که با ساختار روش پیشنهادی و اکثر روش‌های پیشین سازگار نیستند، به مجموعه‌ی داده‌ی کرایتیو-۲۲ می‌رسیم. مجموع تعداد ویژگی‌های باینری استخراج شده از این مجموعه‌ی داده، تا حدود ۷.۲ میلیون کاهش می‌یابد.

[1] Criteo
[2] Hashed
[3] Missing values

کرایتیو-۲۱

مثل مجموعه داده‌ی کرایتیو-۲۲، مجموعه داده‌ی کرایتیو-۲۱ هم از مجموعه داده‌ی کرایتیو ساخته می‌شود. در مجموعه داده‌ی کرایتیو-۲۱، ۵ ویژگی دارای کاردینالیتی بیشتر را حذف کرده و مجموع تعداد ویژگی‌های باینری را به حدود ۵۷۰ هزار می‌رسانیم.

کرایتیو-۲۰

همانطور که انتظار می‌رود، مجموعه داده‌ی کرایتیو-۲۰، دارای تنها ۲۰ ویژگی دسته‌ای بوده و مجموع تعداد ویژگی‌های باینری در آن حدود ۲۸۰ هزار است.

لازم به تذکر است که تعداد داده‌ها و درصد کلی کلیک در هر سه مجموعه داده‌ی ساخته شده یکسان و برابر با مجموعه داده‌ی کرایتیو است.

۴-۲ معیارهای ارزیابی

همانطور که در فصل اول گفته شد، مساله‌ی پیش‌بینی نرخ کلیک به دلیل ویژگی‌های متعدد، از جمله عدم توازن کلاس‌ها، پر تعداد اما تنک بودن ویژگی‌ها و برخی مشکلات دیگر، مساله‌ای خاص است؛ لذا برای ارزیابی راه‌حل‌هایی که برای این مساله پیشنهاد می‌شوند، به معیارهای ارزیابی به خصوصی نیاز داریم. در این بخش به معرفی معیارهای ارزیابی مورد استفاده در این پژوهش می‌پردازیم و دلیل استفاده از برخی از این معیارها را بیان می‌نماییم.

۴-۲-۱ خطای لگاریتمی

خطای لگاریتمی یا آنتروپی متقابل[1]، یکی از مهمترین معیارهای استفاده‌شده در حوزه‌ی پیش‌بینی نرخ کلیک است. در مدل‌هایی که خروجی آن‌ها برابر احتمال کلیک است، مقدار این خطا، با قرینه‌ی لگاریتم درستنمایی[2] این مدل‌ها برابر است. در نتیجه با شهود و درک احتمالاتی از این مساله کاملا تطابق دارد.

در صورتی که از این خطا استفاده کنیم، حتی اگر داده‌ای توسط مدل درست دسته‌بندی شود، باز هم امکان دارد به آن خطایی تخصیص دهیم. در صورتی خطای آنتروپی متقابل برابر صفر می‌شود که علاوه بر طبقه‌بندی درست همه‌ی داده‌ها، به همه‌ی آن‌ها احتمال کاملا باینری اختصاص دهد؛در نتیجه مدل به سمتی پیش می‌رود که خطا در احتمال پیش‌بینی‌شده را کمتر و کمتر کند.

[1] Cross Entropy
[2] Negative Log Likelihood

خطای لگاریتمی به دلیل مشتق پذیر بودن، می‌تواند به عنوان تابع هزینه‌ی مدل‌هایی که از گرادیان کاهشی استفاده می‌کنند، به کار گرفته شود. همانطور که در فصل قبل گفته شد، با وزن‌دار کردن این تابع خطا، می‌توانیم مدل‌ها را نسبت به عدم توازن بین کلاس‌ها مقاوم کنیم.

۴-۲-۲ مساحت تحت منحنی

در ادبیات یادگیری ماشین، معیارهای نرخ مثبت درست[1] و نرخ مثبت غلط[2] کاربردهای بسیاری دارند. نرخ مثبت درست به نرخ عملکرد صحیح در کلاس مثبت و نرخ مثبت غلط به نرخ عملکرد اشتباه در کلاس منفی اشاره می‌کنند. این مقادیر طبق تعریف، همیشه بین صفر و یک هستند. در مدل‌هایی که برای دسته‌بندی دو کلاسه، از یک حد آستانه بهره می‌جویند، با تغییر دادن مقدار حد آستانه، می‌توانند تعادلی بین نرخ مثبت درست و نرخ مثبت غلط بیابند.

یک منحنی پر کاربرد در یادگیری ماشین، منحنی راک[3] است. برای رسم این منحنی، ابتدا مدل را روی همه‌ی داده‌های تست اجرا کرده و مقادیر احتمال را برای همه‌ی داده‌ها به دست می‌آوریم؛ سپس آن‌ها را بر اساس احتمال کلیک صعودی، مرتب می‌کنیم. از نقطه‌ی بالا و راست منحنی شروع کرده و هر بار، در صورتی که داده‌ی مربوطه جزو کلاس منفی باشد، یک گام به سمت چپ و در صورتی که مربوط به کلاس مثبت باشد، یک گام به سمت پایین حرکت می‌کنیم. اندازه‌ی گام‌های به سمت چپ، برابر معکوس تعداد داده‌های منفی و اندازه‌ی گام‌های به سمت راست، برابر معکوس تعداد داده‌های مثبت است؛ لذا پس از مشاهده‌ی همه‌ی داده‌ها، باید به نقطه‌ی چپ و پایین منحنی رسیده باشیم. هر قدر این منحنی به سمت بالا و چپ تمایل داشته‌باشد، به این معنی است که مدل تحت حد آستانه‌های مختلف، عملکرد متوازن و قابل قبولی دارد. همچنین نرمال بودن نرخ مثبت درست و نرخ مثبت غلط باعث می‌شود هیچ مشکلی از جانب غیر متناسب بودن کلاس‌ها عملکرد این معیار را تهدید نکند. در صورتی که مساحت تحت پوشش منحنی راک را محاسبه کنیم، می‌توانیم از آن به عنوان یک معیار عددی کاملا مناسب برای نظارت بر مدل‌های یادگیری ماشین استفاده کنیم. مساحت تحت منحنی راک یا مساحت تحت منحنی[4] عددی نرمال بین صفر و یک بوده ولی مقادیر کمتر از نیم برای آن غیر معقول است.

یکی از نکات مهم در مورد معیار مساحت تحت منحنی، تعبیر احتمالاتی آن است. این معیار نشان دهنده‌ی احتمال تخصیص امتیاز (احتمال کلیک) بیشتر به یک نمونه‌ی (تصادفی) از کلاس مثبت، نسبت به یک نمونه‌ی (تصادفی) از کلاس منفی است. به عنوان مثال، اگر میزان مساحت تحت منحنی برای یک مدل، برابر ۷۵ درصد باشد، اگر یک داده‌ی تصادفی از کلاس مثبت و یک داده‌ی تصادفی از کلاس منفی انتخاب کرده و امتیاز این مدل برای این دو داده را محاسبه کنیم، به احتمال ۷۵ درصد، امتیاز تخصیص داده شده به داده‌ی کلاس مثبت، بیشتر از احتمال تخصیص داده شده به داده‌ی کلاس منفی خواهد بود. این خاصیت مهم، باعث می‌شود مدلی که مساحت تحت منحنی بیشتری دارد، برای اعمالی نظیر مرتب کردن اولویت‌دار،

[1] TPR
[2] FPR
[3] ROC
[4] Area Under Curve

عملکرد بهتری داشته باشند. چون مساله‌ی پیش‌بینی نرخ کلیک، در تبلیغات نمایشی عملا برای مرتب کردن اولویت‌دار بنرهای تبلیغاتی، بر اساس احتمال کلیک کاربران بر روی آن‌ها طراحی شده است، لذا مدلی که مساحت تحت منحنی قابل قبولی داشته باشد، برای استفاده‌ی صنعتی دراین مساله گزینه‌ی مناسبی خواهد بود.

همه‌ی دلایل ذکر شده، باعث می‌شوند در این پژوهش از این معیار به عنوان معیار اصلی ارزیابی مدل استفاده کنیم.

۴-۳ روش‌های تنظیم پارامترها

هر یک از روش‌های ژرف که در فصل دوم معرفی کردیم و همچنین بسیاری از روش‌های دیگر، به دلیل جلوگیری از بیش‌برازش، از روش‌های تنظیم پارامترها استفاده می‌کنند. در این بخش چند روش تنظیم پارامتر که در این پژوهش استفاده کرده‌ایم را به طور مختصر معرفی کرده و با انجام آزمایش‌هایی، بهترین مقادیر ابرپارامترهای مربوط به آن‌ها را انتخاب می‌کنیم.

۴-۳-۱ تنظیم مرتبه‌ی دوم

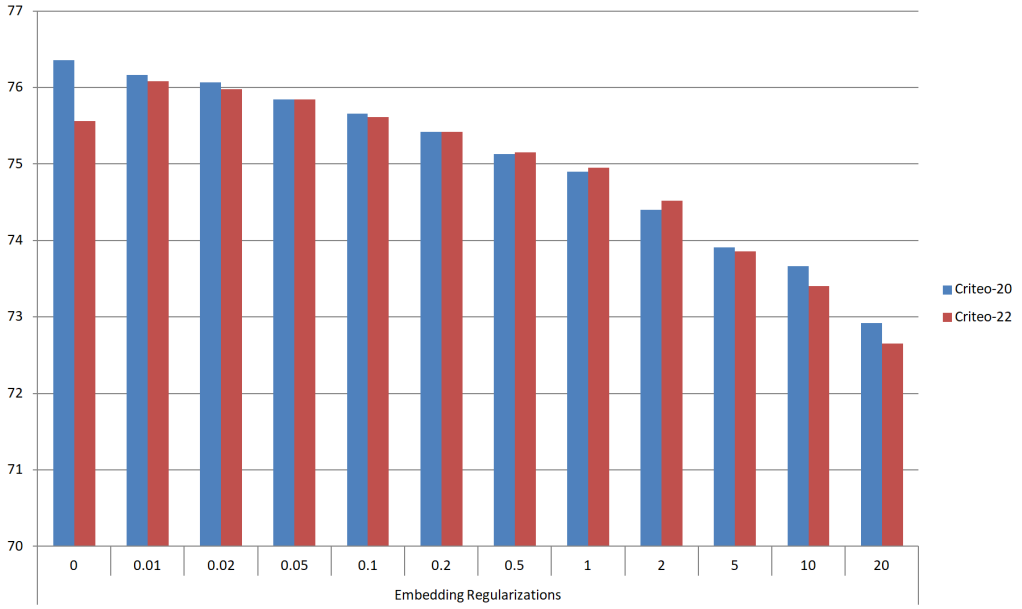
در تنظیم مرتبه‌ی دوم، میزان خطای نهایی مدل را با ضریبی از مجموع توان دوم مقادیر پارامترهای مدل جمع می‌کنند. این عمل باعث می‌شود مدل به استفاده از پارامترهای کوچک‌تر ترغیب شود، که این امر به نوبه‌ی خود باعث کاهش پیچیدگی مدل و همچنین کاهش خطر بیش‌برازش می‌شود. تنظیم مرتبه‌ی دوم را می‌توان در قسمت‌های مختلف مدل از قبیل پارامترهای تعبیه، پارامترهای شبکه‌های تعامل و همچنین پارامترهای شبکه‌ی سر اعمال کرد.

تنظیم مرتبه‌ی دوم روی پارامترهای تعبیه

با اعمال تنظیم مرتبه‌ی دوم بر پارامترهای تعبیه، مدل را به استفاده از بردارهای تعبیه‌ی کوچک‌تر ترغیب می‌کنیم. این عمل باعث ساده‌تر شدن فضاهای تعبیه‌ی مدل شده و در نتیجه خطر بیش‌برازش مدل را کاهش می‌دهد.

در آزمایشی چندین مقدار مختلف برای ضریب تنظیم مرتبه‌ی دوم روی پارامترهای تعبیه در نظر گرفته و مدل پیشنهادی را روی مجموعه‌های داده‌ی کرایتیو-۲۰ و کرایتیو-۲۲ آموزش دادیم. شکل ۴-۱ مقادیر مساحت تحت نمودار در این آزمایش را نشان می‌دهد.

همانطور که مشخص است، برای مجموعه داده‌ی کرایتیو-۲۰، که تعداد ویژگی کمتری دارد، تنظیم مرتبه‌ی دوم پارامترهای تعبیه کمکی به عملکرد مدل نمی‌کند؛ اما مقادیر بسیار اندک در ضریب تنظیم مرتبه‌ی دوم



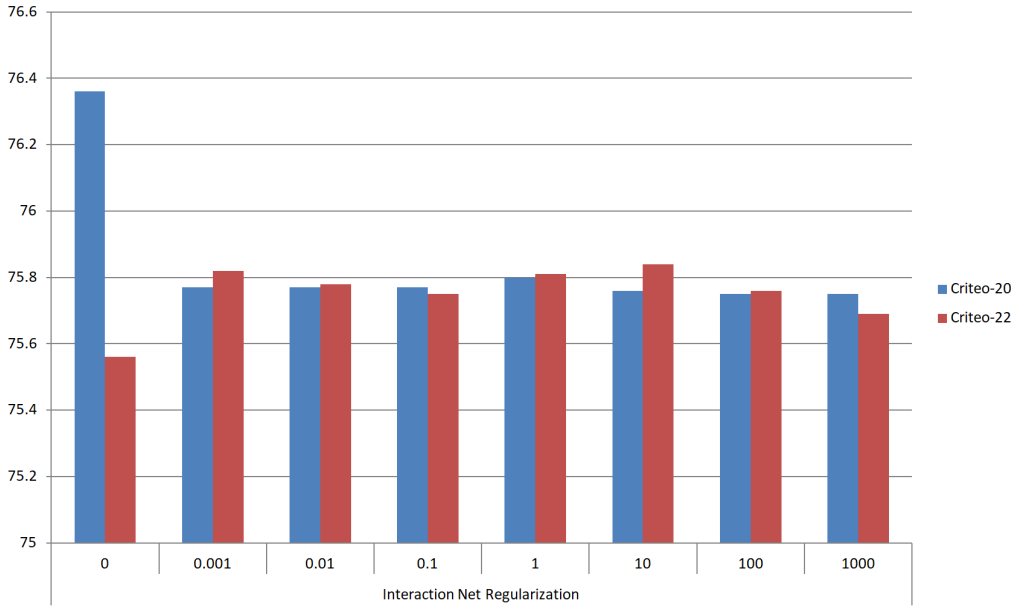
شکل ۴-۱: مساحت تحت نمودار، به ازای مقادیر مختلف ضریب تنظیم مرتبه‌ی دوم برای پارامترهای تعبیه‌ی مدل

روی پارامترهای تعبیه، باعث بهبود عملکرد مدل روی مجموعه داده‌ی کرایتیو-۲۲ می‌شود. از این آزمایش این نتیجه را برداشت می‌کنیم که تنظیم مرتبه‌ی دوم، در مجموعه‌های داده‌ی با تعداد ویژگی زیاد، می‌تواند خطر بیش‌برازش را کاهش دهد.

تنظیم مرتبه‌ی دوم روی پارامترهای شبکه‌های تعامل

شبکه‌های تعامل، بخش مهمی از پیچیدگی مدل پیشنهادی را ایجاد می‌کنند. با اعمال تنظیم مرتبه‌ی دوم روی پارامترهای این شبکه‌ها، مدل را به استخراج روابط ساده و موثر بین بردارهای تعبیه ترغیب کرده و انتظار داریم این کار خطر بیش‌برازش مدل را کاهش دهد.

در آزمایشی، چندین مقدار مختلف برای ضریب تنظیم مرتبه‌ی دوم روی پارامترهای شبکه‌های تعامل در نظر گرفته و مدل پیشنهادی را روی مجموعه‌های داده‌ی کرایتیو-۲۰ و کرایتیو-۲۲ آموزش می‌دهیم. شکل ۴-۲ مقادیر مساحت تحت نمودار در این آزمایش را نشان می‌دهد. همانطور که مشخص است، برای مجموعه داده‌ی کرایتیو-۲۰، که تعداد ویژگی کمتری دارد، تنظیم مرتبه‌ی دوم پارامترهای شبکه‌های تعامل، عملکرد مدل را تضعیف می‌کند؛ اما مقادیر متوسط ضریب تنظیم مرتبه‌ی دوم روی پارامترهای شبکه‌های تعامل، باعث بهبود عملکرد مدل روی مجموعه داده‌ی کرایتیو-۲۲ می‌شود. از این آزمایش نیز برداشت می‌کنیم که تنظیم مرتبه‌ی دوم، در مجموعه‌های داده‌ی با تعداد ویژگی زیاد، موثر بوده و خطر بیش‌برازش را کاهش می‌دهد.



شکل ۴-۲: مساحت تحت نمودار، به ازای مقادیر مختلف ضریب تنظیم مرتبه‌ی دوم برای پارامترهای شبکه‌های تعامل

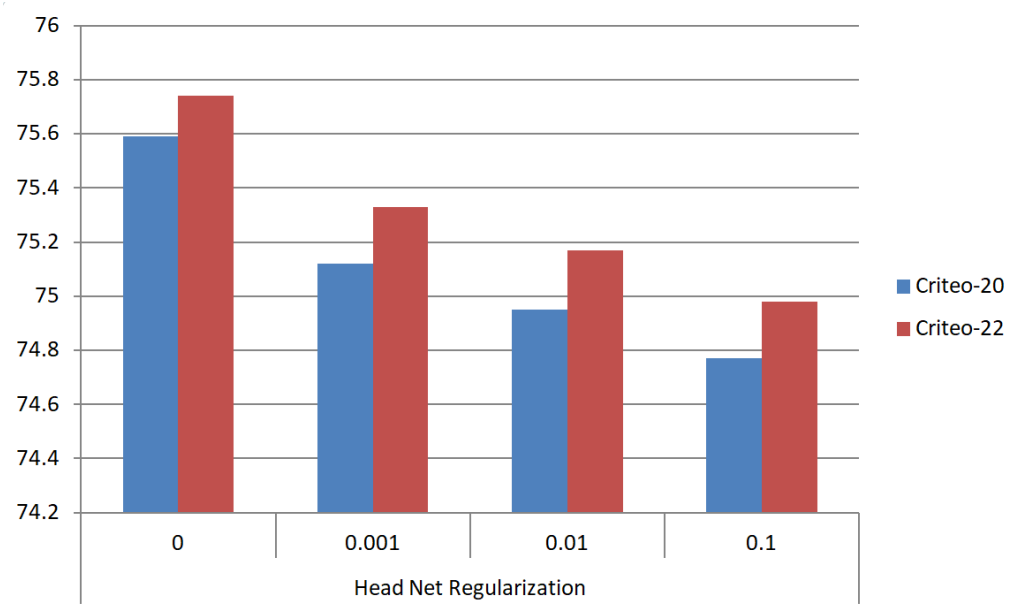
تنظیم مرتبه‌ی دوم روی پارامترهای شبکه‌ی سر

شبکه‌ی سر، نقش مهم استخراج ویژگی‌های سطح بالا از روی بردارهای تعبیه و همچنین بردارهای تعامل مدل را دارد؛ بنابراین با انجام عمل تنظیم مرتبه‌ی دوم روی پارامترهای آن، سعی در کاهش خطر بیش‌برازش مدل می‌نماییم.

در آزمایشی، چندین مقدار مختلف برای ضریب تنظیم مرتبه‌ی دوم روی پارامترهای شبکه‌ی سر در نظر گرفته و مدل پیشنهادی را روی مجموعه‌های داده‌ی کرایتیو-۲۰ و کرایتیو-۲۲ آموزش می‌دهیم. شکل ۴-۳ مقادیر مساحت تحت نمودار در این آزمایش را نشان می‌دهد. همانطور که مشخص است و بر خلاف تصور اولیه، اعمال تنظیم مرتبه‌ی دوم روی پارامترهای شبکه‌ی سر، بر بهبود عملکرد مدل در هیچ یک از مجموعه‌های داده‌ی کرایتیو-۲۰ و کرایتیو-۲۲ کمک نمی‌کند. این نتیجه می‌تواند به این دلیل رخ دهد که شبکه‌ی سر برای مدل‌سازی مناسب، نیاز به پیچیدگی زیادی داشته و در نتیجه با اعمال ضرایب تنظیم، دچار افت عملکرد می‌شود.

۴-۳-۲ حذف تصادفی

در شبکه‌های عصبی ژرف، برای جلوگیری از خطر بیش‌برازش و همچنین ترغیب مدل‌ها به یادگیری چندگانه و قابل اطمینان، از تکنیک حذف تصادفی استفاده می‌کنند. در حذف تصادفی، مقادیر خروجی برخی از نورون‌های شبکه را در زمان آموزش با صفر جایگزین کرده و در نتیجه میزانی از پیچیدگی مدل را کاهش می‌دهیم. این امر باعث می‌شود شبکه برای حفظ عملکرد خود، همه‌ی ویژگی‌های نهانی که در تصمیم‌گیری مدل موثر هستند را به صورت چندگانه فرا بگیرد. یادگیری چندگانه به این معنی است که به جای یک نورون،



شکل ۴-۳: مساحت تحت نمودار، به ازای مقادیر مختلف ضریب تنظیم مرتبه‌ی دوم برای پارامترهای شبکه‌ی سر

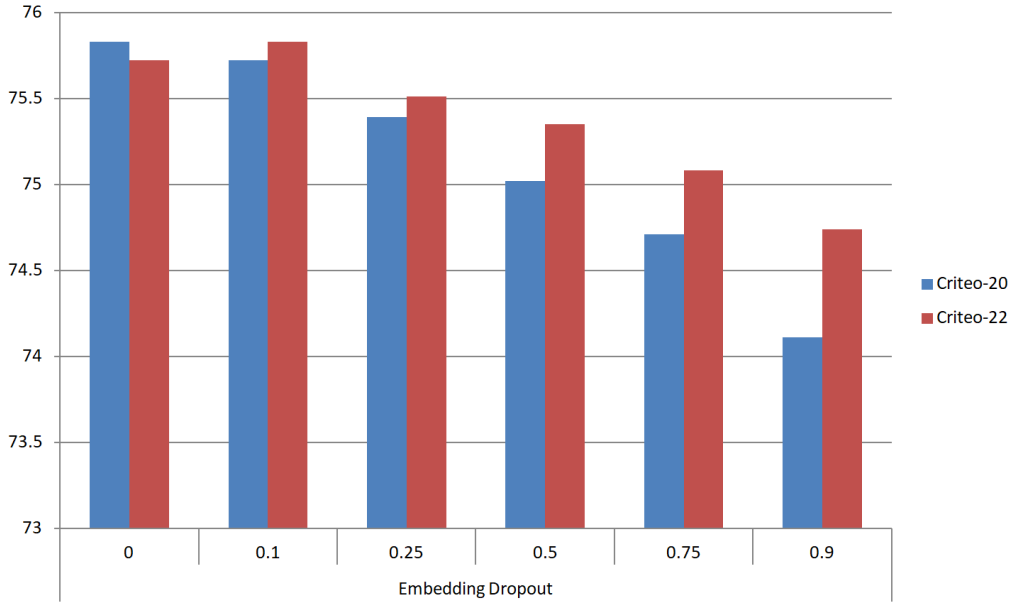
چندین نورون مسئول تشخیص هر ویژگی نهان شده و در نتیجه با حضور یا عدم حضور تنها یکی از ویژگی‌ها، رفتار مدل تفاوت چندانی نمی‌کند. واضح است که این تغییر باعث کاهش واریانس مدل و در نتیجه کاهش خطر بیش‌برازش در مدل می‌شود.

تکنیک حذف تصادفی را می‌توان در قسمت‌های مختلف مدل از جمله بردارهای تعبیه، شبکه‌های تعامل و همچنین شبکه‌ی سر اعمال کرده و انتظار می‌رود مانند تنظیم مرتبه‌ی دوم، باعث بهبود عملکرد مدل در مجموعه‌های داده‌ی حجیم شود.

حذف تصادفی پارامترهای تعبیه

با اعمال تکنیک حذف تصادفی روی پارامترهای تعبیه، باعث کاهش پیچیدگی مدل در این بخش شده و لذا مدل را وادار به یادگیری ساختار ساده‌تر و کاراتر در پارامترهای تعبیه می‌کنیم.

در آزمایشی، با اعمال این تکنیک روی پارامترهای تعبیه، میزان تاثیر آن را بر عملکرد مدل روی مجموعه‌های داده‌ی کرایتیو-۲۰ و کرایتیو-۲۲ اندازه‌گیری می‌کنیم. شکل ۴-۴ مساحت تحت نمودار مدل را در این آزمایش نشان می‌دهد. همانطور که انتظار می‌رفت، مقادیر کم نرخ حذف تصادفی باعث بهبود جزئی عملکرد مدل در مجموعه داده‌ی کرایتیو-۲۲ می‌شوند؛ اما باز هم در مجموعه داده‌ی کرایتیو-۲۰، کوچک بودن مدل باعث می‌شود اعمال تکنیک حذف تصادفی، تاثیر مثبتی بر عملکرد مدل نداشته باشد.



شکل ۴-۴: مساحت تحت نمودار، به ازای مقادیر مختلف نرخ حذف تصادفی در پارامترهای تعبیه‌ی مدل

حذف تصادفی پارامترهای شبکه‌های تعامل

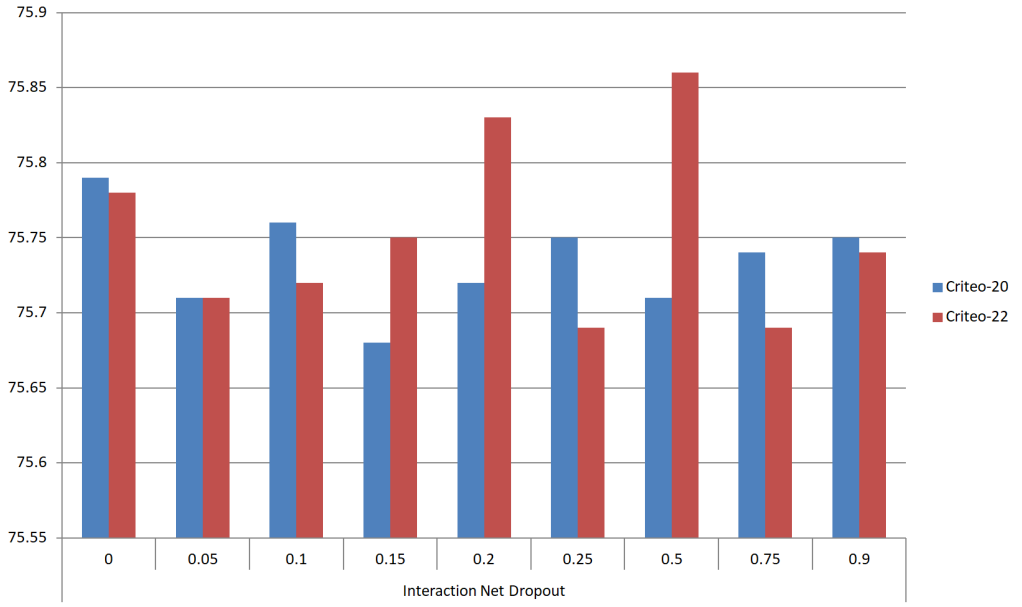
شبکه‌های تعامل به دلیل این که از بردارهای تعبیه استفاده می‌کنند و برخی از بردارهای تعبیه به دلیل چالش شروع سرد، مقادیر مناسبی ندارند، می‌توانند باعث بروز مشکل بیش‌برازش شوند. با اعمال تکنیک حذف تصادفی روی پارامترهای شبکه‌های تعامل، خطر بیش‌برازش مدل را در این بخش‌ها کاهش می‌دهیم.

در آزمایشی، با اعمال این تکنیک روی پارامترهای شبکه‌های تعامل، میزان تاثیر آن را بر عملکرد مدل روی مجموعه‌های داده‌ی کرایتیو-۲۰ و کرایتیو-۲۲ اندازه‌گیری می‌کنیم. شکل ۴-۵ مساحت تحت نمودار مدل را در این آزمایش نشان می‌دهد. در مجموعه داده‌ی کرایتیو-۲۰ به دلیل سادگی مدل، حذف تصادفی پارامترهای شبکه‌های تعامل کمکی به بهبود عملکرد مدل نمی‌کند؛ اما در مجموعه داده‌ی کرایتیو-۲۲، که ابعاد مدل نیز درخور تعداد ویژگی‌های مجموعه‌ی داده رشد کرده است، مقادیر متوسط نرخ حذف تصادفی، باعث بهبود عملکرد مدل می‌شوند. همچنین می‌توانیم رفتار تصادفی تکنیک حذف تصادفی را عامل اصلی ناهموار بودن نتایج در آزمایش فوق در نظر بگیریم.

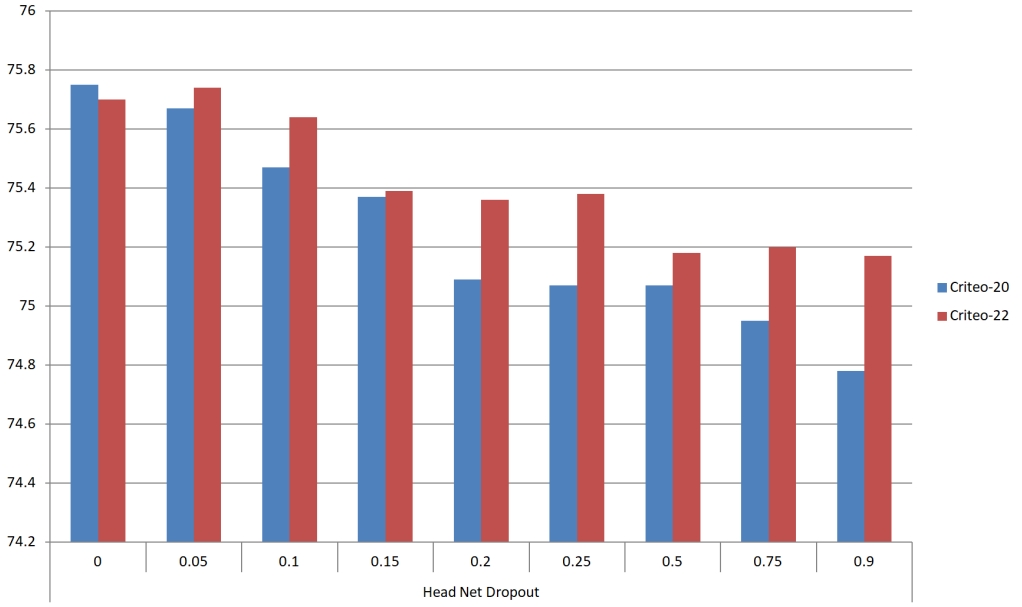
حذف تصادفی پارامترهای شبکه‌ی سر

تکنیک حذف تصادفی، در مدل‌های ژرف کاربرد بیشتری از مدل‌های غیر ژرف دارد؛ در نتیجه انتظار می‌رود تاثیر اعمال این تکنیک در بخش‌های ژرف مدل، احساس شود.

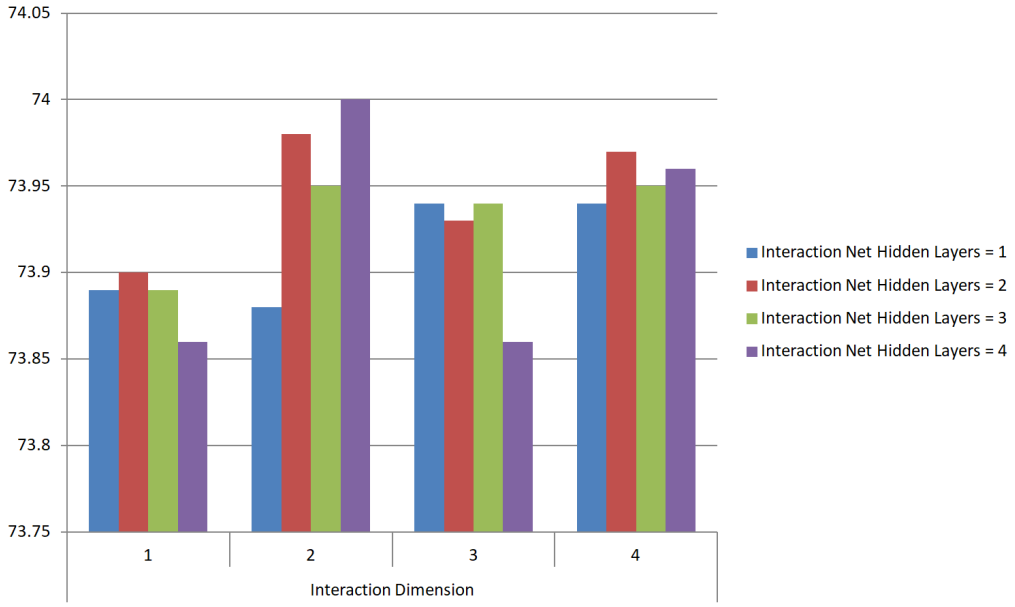
در آزمایشی، با اعمال تکنیک حذف تصادفی روی پارامترهای شبکه‌ی سر، میزان تاثیر آن را بر عملکرد مدل روی مجموعه داده‌های کرایتیو-۲۰ و کرایتیو-۲۲ اندازه‌گیری می‌کنیم. شکل ۴-۶ مساحت تحت نمودار مدل را در این آزمایش نشان می‌دهد. همان‌طور که از نتایج این آزمایش مشخص است، مقادیر اندک نرخ



شکل ۴-۵: مساحت تحت نمودار، به ازای مقادیر مختلف نرخ حذف تصادفی در پارامترهای شبکه‌های تعامل



شکل ۴-۶: مساحت تحت نمودار، به ازای مقادیر مختلف نرخ حذف تصادفی در پارامترهای شبکه‌ی سر



شکل ۴-۷: مساحت تحت نمودار، به ازای تعداد لایه‌های مختلف شبکه‌های تعامل و همچنین مقادیر مختلف بعد بردارهای تعامل

حذف تصادفی باعث بهبود عملکرد مدل بر مجموعه داده‌ی کرایتیو-۲۲ می‌شود؛ اما مثل آزمایش‌های قبل، مجموعه داده‌ی کرایتیو-۲۰ به دلیل سادگی بیش از حد مدل، نیازی به اعمال روش‌های تنظیم احساس نشده و با افزایش نرخ حذف تصادفی، عملکرد مدل پیوسته کاهش می‌یابد.

۴-۴ سایر آزمایش‌ها

در بخش قبل با انجام چندین آزمایش، بهترین مقادیر برای ابرپارامتر[1]های مربوط به تنظیم را یافته و تاثیر اعمال هرکدام از روش‌های تنظیم را بر مدل بررسی کردیم. در این بخش، با طراحی و انجام چند آزمایش دیگر، سایر ابرپارامترهای مدل را بررسی کرده و مقادیر مناسب را برای آن‌ها خواهیم یافت.

۴-۴-۱ تعداد لایه‌های شبکه‌های تعامل و بعد بردارهای تعامل

برای تعیین تعداد لایه‌ها در شبکه‌های تعامل و همچنین بعد بردارهای تعامل، که تنها ابرپارامترهای موجود در ساختار شبکه‌های تعامل هستند، آزمایشی روی مجموعه داده‌ی آوت‌برین پیش‌پردازش شده طراحی و اجرا می‌کنیم. در این آزمایش، تعداد لایه‌های شبکه‌های تعامل را از یک تا چهار تغییر داده و برای هر حالت، بعد بردارهای تعامل را از یک تا چهار تغییر می‌دهیم. نتایج این آزمایش را در شکل ۴-۷ مشاهده می‌کنید. همانطور که از نتایج این آزمایش مشخص است، زمانی که بعد بردارهای تعبیه از ۱ بیشتر باشند، عملکرد

[1] Hyper-Parameter

مدل بهبود می‌یابد. پس می‌توانیم از این نتیجه، برداشت کنیم افزایش ابعاد بردارهای تعبیه، ایده‌ی موثری برای بهبود عملکرد مدل است. همچنین قابل ملاحظه است که تعداد لایه‌های شبکه‌های تعامل، رابطه‌ی واضحی با عملکرد مدل در این مجموعه‌ی داده ندارد.

۴-۴-۲ تعداد لایه‌ها و نورون‌های شبکه‌ی سر

شبکه‌ی سر، همان‌طور که در بخش‌های قبل گفته شد، نقش تصمیم‌گیری نهایی مدل را بر عهده دارد. تنظیم دقیق تعداد لایه‌ها و نورون‌های این شبکه، می‌تواند میزان پیچیدگی مدل و توان مدل‌سازی آن را تحت تاثیر قرار دهد؛ پس با طراحی آزمایشی، میزان تاثیر تعداد لایه‌ها و همچنین تعداد نورون‌های هر لایه از این شبکه را بین مقادیر مختلف تغییر داده و عملکرد مدل را روی مجموعه داده‌ی کرایتیو-۲۲ با مساحت تحت منحنی می‌سنجیم. نتایج این آزمایش در شکل ۴-۸ قابل مشاهده است.

همچنین این آزمایش را روی مجموعه داده‌ی آوت‌برین هم تکرار کرده و به دلیل سرعت بالای اجرا بر روی این مجموعه داده، مقادیر متنوع‌تری را از این ابرپارامترها می‌آزماییم. در شکل ۴-۹ مساحت تحت منحنی را برای مدل در این آزمایش گزارش کرده‌ایم.

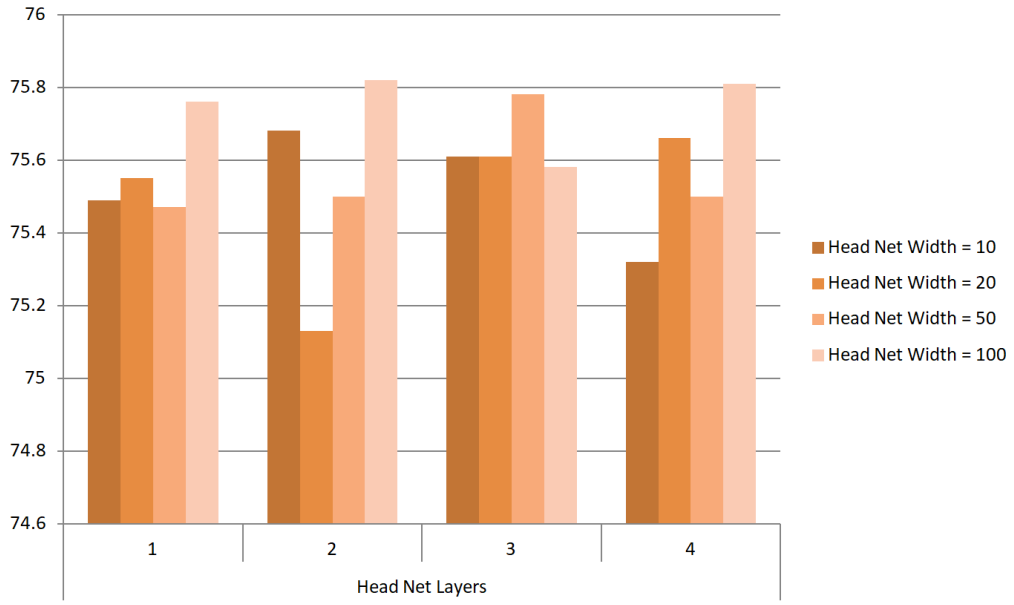
همان‌طور که از نتایج آزمایش‌های فوق مشخص است، تاثیر تعداد لایه‌های شبکه‌ی سر، تنها در یکی از مجموعه‌های داده و آن‌هم به صورت محدود مشاهده می‌شود؛ اما با افزایش تعداد نورون‌های هر لایه از شبکه‌ی سر، عملکرد مدل به صورت مداوم بهبود می‌یابد. می‌توانیم از این نتایج این نکته را برداشت کنیم که به دلیل استخراج ویژگی‌های مرتبه اول (بردارهای تعبیه) و دوم (بردارهای تعامل) مناسب، مدل به عمق زیادی برای پیش‌بینی نرخ کلیک نیاز ندارد؛ اما با افزایش تعداد نورون‌های هر لایه از شبکه‌ی سر، مدل می‌تواند جزئیات بیشتری از این ویژگی‌ها استخراج کرده و مرز تصمیم‌گیری را دقیق‌تر ترسیم کند.

۴-۴-۳ بررسی فضای تعبیه

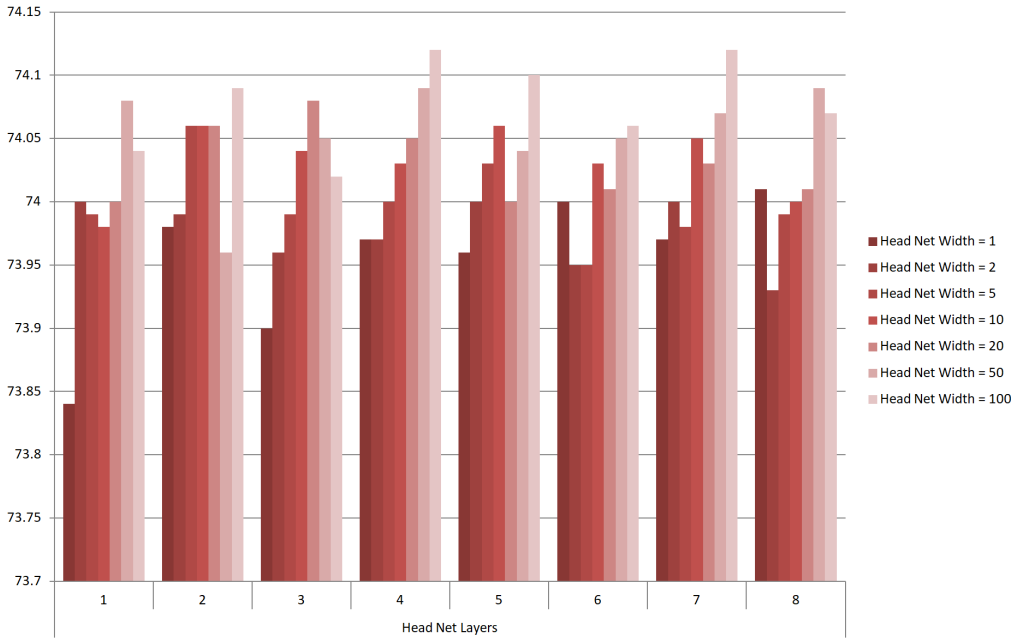
یکی از پرسش‌های مهمی که ممکن است در مورد نتایج این پژوهش به وجود بیاید، تاثیر استفاده از تکنیک‌های مختلف روی کیفیت فضای بردارهای تعبیه است. **آیا متغیر در نظر گرفتن ابعاد بردارهای تعبیه و همچنین تخصیص چندین مسیر مختلف برای انتقال گرادیان به متغیرهای تعبیه‌ی مدل، باعث شکل‌گیری یک فضای تعبیه‌ی مفید می‌شود؟**

برای پاسخ به این پرسش، به تنها فیلد درهم‌سازی نشده‌ی مجموعه داده‌ی آوت‌برین که موقعیت جغرافیایی است، رجوع می‌کنیم. این فیلد نشان دهنده‌ی کشور، استان یا ایالتی است که آدرس آی‌پی کاربر به آن ناحیه تعلق دارد. می‌توانیم فرض کنیم استان‌ها و ایالت‌های مختلف یک کشور، به دلیل شباهت فرهنگی و زبانی، تاثیر مشابهی در برخورد کاربران با تبلیغات آنلاین داشته باشند؛ در نتیجه انتظار داریم استان‌ها یا ایالت‌های مختلف یک کشور، در فضای تعبیه‌ی این فیلد، نزدیک به هم باشند.

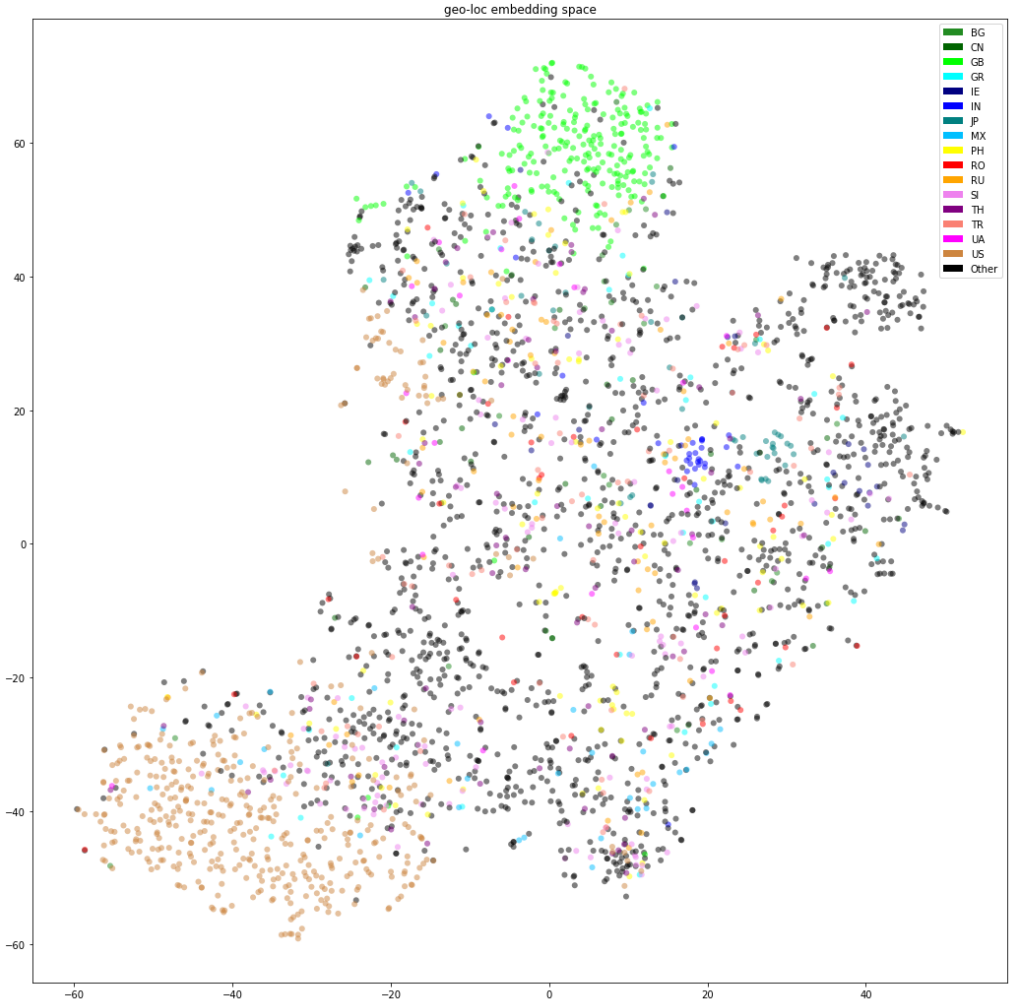
چون فضای تعبیه‌ی این فیلد بیش از دو بعد دارد، نمی‌توانیم بردارهای تعبیه را به صورت خام نمایش دهیم؛



شکل ۴-۸: مساحت تحت نمودار، به ازای تعداد لایه‌های مختلف شبکه‌های تعامل و همچنین مقادیر مختلف بعد بردارهای تعامل روی مجموعه داده‌ی کرایتیو-۲۲



شکل ۴-۹: مساحت تحت نمودار، به ازای تعداد لایه‌های مختلف شبکه‌های تعامل و همچنین مقادیر مختلف بعد بردارهای تعامل روی مجموعه داده‌ی آوت‌برین



شکل ۴-۱۰: نمایی از فضای تعبیه‌ی استخراج شده از فیلد موقعیت جغرافیایی در مجموعه‌ی داده‌ی آوت‌برین توسط روش پیشنهادی

پس از یک روش کاهش ابعاد[۳۸] به نام $T-SNE$ استفاده می‌کنیم و این بردارها را به فضای دو بعدی منتقل می‌کنیم. الگوریتم $T-SNE$ به نحوی کار می‌کند که فاصله‌ی نقاط در فضای خروجی، مانند همین فواصل در فضای ورودی بوده و عملا نقاط نزدیک به هم، پس از کاهش ابعاد باز هم نزدیک به هم قرار گرفته و نقاط دور از هم، پس از کاهش ابعاد همچنان دور از یکدیگر باشند.

در شکل ۴-۱۰ نتایج این آزمایش را مشاهده می‌نمایید. قابل توجه است این شکل پس از کاهش ابعاد این فضا توسط الگوریتم $T-SNE$ به دو بعد رسم شده است. برای سادگی مشاهده‌ی نتایج، استان‌ها و ایالت‌های مختلف هر کشور را به یک رنگ خاص نمایش داده‌ایم. همان طور که انتظار داشتیم، نقاط هم رنگ نزدیک به هم و به صورت خوشه‌های با اندازه‌های متغیر قرار گرفته‌اند. این آزمایش به ما نشان می‌دهد همان‌طور که انتظار داشتیم، اقدامات انجام شده به منظور بهبود کیفیت فضای تعبیه، موثر بوده و مدل پیشنهادی، در ایجاد و استفاده از فضاهای تعبیه‌ی مفید، موفق شده است.

۴-۴-۴ مقایسه با روش‌های پیشین

پس از تنظیم مقادیر ابرپارامترها و اطمینان از عملکرد مدل پیشنهادی، نوبت به مقایسه‌ی آن با برخی از روش‌های پیشین می‌رسد. به دلیل محدودیت‌های سخت‌افزاری، این مقایسه را به ماشین‌های فاکتورگیری ساده و همچنین ماشین‌های فاکتورگیری ژرف محدود می‌کنیم. قابل ذکر است ماشین‌های فاکتورگیری ساده، نماینده‌ی روش‌های غیر ژرف و ماشین‌های فاکتورگیری ژرف، نماینده‌ی روش‌های ژرف در این مقایسه هستند.

مجموعه داده‌ی آوت‌برین

در جدول ۴-۱ نتایج مقایسه‌ی مدل پیشنهادی با روش‌های پیشین را در مجموعه‌ی داده‌ی آوت‌برین مشاهده می‌کنید.

جدول ۴-۱: مقایسه‌ی نهایی عملکرد روی مجموعه‌ی آوت‌برین

نام و جزئیات مدل	مساحت تحت منحنی (درصد)
ماشین فاکتورگیری ساده بعد بردارهای تعبیه = ۹	۷۴٫۲۲
ماشین فاکتورگیری ژرف بعد بردارهای تعبیه = ۱۰ تعداد لایه‌ها = ۳ تعداد نورون‌های هر لایه = ۲۰	۷۲٫۲۷
ماشین فاکتورگیری ژرف بعد بردارهای تعبیه = ۱۰ تعداد لایه‌ها = ۳ تعداد نورون‌های هر لایه = ۱۰۰	۷۳٫۰۰
ماشین فاکتورگیری ژرف بعد بردارهای تعبیه = ۱۰ تعداد لایه‌ها = ۳ تعداد نورون‌های هر لایه = ۴۰۰	۷۳٫۴۴
روش پیشنهادی	۷۴٫۱۳

همان طور که از نتایج قابل مشاهده است، مدل پیشنهادی و ماشین فاکتورگیری ساده، عملکردی مناسب و نزدیک به هم ارائه کرده‌اند. این نکته قابل توجه است که ماشین فاکتورگیری ژرف، در مجموعه داده‌ی آوت‌برین عملکرد مناسبی ندارد. این در حالی است که روش ژرف پیشنهادی، حتی با وجود تعداد بسیار کم ویژگی‌های این مجموعه داده، می‌تواند عملکردی بسیار نزدیک به ماشین فاکتورگیری ساده (مدل غیر ژرف) ارائه کند. این نتیجه نشان می‌دهد روش‌های تنظیم استفاده شده، عملکرد قابل قبولی داشته و جلوی بیش‌برازش مدل پیشنهادی را گرفته‌اند.

مجموعه داده‌ی کرایتیو-۲۲

در جدول ۴-۲ نتایج مقایسه‌ی روش پیشنهادی و ماشین فاکتورگیری ساده را، در مجموعه داده‌ی کرایتیو-۲۲ مشاهده می‌کنید. لازم به ذکر است اجرای مدل ماشین فاکتورگیری ژرف در این مجموعه داده، به دلیل تعداد پارامترهای بسیار بالا قابل انجام نبوده و به ناچار، مقایسه در این مجموعه داده را تنها بین روش پیشنهادی و روش ماشین فاکتورگیری ساده انجام می‌دهیم.

جدول ۴-۲: مقایسه‌ی نهایی عملکرد روی مجموعه‌ی کرایتیو-۲۲

نام و جزئیات مدل	مساحت تحت منحنی (درصد)	دقت (درصد)	بازیابی (درصد)	اف ۱ (درصد)
ماشین فاکتورگیری ساده بعد بردارهای تعبیه = ۵	۷۵٫۴۱	۵۶٫۵۵	۳۴٫۵۸	۴۲٫۹۲
ماشین فاکتورگیری ساده بعد بردارهای تعبیه = ۱۰	۷۴٫۷۵	۵۴٫۸۹	۳۵٫۴۲	۴۳٫۰۶
ماشین فاکتورگیری ساده بعد بردارهای تعبیه = ۴۰	۷۲٫۳۸	۵۰٫۱۲	۳۷٫۲۰	۴۲٫۷۰
ماشین فاکتورگیری ساده بعد بردارهای تعبیه = ۱۰۰	۷۰٫۳۰	۴۶٫۹۲	۳۸٫۳۲	۴۲٫۱۹
روش پیشنهادی	۷۶٫۰۸	۴۳٫۰۷	۷۰٫۳۹	۵۳٫۴۴

نتایج این آزمایش نشان می‌دهد ماشین فاکتورگیری ساده، با افزایش بعد تعبیه، دچار مشکل بیش‌برازش شده و عملکرد آن افت می‌کند. همچنین واضح است که روش پیشنهادی عملکرد بهتری را ارائه می‌کند.

مجموعه داده‌ی کرایتیو-۲۱

در جدول ۴-۳ عملکرد روش پیشنهادی را با روش‌های ماشین فاکتورگیری ساده و ماشین فاکتورگیری ژرف مقایسه می‌کنیم.

جدول ۴-۳: مقایسه‌ی نهایی عملکرد روی مجموعه‌ی کرایتیو-۲۱

نام و جزئیات مدل	مساحت تحت منحنی (درصد)	دقت (درصد)	بازیابی (درصد)	اف ۱ (درصد)
ماشین فاکتورگیری ساده بعد بردارهای تعبیه = ۵	۷۵٫۸۳	۵۸٫۷۷	۳۱٫۷۳	۴۱٫۲۱
ماشین فاکتورگیری ساده بعد بردارهای تعبیه = ۱۰	۷۵٫۴۹	۵۷٫۷۵	۳۲٫۴۹	۴۱٫۵۹
ماشین فاکتورگیری ساده بعد بردارهای تعبیه = ۴۰	۷۳٫۶۸	۵۳٫۶۰	۳۴٫۴۰	۴۱٫۹۱
ماشین فاکتورگیری ساده بعد بردارهای تعبیه = ۱۰۰	۷۱٫۷۱	۵۰٫۱۴	۳۵٫۰۸	۴۱٫۲۸
ماشین فاکتورگیری ژرف بعد بردارهای تعبیه = ۱۰ تعداد لایه‌ها = ۳ تعداد نورون‌های هر لایه = ۲۰	۷۴٫۸۵	۳۲٫۷۱	۹۱٫۸۱	۴۸٫۲۳
ماشین فاکتورگیری ژرف بعد بردارهای تعبیه = ۱۰ تعداد لایه‌ها = ۳ تعداد نورون‌های هر لایه = ۱۰۰	۷۶٫۰۱	۳۸٫۱۶	۸۲٫۵۱	۵۲٫۱۸
ماشین فاکتورگیری ژرف بعد بردارهای تعبیه = ۱۰ تعداد لایه‌ها = ۳ تعداد نورون‌های هر لایه = ۴۰۰	۷۶٫۲۴	۴۲٫۲۱	۷۳٫۳۴	۵۳٫۵۸
روش پیشنهادی	۷۶٫۷۰	۴۳٫۷۰	۶۹٫۹۴	۵۳٫۷۹

همان‌طور که از نتایج مشخص است، روش پیشنهادی در این مجموعه داده، عملکرد بهتری نسبت به ماشین‌های فاکتورگیری ساده و ماشین‌های فاکتورگیری ژرف به نمایش گذاشته است.

مجموعه داده‌ی کرایتیو-۲۰

در جدول ۴-۴ عملکرد نهایی روش پیشنهادی را با روش‌های ماشین فاکتورگیری ساده و ماشین فاکتورگیری ژرف بر روی مجموعه داده‌ی کرایتیو-۲۰ مقایسه می‌کنیم.

همان‌طور که از نتایج قابل مشاهده است، مدل پیشنهادی در این مجموعه داده نیز عملکرد بهتری نشان داده و روش‌های ماشین فاکتورگیری ساده و همچنین ماشین فاکتورگیری ژرف را پشت سر گذاشته است.

جدول ۴-۴: مقایسه‌ی نهایی عملکرد روی مجموعه‌ی کرایتیو-۲۰

نام و جزئیات مدل	مساحت تحت منحنی (درصد)	دقت (درصد)	بازیابی (درصد)	اف ۱ (درصد)
ماشین فاکتورگیری ساده بعد بردارهای تعبیه = ۵	۷۵٫۵۷	۵۹٫۲۰	۳۰٫۳۵	۴۰٫۱۲
ماشین فاکتورگیری ساده بعد بردارهای تعبیه = ۱۰	۷۵٫۳۰	۵۸٫۲۲	۳۱٫۱۳	۴۰٫۵۶
ماشین فاکتورگیری ساده بعد بردارهای تعبیه = ۴۰	۷۳٫۶۲	۵۴٫۲۴	۳۲٫۹۳	۴۰٫۹۸
ماشین فاکتورگیری ساده بعد بردارهای تعبیه = ۱۰۰	۷۱٫۷۵	۵۰٫۶۲	۳۴٫۳۲	۴۰٫۹۰
ماشین فاکتورگیری ژرف بعد بردارهای تعبیه = ۱۰ تعداد لایه‌ها = ۳ تعداد نورون‌های هر لایه = ۲۰	۷۴٫۷۰	۴۲٫۸۵	۶۶٫۴۵	۵۲٫۱۰
ماشین فاکتورگیری ژرف بعد بردارهای تعبیه = ۱۰ تعداد لایه‌ها = ۳ تعداد نورون‌های هر لایه = ۱۰۰	۷۵٫۴۴	۵۵٫۹۴	۳۲٫۰۶	۴۰٫۷۶
ماشین فاکتورگیری ژرف بعد بردارهای تعبیه = ۱۰ تعداد لایه‌ها = ۳ تعداد نورون‌های هر لایه = ۴۰۰	۷۵٫۴۵	۳۳٫۶۴	۹۰٫۶۳	۴۹٫۰۷
روش پیشنهادی	۷۶٫۳۷	۴۲٫۷۶	۶۸٫۶۱	۵۳٫۴۴

فصل ۵

جمع بندی و کارهای آتی

در فصل اول این پایان‌نامه، به معرفی مساله‌ی پیش‌بینی احتمال تعامل کاربران با تبلیغات نمایشی آنلاین و پیش‌نیازهای آن پرداختیم؛سپس چالش‌های موجود پیرامون این مساله را معرفی کردیم. در فصل دوم با بررسی پژوهش‌های پیشین، متوجه شدیم استفاده از بردارهای تعبیه‌ی با ابعاد یکسان، یکی از خصوصیت‌های مشترک همه‌ی این پژوهش‌ها است.

در فصل سوم با وارسی بیشتر این مساله از دو زاویه‌ی مختلف، به این نتیجه‌ی یکسان رسیدیم که این خصوصیت مشترک، می‌تواند یک اشتباه رایج باشد. پس به طراحی یک مدل پیش‌بینی نرخ کلیک پرداختیم که از بردارهای تعبیه با ابعاد متفاوت استفاده کند؛ اما این فرض، باعث ایجاد محدودیت در محاسبه‌ی تعامل در روش پیشنهادی شد. به کمک ایده‌ای از یک پژوهش دیگر، شیوه‌ی محاسبه‌ی تعامل را نیز در مدل پیشنهادی طراحی نمودیم و بقیه‌ی قسمت‌های مدل را بر اساس شرایط مساله طراحی کرده و در فصل چهارم، این مدل را در شرایط گوناگون آزمودیم. نتایج این آزمایش‌ها را مقایسه کرده و نتیجه گرفتیم روش پیشنهادی، از سایر روش‌های موجود در ادبیات پیش‌بینی نرخ کلیک عملکرد بهتری دارد.

۵-۱ کارهای آتی

معرفی یک روش پیشنهادی که عملکرد مناسبی روی مجموعه‌های داده‌ی موجود داشته باشد، تنها آغاز یک مسیر پژوهشی است. برای مفید واقع شدن پژوهش انجام شده، نیاز به برداشتن گام‌های دیگری است که در این بخش به معرفی برخی از این گام‌ها می‌پردازیم.

۵-۱-۱ ارائه‌ی پیاده‌سازی کارا

همانطور که در فصل اول بررسی شد، سرعت اجرای فرآیند مزایده‌ی بلادرنگ بسیار بالا است؛ پس مدل‌های پیش‌بینی نرخ کلیک، باید در زمان بسیار کوتاهی، نرخ کلیک کاربر بر تعداد بسیار زیادی از بنرهای تبلیغاتی را تخمین بزنند. این امر باعث می‌شود ارائه‌ی یک پیاده‌سازی سریع و کارا، یکی از مهم‌ترین گام‌های لازم برای ادامه‌ی این پژوهش به شمار رود.

۵-۱-۲ طراحی مدل برای استفاده در شرایط آنلاین

شرایط آنلاین به شرایطی گفته می‌شود که در آن لیست موجودیت‌های هر فیلد، هر لحظه قابل رشد باشد. یعنی هر لحظه ممکن باشد یک کاربر جدید وارد چرخه شده یا یک بنر تبلیغاتی جدید ایجاد شود. این تنها شرایطی است که می‌توان میزان مقاومت یک مدل پیش‌بینی نرخ کلیک را در برابر چالش شروع سرد اندازه‌گیری نمود؛ اما برای آزمودن روش پیشنهادی در چنین شرایطی، باید تغییراتی در ساختار آن لحاظ شود. به عنوان مثال، در شرایط آفلاین، تعداد سطرهای ماتریس‌های تعبیه همیشه ثابت است؛ اما در صورت آنلاین بودن شرایط، ابعاد این ماتریس‌ها هر لحظه می‌توانند رشد کنند. چگونگی مقداردهی اولیه‌ی سطرهای جدید این ماتریس‌ها یکی از پرسش‌هایی است که برای ادامه‌ی مسیر این پژوهش، باید پاسخ داده شوند.

۵-۱-۳ یافتن راهی برای ایجاد تعادل بین اکتشاف و بهره‌برداری

در بسیاری از مسائل دنیای واقعی، چالش موازنه‌ی بین اکتشاف و بهره‌برداری خودنمایی می‌کند. به عنوان مثال، یک مدل پیش‌بینی نرخ کلیک که در شرایط آنلاین کار می‌کند، هر بار باید تصمیم بگیرد که **آیا بنر تبلیغاتی دارای بیشترین احتمال کلیک را به کاربر نمایش دهد، یا بنر جدیدی که هنوز اطلاعات خاصی در مورد رفتار کاربران با آن وجود ندارد؟** یافتن راهی برای برقراری این موازنه، یک گام دیگر در ادامه‌ی راه این پژوهش خواهد بود.

مراجع

[1] H. Choi, C. F. Mela, S. R. Balseiro, and A. Leary, "Online display advertising markets: A literature review and future directions." *Inf. Syst. Res.*, vol. 31, no. 2, pp. 556–575, 2020. [Online]. Available: http://dblp.uni-trier.de/db/journals/isr/isr31.html#ChoiMBL20

[2] Y. Yuan, F. Wang, J. Li, and R. Qin, "A survey on real time bidding advertising," in *Proceedings of 2014 IEEE International Conference on Service Operations and Logistics, and Informatics*, 2014, pp. 418–423.

[3] R. Qin, X. Ni, Y. Yuan, J. Li, and F.-Y. Wang, "Revenue models for demand side platforms in real time bidding advertising." in *SMC*. IEEE, 2017, pp. 438–443. [Online]. Available: http://dblp.uni-trier.de/db/conf/smc/smc2017.html#QinNYLW17

[4] C. X. Ling and V. S. Sheng, "Class imbalance problem." in *Encyclopedia of Machine Learning and Data Mining*, C. Sammut and G. I. Webb, Eds. Springer, 2017, pp. 204–205. [Online]. Available: http://dblp.uni-trier.de/db/reference/ml/ml2017.html#LingS17

[5] A. M. Pires and J. A. Branco, "High dimensionality: The latest challenge to data analysis," 2019.

[6] B. Lika, K. Kolomvatsos, and S. Hadjiefthymiades, "Facing the cold start problem in recommender systems." *Expert Syst. Appl.*, vol. 41, no. 4, pp. 2065–2073, 2014. [Online]. Available: http://dblp.uni-trier.de/db/journals/eswa/eswa41.html#LikaKH14

[7] Z. Zhang, C. Liu, Y. Zhang, and T. Zhou, "Solving the cold-start problem in recommender systems with social tags," *CoRR*, vol. abs/1004.3732, 2010. [Online]. Available: http://arxiv.org/abs/1004.3732

[8] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the 5th Annual Workshop on Computational Learning Theory (COLT'92)*, D. Haussler, Ed. Pittsburgh, PA, USA: ACM Press, July 1992, pp. 144–152. [Online]. Available: http://doi.acm.org/10.1145/130385.130401

[9] K. Gai, X. Zhu, H. Li, K. Liu, and Z. Wang, "Learning piece-wise linear models from large scale data for ad click prediction." *CoRR*, vol. abs/1704.05194, 2017. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1704.html#GaiZLLW17

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[11] Y. Xiao, Z. Wei, and Z. Wang, "A limited memory bfgs-type method for large-scale unconstrained optimization." *Comput. Math. Appl.*, vol. 56, no. 4, pp. 1001–1009, 2008. [Online]. Available: http://dblp.uni-trier.de/db/journals/cma/cma56.html#XiaoWW08

[12] T. Graepel, J. Q. Candela, T. Borchert, and R. Herbrich, "Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine." in *ICML*, J. Fürnkranz and T. Joachims, Eds. Omnipress, 2010, pp. 13–20. [Online]. Available: http://dblp.uni-trier.de/db/conf/icml/icml2010.html#GraepelCBH10

[13] S. Rendle, "Factorization Machines," in *Proceedings of the 2010 IEEE International Conference on Data Mining*, ser. ICDM '10. IEEE, Dec. 2010, pp. 995–1000. [Online]. Available: http://ieeexplore.ieee.org/document/5694074/

[14] Y.-C. Juan, Y. Zhuang, W.-S. Chin, and C.-J. Lin, "Field-aware factorization machines for ctr prediction." in *RecSys*, S. Sen, W. Geyer, J. Freyne, and P. Castells, Eds. ACM, 2016, pp. 43–50. [Online]. Available: http://dblp.uni-trier.de/db/conf/recsys/recsys2016.html#JuanZCL16

[15] Y. Juan, D. Lefortier, and O. Chapelle, "Field-aware factorization machines in a real-world online advertising system." *CoRR*, vol. abs/1701.04099, 2017. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1701.html#JuanLC17

[16] J. Pan, J. Xu, A. L. Ruiz, W. Zhao, S. Pan, Y. Sun, and Q. Lu, "Field-weighted factorization machines for click-through rate prediction in display advertising." *CoRR*, vol. abs/1806.03514, 2018. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1806.html#abs-1806-03514

[17] S.-T. L. Freudenthaler, C. and S. Rendle, "Bayesian factorization machines," in *In Proceedings of the NIPS Workshop on Sparse Representation and Low-rank Approximation*, 2011.

[18] Z. Pan, E. Chen, Q. Liu, T. Xu, H. Ma, and H. Lin, "Sparse factorization machines for click-through rate prediction." in *ICDM*, F. Bonchi, J. Domingo-Ferrer, R. Baeza-Yates, Z.-H. Zhou, and X. Wu, Eds. IEEE Computer Society, 2016, pp. 400–409. [Online]. Available: http://dblp.uni-trier.de/db/conf/icdm/icdm2016.html#PanCLXML16

[19] J. Xiao, H. Ye, X. He, H. Zhang, F. Wu, and T.-S. Chua, "Attentional factorization machines: Learning the weight of feature interactions via attention networks." *CoRR*, vol. abs/1708.04617, 2017. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1708.html#abs-1708-04617

[20] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014. [Online]. Available: http://portal.acm.org/citation.cfm?id=2670313

[21] A. N. Tikhonov, "On the stability of inverse problems," in *Dokl. Akad. Nauk SSSR*, vol. 39, 1943, pp. 195–198.

[22] S. Zhang, L. Yao, and A. Sun, "Deep learning based recommender system: A survey and new perspectives." *CoRR*, vol. abs/1707.07435, 2017. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1707.html#ZhangYS17aa

[23] J. Chen, B. Sun, H. Li, H. Lu, and X.-S. Hua, "Deep ctr prediction in display advertising." in *ACM Multimedia*, A. Hanjalic, C. Snoek, M. Worring, D. C. A. Bulterman, B. Huet, A. Kelliher, Y. Kompatsiaris, and J. Li, Eds. ACM, 2016, pp. 811–820. [Online]. Available: http://dblp.uni-trier.de/db/conf/mm/mm2016.html#ChenSLLH16

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015, cite arxiv:1512.03385Comment: Tech report. [Online]. Available: http://arxiv.org/abs/1512.03385

[25] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines." in *ICML*, J. Fürnkranz and T. Joachims, Eds. Omnipress, 2010, pp. 807–814. [Online]. Available: http://dblp.uni-trier.de/db/conf/icml/icml2010.html#NairH10

[26] C. Guo and F. Berkhahn, "Entity embeddings of categorical variables." *CoRR*, vol. abs/1604.06737, 2016. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1604.html#GuoB16

[27] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, cite arxiv:1502.03167. [Online]. Available: http://arxiv.org/abs/1502.03167

[28] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "Deepfm: A factorization-machine based neural network for ctr prediction." in *IJCAI*, C. Sierra, Ed. ijcai.org, 2017, pp. 1725–1731. [Online]. Available: http://dblp.uni-trier.de/db/conf/ijcai/ijcai2017.html#GuoTYLH17

[29] H. Guo, R. Tang, Y. Ye, Z. Li, X. He, and Z. Dong, "Deepfm: An end-to-end wide and deep learning framework for ctr prediction." *CoRR*, vol. abs/1804.04950, 2018. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1804.html#abs-1804-04950

[30] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, and H. Shah, "Wide and deep learning for recommender systems." in *DLRS@RecSys*, A. Karatzoglou, B. Hidasi, D. Tikk, O. S. Shalom, H. Roitman, B. Shapira, and L. Rokach, Eds. ACM, 2016, pp. 7–10. [Online]. Available: http://dblp.uni-trier.de/db/conf/recsys/dlrs2016.html#Cheng0HSCAACCIA16

[31] Q. Wang, F. Liu, S. Xing, and X. Zhao, "A new approach for advertising ctr prediction based on deep neural network via attention mechanism." *Comput. Math. Methods Medicine*, vol. 2018, pp. 8 056 541:1–8 056 541:11, 2018. [Online]. Available: http://dblp.uni-trier.de/db/journals/cmmm/cmmm2018.html#WangLXZ18

[32] D. H. Ballard, "Modular learning in neural networks." in *AAAI*, K. D. Forbus and H. E. Shrobe, Eds. Morgan Kaufmann, 1987, pp. 279–284. [Online]. Available: http://dblp.uni-trier.de/db/conf/aaai/aaai87.html#Ballard87

[33] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*. Urbana and Chicago: University of Illinois Press, 1949.

[34] M. Naumov, "On the dimensionality of embeddings for sparse features and data." *CoRR*, vol. abs/1901.02103, 2019. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1901.html#abs-1901-02103

[35] A. Ginart, M. Naumov, D. Mudigere, J. Yang, and J. Zou, "Mixed dimension embeddings with application to memory-efficient recommendation systems." *CoRR*, vol. abs/1909.11810, 2019. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1909.html#abs-1909-11810

[36] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th International Conference on World Wide Web*, ser. WWW '17. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2017, p. 173–182. [Online]. Available: https://doi.org/10.1145/3038912.3052569

[37] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, no. 1. Citeseer, 2013, p. 3.

[38] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008. [Online]. Available: http://www.jmlr.org/papers/v9/vandermaaten08a.html

واژه‌نامه فارسی به انگلیسی

آدام . . . . ADAptive Moment estimation
آدرس آی پی . . . . . IP Address
آنتروپی . . . . . Entropy
آنتروپی متقابل . . . . . Cross Entropy
ابر پیشین . . . . . Hyperprior
ابر پیشین‌های سلسله مراتبی . . . . Hierarchical hyperpriors
ابرپارامتر . . . . . Hyper-Parameter
اطلاعات مشترک . . . . Mutual information
اچ‌تی‌ام‌ال . . . . . HTML
اچ‌تی‌تی‌پی . . . . . HTTP
اکتشاف . . . . . Expolration
اکتشاف یا استفاده . . . . . Explore / Exploit
بازار اپلیکیشن گوگل پلی . . . . . Google Play Application Store
بازاریاب . . . . . Marketer
بردار تک داغ . . . . . One hot vector
بردار چند داغ . . . . . Multi hot vector
برنامه‌ریزی درجه دوم . . . . . Quadratic Programming
بنر . . . . . Banner
بهره برداری . . . . . Exploitation
بهینه‌سازی موتورهای جستجو . . . . . SEO
بیشینه‌سازی امید ریاضی . . . . . Expectation Maximization
بینگ . . . . . Bing
تبلیغ کننده . . . . . Advertisier
تبلیغات نمایشی . . . . . Display Advertising
تبلیغات کلمه کلیدی . Keyword Advertising
تخمین زننده‌ی سراسری . . . . . Global Approximator
تعامل . . . . . Interaction
تعامل‌های ضمنی . . . Implicit Interactions
تعبیه . . . . . Embedding
تنظیم مرتبه دوم L2-Regularization
تنک . . . . . Sparse
تنگنا . . . . . Bottleneck
توجه . . . . . Attention
جدول تعبیه . . . . . Embedding Table
جستجوی توری . . . . . Grid search
جستجوی حمایت شده . . Sponsored search
حذف تصادفی . . . . . dropout
خطای لگاریتمی . . . . . Log Loss
خودکارسازی . . . . . Automation
خودکدگذار . . . . . Auto Encoder
خودکدگذار پشته شده Stacked Auto Encoder
درهمسازی ویژگی‌ها . . . . . Feature Hashing
درهم‌سازی شده . . . . . Hashed
دسته آموزش . . . . . Batch
دسته بندی . . . . . Classification
دسته‌ای . . . . . Categorical
دسته‌ی اقلیت . . . . . Minority Class
دلخواه . . . . . Arbitrary
راک . . . . . ROC
رز نت . . . . . ResNet
رگرسیون پرابیت . . . . . Probit Regression

زنجیره‌ی مارکوف مونت کارلو . Markov Chain Monte Carlo
سافت مکس . . . . . Softmax
سر تا سر . . . . . End To End
سکوی سمت تامین . Supply Side Platform
سکوی سمت نیاز . Demand Side Platform
سیستم‌های پیشنهاد دهنده . . . Recommender systems
سیگموید . . . . . Sigmoid
شبکه عصبی تماما متصل . . Fully Connected Neural Network
شبکه‌های اجتماعی . . . . . Social Networks
شبکه‌ی سر . . . . . Head Network
شبکه‌ی عصبی چند لایه . . . . . Multi Layer Perceptron
شناسه . . . . . ID
صریح . . . . . Explicit
ضمنی . . . . . Implicit
علی‌بابا . . . . . Alibaba
فرم دوگان . . . . . Dual Form
فضای نهان . . . . . Latent Space
فیلتر کردن مشترک . Collaborative Filtering
فیلد . . . . . Field
قرارداد مستقیم . . . . . Direct Deas
قراردادهای تضمین شده‌ی اتوماتیک Automated Guaranteed Deals
قراردادهای تضمین شده‌ی برنامه‌ریزی شده . . . . . Programmatic Guaranteed Deals
قرینه‌ی درستنمایی . . . . Negative likelihood
قرینه‌ی لگاریتم درستنمایی . . . . Negative Log Likelihood
ماشین‌های فاکتورگیری . . . . . Factorization Machines
محدب . . . . . Convex
مخلوط مقیاس‌شده . . . . . Scale mixture
مزایده . . . . . Auction
مزایده‌های بلادرنگ آزاد . . . . Open Realtime Auction
مزایده‌های بلادرنگ خصوصی . . . . . Private Realtime Auction
مزایده‌ی بلادرنگ Realtime Bidding (RTB)
مساحت تحت منحنی . . Area Under Curve
معاملات برنامه‌ریزی شده . . . Programmatic Deals
معاملات تضمین شده . . . Guaranteed Deals
مقادیر گم شده . . . . . Missing values
موقعیت قابل تبلیغ . . . . . Impression
ناشر . . . . . Publisher
ناشناس شده . . . . . Anonimized
ناهموار . . . . . Non-smooth
نرخ تبدیل . . . . . Convertion Rate
نرخ مثبت درست . . . . . TPR
نرخ مثبت غلط . . . . . FPR
نرخ کلیک . . . . . Click Through Rate
نرم . . . . . Smooth
نرمال‌سازی دسته‌ای . . Batch Normalization
نظریه‌ی اطلاعات . . . . Information Theory
نفرین ابعاد . . . . . Curse of dimentionality
نمونه برداری گیبس . . . . . Gibbs Sampling
نهفته . . . . . Latent
نگاشت . . . . . Mapping
واحد خطی یکسو کننده (رلو) . . . . . ReLU
واحد خطی یکسو کننده‌ی نشت کننده . . . . . LeakyReLU
ویژگی باینری . . . . . Binary feature
ویژگی‌های نهان . . . . . Latent Features
پرابیت . . . . . Probit
پیام‌رسانی . . . . . Message passing
پیش‌آموزش . . . . . Pretrain
یادگیری ژرف . . . . . Deep Learning
چالش ابعاد بالا . . . . . High dimentionality challenge

چالش شروع سرد . . . . Cold start challenge
چالش عدم توازن شدید کلاس‌ها . . High class imbalance challenge
چگال . . . . . Dense
کاربر . . . . . User
کدگذاری . . . . . Encoding
کدگذاری یک از k . . . . . One of k coding
کدگشایی . . . . . Decoding
کرایتیو . . . . . Criteo
کواسی-نیوتون با حافظه‌ی محدود . . . . LBFGS
کگل . . . . . Kaggle
گرادیان کاهشی تصادفی Stochastic Gradient Descent
گرادیان کاهشی دسته‌ای Mini-Batch Gradient Descent
گوگل . . . . . Google

واژه‌نامه انگلیسی به فارسی

آدام ..... ADAptive Moment estimation
تبلیغ کننده ..... Advertisier
علی‌بابا ..... Alibaba
ناشناس شده ..... Anonimized
دلخواه ..... Arbitrary
مساحت تحت منحنی ... Area Under Curve
توجه ..... Attention
مزایده ..... Auction
خودکدگذار ..... Auto Encoder
قراردادهای تضمین شده‌ی اتوماتیک Automated Guaranteed Deals
خودکارسازی ..... Automation
بنر ..... Banner
دسته آموزش ..... Batch
نرمال‌سازی دسته‌ای .. Batch Normalization
ویژگی باینری ..... Binary feature
بینگ ..... Bing
تنگنا ..... Bottleneck
دسته‌ای ..... Categorical
دسته بندی ..... Classification
نرخ کلیک ..... Click Through Rate
چالش شروع سرد ..... Cold start challenge
فیلتر کردن مشترک . Collaborative Filtering
نرخ تبدیل ..... Convertion Rate
محدب ..... Convex
کرایتیو ..... Criteo
آنتروپی متقابل ..... Cross Entropy
نفرین ابعاد ..... Curse of dimentionality

کدگشایی ..... Decoding
یادگیری ژرف ..... Deep Learning
سکوی سمت نیاز .. Demand Side Platform
چگال ..... Dense
قرارداد مستقیم ..... Direct Deas
تبلیغات نمایشی ..... Display Advertising
حذف تصادفی ..... dropout
فرم دوگان ..... Dual Form
تعبیه ..... Embedding
جدول تعبیه ..... Embedding Table
کدگذاری ..... Encoding
سر تا سر ..... End To End
آنتروپی ..... Entropy
بیشینه‌سازی امید ریاضی .. Expectation Maximization
صریح ..... Explicit
بهره برداری ..... Exploitation
اکتشاف یا استفاده ..... Explore / Exploit
اکتشاف ..... Expolration
ماشین‌های فاکتورگیری ..... Factorization Machines
درهمسازی ویژگی‌ها ..... Feature Hashing
فیلد ..... Field
نرخ مثبت غلط ..... FPR
شبکه عصبی تماما متصل . Fully Connected Neural Network
نمونه برداری گیبس ..... Gibbs Sampling
تخمین زننده‌ی ..... Global Approximator

سراسری
گوگل ..... Google
بازار ..... Google Play Application Store
اپلیکیشن گوگل پلی
جستجوی توری ..... Grid search
معاملات تضمین شده ... Guaranteed Deals
درهم‌سازی شده ..... Hashed
شبکه‌ی سر ..... Head Network
ابر پیشین‌های .... Hierarchical hyperpriors
سلسله مراتبی
چالش ... High class imbalance challenge
عدم توازن شدید کلاس‌ها
چالش ابعاد High dimentionality challenge
بالا
اچ‌تی‌ام‌ال ..... HTML
اچ‌تی‌تی‌پی ..... HTTP
ابرپارامتر ..... Hyper-Parameter
ابر پیشین ..... Hyperprior
شناسه ..... ID
ضمنی ..... Implicit
تعامل‌های ضمنی .... Implicit Interactions
موقعیت قابل تبلیغ ..... Impression
نظریه‌ی اطلاعات ..... Information Theory
تعامل ..... Interaction
آدرس آی پی ..... IP Address
کگل ..... Kaggle
تبلیغات کلمه کلیدی . Keyword Advertising
تنظیم مرتبه دوم L2-Regularization
نهفته ..... Latent
ویژگی‌های نهان ..... Latent Features
فضای نهان ..... Latent Space
کواسی-نیوتون با حافظه‌ی محدود .... LBFGS
واحد خطی یکسو کننده‌ی نشت .. LeakyReLU
کننده
خطای لگاریتمی ..... Log Loss
نگاشت ..... Mapping
بازاریاب ..... Marketer
زنجیره‌ی .... Markov Chain Monte Carlo
مارکوف مونت کارلو
پیام‌رسانی ..... Message passing
گرادیان ... Mini-Batch Gradient Descent
کاهشی دسته‌ای
دسته‌ی اقلیت ..... Minority Class
مقادیر گم شده ..... Missing values
بردار چند داغ ..... Multi hot vector
شبکه‌ی عصبی چند Multi Layer Perceptron
لایه
اطلاعات مشترک ..... Mutual information
قرینه‌ی درستنمایی ..... Negative likelihood
قرینه‌ی لگاریتم .. Negative Log Likelihood
درستنمایی
ناهموار ..... Non-smooth
بردار تک داغ ..... One hot vector
کدگذاری یک از k ..... One of k coding
مزایده‌های ..... Open Realtime Auction
بلادرنگ آزاد
پیش‌آموزش ..... Pretrain
مزایده‌های ..... Private Realtime Auction
بلادرنگ خصوصی
پرابیت ..... Probit
رگرسیون پرابیت ..... Probit Regression
معاملات برنامه‌ریزی .. Programmatic Deals
شده
..... Programmatic Guaranteed Deals
قراردادهای تضمین شده‌ی برنامه‌ریزی شده
ناشر ..... Publisher
برنامه‌ریزی درجه . Quadratic Programming
دوم
مزایده‌ی بلادرنگ Realtime Bidding (RTB)
سیستم‌های پیشنهاد . Recommender systems
دهنده
واحد خطی یکسو کننده (رلو) ..... ReLU

رز نت ..... ResNet
راک ..... ROC
مخلوط مقیاس‌شده ..... Scale mixture
بهینه‌سازی موتورهای جستجو ..... SEO
سیگموید ..... Sigmoid
نرم ..... Smooth
شبکه‌های اجتماعی ..... Social Networks
سافت مکس ..... Softmax
تنک ..... Sparse
جستجوی حمایت شده ... Sponsored search
خودکدگذار پشته شده ... Stacked Auto Encoder
گرادیان کاهشی تصادفی .... Stochastic Gradient Descent
سکوی سمت تامین .. Supply Side Platform
نرخ مثبت درست ..... TPR
کاربر ..... User

Sharif University of Technology
Department of Computer Engineering

M.Sc. Thesis
Artificial Intelligence

Topic

User Convertion Prediction In Display Advertisement

By
Mohammadreza Rezaei

Supervisor
Hamid R. Rabiee

Winter 2021