دانشگاه صنعتی شریف
دانشکده‌ی مهندسی کامپیوتر

پایان‌نامه‌ی کارشناسی
مهندسی کامپیوتر

عنوان:

بررسی استفاده از داده‌های یکنواخت برای حل مسئله‌ی پیش‌قدر در سامانه‌های توصیه‌گر

نگارش:

امیرعباس عسگری

استادان راهنما:

دکتر سید عباس حسینی ـ دکتر حمیدرضا ربیعی

تیر ۱۴۰۱


بسم الله الرحمن الرحیم

به نام خدا

دانشگاه صنعتی شریف

دانشکده‌ی مهندسی کامپیوتر

پایان‌نامه‌ی کارشناسی

عنوان: بررسی استفاده از داده‌های یکنواخت برای حل مسئله‌ی پیش‌قدر در سامانه‌های توصیه‌گر

نگارش: امیرعباس عسگری

کمیته‌ی ممتحنین

استادان راهنما: دکتر سید عباس امضاء:
حسینی ـ دکتر حمیدرضا ربیعی

امضاء:

استاد مدعو: دکتر محمدحسین رهبان امضاء:

تاریخ:

سپاس

از استادان بزرگوارم، جناب آقایان دکتر سید عباس حسینی و دکتر حمیدرضا ربیعی، که با راهنمایی‌های خود بنده را در انجام این پروژه یاری نموده‌اند کمال تشکر و قدردانی را دارم. همچنین از جناب آقای مهندس سید محمدجواد فیض‌آبادی ثانی که از آغاز تعریف پروژه و در همه‌ی مراحل اجرای آن با پیگیری مستمر خود بنده را در پیشبرد هر چه بهتر پروژه و رفع چالش‌ها و مسائل آن همراهی کرده است، صمیمانه سپاسگزاری می‌کنم. در پایان نیز از خانواده‌ی عزیزم تشکر می‌کنم که در تمام مراحل زندگی جدی‌ترین پشتیبان من بوده‌اند و همواره خود را مرهون زحمات بی‌دریغ آنها می‌دانم.

چکیده

با گسترش روزافزون کسب‌وکارها و فروشگاه‌های اینترنتی در سال‌های اخیر، حجم عظیمی از داده‌های مربوط به تراکنش‌ها و سابقه‌ی فعالیت‌های کاربران در این بسترهای فروش ثبت شده است. تمایل به استفاده‌ی هرچه بهینه‌تر از این حجم اطلاعات ارزشمند، نیاز به حضور سامانه‌های توصیه‌گر را به عنوان یکی از عناصر کلیدی این‌گونه کسب‌وکارها بیش از پیش برجسته کرده است. هدف اصلی سامانه‌های توصیه‌گر هوشمندسازی فرآیند تبلیغات است؛ به گونه‌ای که محصولات یا خدمات پیشنهادی به هر یک از کاربران هدف سامانه تا حد امکان شخصی‌سازی شود و به این ترتیب احتمال واکنش مثبت هر کاربر به موارد پیشنهادشده به او بیشینه گردد. یکی از موانع مهم درمسیر عملکرد مطلوب سامانه‌های توصیه‌گر، مسئله‌ی وجود پیش‌قدر در داده‌های جمع‌آوری‌شده برای آموزش این‌گونه سامانه‌هاست که معمولا از آن با عنوان پیش‌قدر در مدل پیشین یاد می‌شود. یکی از راهکارهای مطرح‌شده برای رویارویی با این مسئله، بهره‌گیری از داده‌هایی است که به صورت کاملا تصادفی و بدون توجه به علایق پیشین کاربران به آنها پیشنهاد شده‌اند. البته واضح است که اگر همه‌ی داده‌های آموزشی سامانه از پیشنهادهای کاملا تصادفی به کاربران جمع‌آوری شده باشند، مسئله‌ی پیش‌قدر مذکور منتفی است اما این کار در عمل ممکن نیست؛ چرا که به سطح رضایت عمومی کاربران از سامانه بسیار لطمه می‌زند. لذا در این رویکرد فقط بخش کوچکی از جمع‌آوری داده‌ها به این شکل اصطلاحاً «یکنواخت» انجام می‌گیرد. در این پایان‌نامه ابتدا به طور مختصر انواع مختلف مسئله‌ی پیش‌قدر در سامانه‌های توصیه‌گر و برخی راهکارهای آنها در ادبیات مسئله را ذکر می‌کنیم. سپس به شرح جزئیات دو راهبرد مبتنی بر برچسب به نام‌های «پل» و «تصفیه» می‌پردازیم که از داده‌های یکنواخت به دو شکل مختلف در فرآیند آموزش مدل استفاده می‌کنند. همچنین پیاده‌سازی جدید خودمان با استفاده از شبکه‌های عصبی روبه‌جلو برای این دو روش، و نیز نتایج آزمایش‌های انجام‌شده با آن را ارائه می‌کنیم.

کلیدواژه‌ها: سامانه‌های توصیه‌گر، مسئله‌ی پیش‌قدر در داده‌ها، استفاده از داده‌های یکنواخت

فهرست مطالب

فهرست شکل‌ها

فهرست جدول‌ها

فصل ۱

مقدمه

۱ـ۱ تعریف مسئله

هسته‌ی اصلی بسیاری از **سامانه‌های توصیه‌گر**[۱] کنونی روش‌های یادگیری بانظارت[۲] است که در آنها از الگوریتم‌های یادگیری ماشین[۳] برای ساختن مدل‌هایی استفاده می‌شود که پس از آموزش دیدن روی حجم مناسبی از داده‌های مربوط به بازخورد افراد مختلف نسبت به محصولات پیشنهادی به آنها، بتوانند رفتار افراد را در صورت رویارویی با پیشنهادهای جدید پیش‌بینی کنند.

یکی از مشکلات مهم این سامانه‌ها مسئله‌ی پیش‌قدر در داده‌ها[۴]ی مورد استفاده برای آموزش آنهاست. این مشکل از آنجا ناشی می‌شود که ممکن است سامانه‌ی توصیه‌گر برخی محصولات را به دلایلی از قبیل محبوبیت قبلی، بالا بودن نرخ پیشنهادی[۵] تبلیغ‌کننده برای آنها، و ... بیشتر به مخاطبین رسانه نمایش دهد و لذا داده‌های بیشتری در مورد بازخورد افراد به این تبلیغ‌ها وجود خواهد داشت. در این شرایط برخی محصولات دیگر به دلایلی برعکس آنچه گفته شد، فرصت کمتری برای نمایش به مشتریان بالقوه می‌یابند. لذا داده‌های جمع‌آوری‌شده برای آنها محدودتر خواهد بود و این فرصت محدودتر غالباً باعث می‌شود که درصد بازخوردهای مثبت این محصولات کمتر از سایرین باشد و در نگاه اول ممکن است نتیجه بگیریم که این محصولات محبوبیت عمومی کمتری در بین کاربران هدف داشته‌اند. اما این

[۱] Recommendation Systems
[۲] Supervised Learning
[۳] Machine Learning
[۴] Biased Data Problem
[۵] Bid Rate

نتیجه‌گیری لزوما درست نیست؛ چرا که اساساً فرصتی برابر برای عرضه‌شدن به محصولات مختلف داده نشده است و لذا نسبت بازخوردهای مثبت و منفی ثبت‌شده برای هر محصول در دادگان تحت تأثیر شیوه‌ی پیشنهاددهی نامتوازن سامانه و پیش‌قدر آن قرار گرفته است.

یک راهبرد برای حل این مسئله استفاده از داده‌های یکنواخت[۶] در کنار داده‌های دارای پیش‌قدر است. منظور از داده‌های یکنواخت مجموعه‌ی داده‌هایی است که از طریق پیشنهاد دادن تصادفی تعدادی محصول به کاربران و بدون توجه به سابقه‌ی علایق و فعالیت‌های هر یک در دادگان موجود جمع‌آوری می‌شوند. اگرچه وقتی همه‌ی پیشنهادهای یک سامانه‌ی توصیه‌گر به کاربران به طور کاملاً یکنواخت انجام گیرد، می‌توان گفت که مسئله‌ی پیش‌قدر در داده‌ها به کلی از بین می‌رود اما طبعا چنین رویکردی ما را از هدف اصلی سامانه‌های توصیه‌گر که تبلیغات هدفمند و حتی‌الامکان شخصی‌سازی شده است دور خواهد کرد و رضایت کاربران نیز از پینشهادهای کاملاً تصادفی چنین سامانه‌ای بسیار پایین خواهد بود.

به دست آوردن داده‌های یکنواخت در سامانه‌های توصیه‌گر واقعی همواره سخت و پرهزینه است؛ چرا که جمع‌آوری آنها مستلزم دخالت در عملکرد معمول سامانه‌ی توصیه‌گر است تا به جای مدل پیشنهاددهی خود، از یک سیاست پیشنهاددهی با توزیع یکنواخت برای هر یک از افراد استفاده کند. در این صورت داده‌های جمع‌آوری‌شده می‌توانند به عنوان منبع یک منبع خوب بدون پیش‌قدر تلقی شوند؛ چرا که این نتایج تحت تأثیر عملکرد یک سامانه‌ی توصیه‌گر قبلاً توسعه‌یافته قرار نداشته‌اند [۱].

لذا یک راهکار میانی، تخصیص بخش کمی (برای مثال ٪۱) از کل داده‌های آموزشی[۷] سامانه به داده‌های جمع‌آوری‌شده به صورت یکنواخت است و باقی داده‌های آموزشی را همان داده‌های دارای پیش‌قدر تشکیل خواهند داد. این ترکیب با هدف اصلاح پیش‌قدر موجود در بخش عمده‌ی داده‌ها با استفاده از حجم بسیار کمتری داده‌ی بکنواخت است تا مدل‌های یاد گرفته شده روی داده‌های با پیش‌قدر تا حد امکان به مدل‌های آموزش دیده روی داده‌های یکنواخت (به نوعی ایده‌آل در این مسئله) نزدیک شوند و در نهایت نتایج بهتری روی داده‌های آزمایشی[۸] ــ که خود بخشی از کل داده‌های یکنواخت موجودند ــ بگیریم. در واقع شرکت دادن داده‌های با پیش‌قدر به عنوان بخشی از داده‌های آزمایشی برای ارزیابی نهایی، معقول نیست؛ چرا که ما می‌خواهیم عملکرد مدلمان را در شرایطی واقعی بسنجیم که در آن به هر محصول، فارغ از حجم بازخوردهای موجود از آن در دادگان قبلی ما، شانس برابری برای

[۶] Uniform Data
[۷] Train Data
[۸] Test Data

عرضه به کاربران مختلف داده شود تا بازخوردهای جمع‌آوری‌شده از کاربران قابل اعتمادتر و به واقعیت نزدیک‌تر باشد.

۱ـ۲ اهمیت موضوع

با گسترش روزافزون شبکه‌ی اینترنت و فضای مجازی، کسب‌وکارهای جدید بسیاری در این بستر شکل گرفته‌اند که به دنبال توسعه‌ی ابعاد و افزایش درآمد خود از این طریق هستند. با توجه به تعداد بسیار بالای کاربران فعال در این فضا، فرصت بالقوه‌ی بسیار ارزشمندی در این حوزه وجود دارد. اما از سوی دیگر حجم محصولات عرضه‌شده در این بستر در زمینه‌های مختلف کالایی و خدماتی نیز بسیار عظیم است.

لذا می‌توان گفت در این مقوله حجم بسیار بزرگی از داده‌ها در مورد عرضه و تقاضای محصولات مختلف در دسترس است که استفاده‌ی بهینه از آن در جهت افزایش درآمد کسب‌وکارها مستلزم هوشمندسازی فرآیند تبلیغ و اطلاع‌رسانی کالا و خدمات است؛ به نحوی که هر کاربر متناسب با سابقه‌ی فعالیت‌ها و علاقمندی‌های موجود از او در داده‌های جمع‌آوری شده‌ی قبلی، با تبلیغاتی روبرو شود که احتمال پسندیدن آن برای او در بین همه‌ی گزینه‌های تبلیغی موجود بیشینه باشد. به بیان دیگر به جای تبلیغات هر محصول به عموم مخاطبان یک ناشر[۹] در فضای مجازی، از روش‌های هدفمند استفاده کنیم که در آن محصول نمایش‌داده‌شده به کاربران مختلف، متناسب با نیازهای هر یک بهینه‌سازی شده است تا احتمال انجام عمل مطلوب تبلیغ‌کننده[۱۰] توسط کاربر هدف بیشینه شود.

یکی از پراستفاده‌ترین ابزارها برای رسیدن به این هدف، سامانه‌های توصیه‌گر هستند. امروزه اهمیت کاربرد این سامانه‌ها را در سهم آنها از درآمد شرکت‌های بزرگ فعال در حوزه‌ی تجارت الکترونیک مانند Amazon، Netflix، و YouTube می‌توان دید. برای مثال شرکت Amazon در گزارشی اعلام کرده است که ٪۳۵ کل فروش این مجموعه متعلق به عملکرد سامانه‌ی توصیه‌گر آن است [۲].

[۹] Publisher
[۱۰] Advertiser

۱ـ۳ ادبیات موضوع

یک سامانه‌ی توصیه‌گر به عنوان یک سامانه‌ی چرخه‌ی بازخورد[۱۱] ممکن است با چند نوع مسئله‌ی پیش‌قدر از جمله پیش‌قدر محبوبیت [۳، ۴]، پیش‌قدر در مدل پیشین [۵، ۶، ۷]، و پیش‌قدر موقعیت [۸، ۹] روبرو باشد. مطالعات قبلی نشان داده است که ارزیابی سامانه‌های توصیه‌گر بدون توجه به مسئله‌ی پیش‌قدر، نتایج درستی از کارآیی سامانه‌ی مدنظر به دست نمی‌دهد. در حالی که تلاش در جهت رفع این مسئله می‌تواند منجر به بهبود کارآیی مدل‌ها شود [۵، ۹، ۱۰].

اکثر پژوهش‌های قبلی در جهت حل مسئله‌ی پیش‌قدر به دو دسته‌ی کلی تقسیم می‌شوند: روش‌های مبتنی بر یادگیری خلاف واقع[۱۲] [۱۱] و روش‌های ابتکاری[۱۳]. شاخص تمایل معکوس[۱۴] [۱۲] و اصل کمینه‌سازی ریسک خلاف واقع[۱۵] [۱۱] از ابزارهای اصلی مورد استفاده در دسته‌ی اول روش‌های مذکورند. در حالی که دسته‌ی دوم روش‌ها معمولا متکی بر فرض‌هایی در مورد مفقود بودن داده‌ها به شکلی غیرتصادفی[۱۶] هستند.

از یک زاویه‌ی دید دیگر، رویکردهای اخیر مبتنی بر یادگیری عمیق[۱۷] را که در حوزه‌ی سامانه‌های توصیه‌گر ارائه‌شده‌اند، می‌توان به دو دسته‌ی کلی تقسیم کرد که اتفاقا نتایج امیدوارکننده‌ای هم به همراه داشته‌اند: دسته‌ی نخست شامل روش‌هایی می‌شود که در صدد یادگیری نهفته‌سازی[۱۸]هایی برای محصولات هستند که هدفشان بهینه کردن پیش‌بینی جفت محصولات مشابه است [۱۳، ۱۴، ۱۵]. در دسته‌ی دوم روش‌هایی جای می‌گیرند که به دنبال یادگیری نهفته‌سازی‌هایی از توالی‌های کاربری[۱۹] هستند تا عمل پیش‌بینی بهترین محصول پیشنهادی بعدی را بهینه کنند [۱۶، ۱۷].

[۱۱] Feedback Loop System
[۱۲] Counterfactual Learning-Based
[۱۳] Heuristic-Based
[۱۴] Inverse Propensity Score (IPS)
[۱۵] Counterfactual Risk Minimization (CRM)
[۱۶] Missing Not At Random (MNAR)
[۱۷] Deep Learning
[۱۸] Embedding
[۱۹] User Sequence

۱ـ۴ اهداف تحقیق

از اهداف اصلی این پروژه در گام اول، مطالعه‌ی دقیق روش‌های ارائه‌شده در [۱] برای حل مسئله‌ی پیش‌قدر در داده‌های سامانه‌ی توصیه‌گر، و در گام دوم ارائه‌ی یک پیاده‌سازی جدید با استفاده از شبکه‌های عصبی تغذیه‌به‌جلو[۲۰] برای راهبردهای مبتنی بر برچسب پیشنهادشده در این مقاله بوده است. این در حالی است که نویسندگان [۱] راهبرد پیشنهادی خود را با استفاده از روش‌های تجزیه‌ی ماتریسی با رتبه‌ی پایین[۲۱] و با کمک شبکه‌های عصبی خودکدگذار[۲۲] پیاده‌سازی کرده‌اند. البته پیش از ارائه‌ی پیاده‌سازی جدید، به مطالعه‌ی دقیق و جزئی پیاده‌سازی اصلی مذکور و نیز مطالعه‌ی دقیق مقالات مرتبط دیگری مانند [۱۸] برای کسب تسلط کافی در حوزه‌ی مسئله پرداخته بودیم.

۱ـ۵ ساختار پایان‌نامه

این پایان‌نامه، به جز مقدمه‌ی آن، شامل پنج فصل اصلی به قرار زیر است:

- فصل ۲ شامل معرفی اصطلاحات و مفاهیم اولیه‌ی مورد نیاز برای مطالعه‌ی این پایان‌نامه است؛
- فصل ۳ به بیان ادبیات مسئله و مطالعات قبلی انجام گرفته در حوزه‌ی مسئله‌ی پیش‌قدر در سامانه‌های توصیه‌گر و راهکارهای رویارویی با آن می‌پردازد؛
- در فصل ۴ فعالیت‌های اصلی انجام‌گرفته طی این پژوهش مانند جزئیات دادگان‌ها و ساختار روش‌های مورد استفاده برای آموزش مدل‌ها به تفصیل شرح داده شده است؛
- فصل ۵ نتایج عددی آزمایش‌ها و نمودارها و جدول‌های مربوط به آن‌ها را در بر می‌گیرد؛
- و نهایتا فصل ۶ نیز به یک جمع‌بندی کلی و نهایی از روند اجرای پروژه و مسیرهای احتمالی پیش‌رو برای ادامه‌ی مطالعه‌ی مسئله‌ی هدف اختصاص دارد.

---

[۲۰] Feedforward neural networks
[۲۱] Low-rank matrix factorization
[۲۲] Autoencoders

فصل ۲

مفاهیم اولیه

۲ـ۱ تعاریف

یادگیری تقویتی[۱]: یکی از سه الگوواره‌ی اصلی در یادگیری ماشین است که در آن یادگیری عامل[۲] هوشمند از طریق تعامل او با محیط صورت می‌گیرد. به این صورت که در هر مرحله، عامل یکی از حرکات[۳] ممکن خود را انجام می‌دهد. این حرکت برای او پاداشی[۴] به همراه دارد و حالت[۵] محیط نیز تحت تأثیر این حرکت او تغییر می‌کند. در این نوع یادگیری، هدف عامل یافتن حرکاتی است که انجام آنها برآیند پاداش او را در طول زمان بیشینه کند.

پاداش: در ادبیات یادگیری تقویتی، امتیازی است که محیط در اثر هر حرکت یک عامل به او می دهد که بسته به میزان سودمندی حرکت در جهت هدف عامل، می‌تواند از نوع مثبت، منفی، یا صفر باشد.

داده‌های مشاهده‌نشده[۶]: به جفت‌هایی از کاربرـ محصول اطلاق می‌شود که به ازای آنها امتیاز[۷] آن کاربر به آن محصول را در اختیار نداریم و منظور از «مشاهده‌نشده» هم در واقع همان برچسب خروجی

[۱] Reinforcement Learning
[۲] Agent
[۳] Actions
[۴] Reward or feedback
[۵] State
[۶] Unobserved data
[۷] Rating

به ازای این داده‌ی ورودی (جفت مذکور) است که از آن بی‌اطلاع هستیم.

نمونه[۸]: هر عضو از یک مجموعه‌داده[۹] و در واقع هر واحد داده‌ی جمع‌آوری شده را گویند.

یادگیری بانظارت[۱۰]: دسته‌ای از روش‌های یادگیری ماشین را شامل می‌شود که در آن مجموعه‌ای از نمونه‌داده‌های برچسب‌دار جمع‌آوری‌شده در اختیار داریم. در واقع هر نمونه شامل دو بخش ویژگی‌ها[۱۱] و برچسب[۱۲] می‌شود و هدف نهایی از توسعه‌ی مدل هم این است که با دادن یک مجموعه‌ویژگی به آن، برچسب خروجی متناظر را پیش‌بینی کند.

ناموجود به شکل غیرتصادفی[۱۳]: یکی از انواع حالات داده‌های ناموجود[۱۴] در بحث‌های آماری است. این حالت به این معناست که احتمال ناموجود بودن داده‌ها بنا به دلایلی که برای پژوهشگر نامشخص است، متغیر خواهد بود. برای مثال یک ترازو ممکن است با گذر زمان فرسوده شود و مقادیر ناموجود بیشتری نسبت به قبل تولید کند؛ در حالی که ممکن است پژوهشگر اساسا موفق به کشف چنین علتی در مشاهدات خود از داده‌های ناموجود نشود.

یک مثال دیگر از این حالت داده‌های ناموجود در پژوهش‌های مربوط به نظرخواهی عمومی است و زمانی اتفاق می‌افتد که افراد با دیدگاه‌های ضعیف‌تر کمتر به نظرخواهی‌ها پاسخ دهند. داده‌های ناموجود به شکل غیرتصادفی پیچیده‌ترین نوع مسئله‌ی داده‌های ناموجود است و راهبردهای حل این معضل شامل یافتن داده‌های بیشتر درباره‌ی علل ناموجود بودن می‌شود. همچنین تحلیل‌های موسوم به «چه می‌شود اگر»[۱۵] با هدف مشاهده‌ی اینکه نتایج چقدر نسبت به تغییر سناریوهای مختلف حساس‌اند، در زمره‌ی راهبردهای حل مسئله‌ی MNAR است [۱۹].

جمع‌سپاری[۱۶]: به جمع‌آوری خدمات، ایده‌ها، یا محتوای تولیدشده از طریق مشارکت گروه بزرگی از افراد گفته می‌شود که مستقیما ارتباطی با تجارت متقاضی آن خدمت ندارند و صرفا به شکل کار از راه دور و غیرمستقیم با آن همکاری می‌کنند. در واقع جمع‌سپاری روشی جدید برای تولید ارزش است که در آن یک مجموعه‌ی تجاری به جای استفاده از کارکنان یا سهامداران خود، از اجتماعی از افراد ناشناس بهره می‌برد.

---

[۸] Sample
[۹] Dataset
[۱۰] Supervised Learning
[۱۱] Features
[۱۲] Label
[۱۳] Missing Not At Random/ Not Missing At Random (MNAR/NMAR)
[۱۴] Missing data
[۱۵] What-if analyses
[۱۶] Crowdsourcing

تورکینگ مکانیکی آمازون[۱۷]: یک نمونه بازار استفاده از جمع‌سپاری برای افراد و شرکت‌های تجاری است. امور جمع‌سپاری‌شده می‌تواند شامل هر چیزی از ارزیابی ساده‌ی داده‌ها و پژوهش گرفته تا امور بیشتر وابسته به نظرات شخصی مانند شرکت در نظرسنجی‌ها، تعدیل محتوا، و ... می‌شود. به هر یک از افرادی که از طریق این سازوکار، به سایر افراد یا شرکت‌ها خدمت‌رسانی می‌کنند، به اصطلاح Turker گفته می‌شود.

نمایش یک‌ـ‌بارز[۱۸]: گونه‌ای از نمایش برداری تنک است که در آن تمام مؤلفه‌های بردار به جز دقیقا یکی از آنها صفر هستند و مقدار آن یک مؤلفه برابر با یک است. یکی از کاربردهای این نوع نمایش برای مقداردهی ویژگی‌ها دسته‌ای[۱۹] است که در آن نمونه‌داده تنها می‌تواند یکی از چند حالت ممکن از یک ویژگی را داشته باشد؛ برای مثال اگر بازه‌های سنی ممکن برای انسان‌ها را به چهار دسته‌ی زیر ۲۰ سال، بین ۲۰ تا ۴۰ سال، بین ۴۰ تا ۶۰ سال، و بیش از ۶۰ سال افراز کنیم، ویژگی سن هر فرد را می‌توان در قالب یک بردار یک‌ـ‌بارز چهار بعدی نشان داد.

جستجوی مشبکه‌ای[۲۰]: در بحث تنظیم ابرپارامترهای یک مدل برای یافتن بهترین آرایش از آنها، به نوعی از جستجو گفته می‌شود که در آن همه‌ی ترکیب‌های ممکن از مقدار ابرپارامترها (با توجه به مجموعه‌ی مقادیر از پیش معین برای جستجوی هر کدام) امتحان می‌شوند و هر کدام که بهترین نتایج را روی داده‌های ارزیابی بگیرد، به عنوان بهترین آرایش برگزیده می‌شود. نام آن هم از شکل مشبکه می‌آید که مانند یک دستگاه مختصات با مقادیر گسسته‌ی مدرج روی هر کدام از محورهاست و مختصات هر نقطه از فضای این مشبکه متناظر با یک مقداردهی منحصر به فرد از مقادیر ممکن روی محورهاست.

شبکه‌ی عصبی روبه‌جلو[۲۱]: یکی از انواع شبکه‌های عصبی در بحث مدل‌های یادگیری ماشین است که در آن شبکه‌ی عصبی یادگیرنده از تعدادی لایه‌ی متوالی تشکیل شده‌است. هر لایه شامل تعدادی نورون است و نورون‌های هر لایه می‌توانند تنها به نورون‌های لایه‌ی بلافاصله بعد یا قبل از خود متصل باشند. مطابق ساختار عمومی شبکه‌های عصبی، هر نورون یک تابع فعال‌ساز غیرخطی (مانند سیگموئید[۲۲]، تانژانت هذلولوی[۲۳]، یا ...) را روی ترکیب خطی وزن‌داری از ورودی‌های خود که با یک جمله‌ی پیش‌قدر جمع شده است اعمال می‌کند و حاصل این محاسبه خروجی نورون خواهد بود.

[۱۷] Amazon Mechanical Turk (MTurk)
[۱۸] One-hot representation/encoding
[۱۹] Categorical
[۲۰] Grid search
[۲۱] Feedforward neural network
[۲۲] Sigmoid
[۲۳] Hyperbolic Tangent (tanh)

منحنی مشخصه‌ی عملیاتی گیرنده[۲۴]: نموداری است که کیفیت عملکرد یک مدل دسته‌بند دودویی[۲۵] در تعیین درست برچسب‌های هر نمونه را همزمان با تغییر در مقدار آستانه‌ی تفکیک مدل، ارزیابی می‌کند. یکی از مشهورترین مؤلفه‌های این نمودار سطح زیر آن است که در ادبیات موضوع، آن را با **AUC**[۲۶] نشان می‌دهند و ما نیز در ادامه‌ی این پایان‌نامه از همین نمادگذاری استفاده می‌کنیم. در فصل ۴ به این نمودار و سطح زیر آن مفصل‌تر خواهیم پرداخت.

هزینه‌ی لگاریتمی[۲۷] **یا هزینه‌ی آنتروپی متقاطع**[۲۸]: یک تابع هزینه‌ی مورد استفاده در یادگیری بانظارت است برای ارزیابی برچسب‌های خروجی پیش‌بینی‌شده توسط مدل‌های دسته‌بند مانند رگرسیون لجستیک[۲۹] یا شبکه‌های عصبی.

یکی از حالات خاص مشهور این تابع هزینه، همان حالت مربوط به مسئله‌ی دسته‌بندی دوکلاسه است که این تابع هزینه در این حالت مشهور به آنتروپی متقاطع دودویی[۳۰] است. فرض کنید در مسئله‌ای برچسب خروجی هر نمونه صفر یا یک باشد. اگر یک نمونه با مجموعه‌ی ویژگی‌های x برچسب خروجی‌اش برابر با y باشد و مقدار احتمال پیش‌بینی شده‌ی مدل برای یک بودن برچسب آن، $P(y=۱|x)=\hat{y}$ باشد، می‌توان نوشت

$$P(y|x)=\begin{cases}\hat{y} & y=۱\\ ۱-\hat{y} & y=۰\end{cases}=\hat{y}^{y}(۱-\hat{y})^{۱-y}. \qquad (۱-۲)$$

لذا تابع هزینه ــ که در آن ll همان مخفف log-loss است ــ از رابطه‌ی زیر محاسبه می‌شود:

$$ll(y,\hat{y})=-log(P(y|x))=-\left(ylog(\hat{y})+(۱-y)log(۱-\hat{y})\right) \qquad (۲-۲)$$

اگر هم تعداد نمونه‌ها به جای یک مورد، در حالت کلی برابر با N باشد، تابع هزینه معمولا به صورت میانگین هزینه‌ی هر یک از نمونه‌ها و از رابطه‌ی

[۲۴] Receiver Operating Characteristic (ROC) curve
[۲۵] Binary classifier
[۲۶] Area Under the Curve
[۲۷] Log loss
[۲۸] Cross-entropy loss
[۲۹] Logistic Regression
[۳۰] Binary Cross-Entropy (BCE)

$$ll(Y,\hat{Y}) = -\frac{۱}{N}\sum_{i=۱}^{N}(y_i log(\hat{y}_i) + (۱ - y_i) log(۱ - \hat{y}_i)) \qquad (۳\_۲)$$

محاسبه می‌شود که در آن $Y = \{y_۱, y_۲, \ldots, y_N\}$ و $\hat{Y} = \{\hat{y}_۱, \hat{y}_۲, \ldots, \hat{y}_N\}$.

خطای میانگین مربعی[۳۱]: یک تابع هزینه برای ارزیابی کیفیت عملکرد یک مدل از طریق مقایسه‌ی خروجی‌های پیش‌بینی‌شده‌ی یک مدل با خروجی‌های درست از پیش‌دانسته روی مجموعه‌ای از نمونه‌هاست. مقدار این تابع هزینه از رابطه‌ی

$$MSE(Y,\hat{Y}) = \frac{۱}{N}\sum_{i=۱}^{N}(y_i - \hat{y}_i)^۲ \qquad (۴\_۲)$$

محاسبه می‌شود که در آن $Y = \{y_۱, y_۲, \ldots, y_N\}$ و $\hat{Y} = \{\hat{y}_۱, \hat{y}_۲, \ldots, \hat{y}_N\}$ به ترتیب مجموعه‌ی مقادیر خروجی درست و مجموعه‌ی مقادیر خروجی پیش‌بینی‌شده‌ی مدل برای مجموعه‌ی ورودی‌های $X = \{x_۱, x_۲, \ldots, x_N\}$ هستند.

[۳۱] Mean squared error

فصل ۳

کارهای پیشین

در این فصل، مطالعات انجام‌شده روی هر یک از انواع پیش‌قدر را بررسی می‌کنیم که یک سامانه‌ی توصیه‌گر ممکن است با آن روبرو باشد

۳ـ۱ پیش‌قدر محبوبیت

- پیش‌قدر محبوبیت[۱] نوعی از مسئله‌ی پیش‌قدر است که در [۳] ادعا شده بسیاری از الگوریتم‌های سامانه‌های توصیه‌گر از آن رنج می‌برند. این مسئله بیان می‌دارد که موارد محبوب، بیشتر پیشنهاد داده می‌شوند و موارد با محبوبیت قبلی اصلا فرصتی برای تبلیغ نمی‌یابند و یا به ندرت پیشنهاد داده می‌شوند. هرچند موارد محبوب بلنددنباله[۲] دقیقا همان‌هایی هستند که غالبا هم پیشنهادهایی مطلوب به حساب می‌آیند. پیش‌قدر محبوبیت یک مانع جدی برای اثرگذاری سامانه‌های توصیه‌گر است؛ چرا که این سامانه‌های غربالگر تعاملی معمولا در پیشنهادهای خود بر موارد محبوب از نظر امتیاز کاربران بسیار بیشتر از موارد بلنددنباله‌ی دیگر تأکید می‌کنند. اگرچه موارد محبوب عمدتا گزینه‌های خوبی برای پیشنهاد محسوب می‌شوند اما در عین حال به خودی خود هم شناخته شده هستند. لذا سامانه‌های توصیه‌گر باید به دنبال برقراری یک تعادل بین موارد با محبوبیت زیاد و کم باشند.

---

[۱] Popularity bias
[۲] Long-tail

بازاری که از پیش‌قدر محبوبیت رنج ببرد، فرصت کشف کردن قابلیت‌های محصولات مهجور را از دست می‌دهد و لذا بنا بر تعریف، تحت سلطه‌ی چند نام[۳] قرار خواهد گرفت. در نتیجه چنین بازاری یک بافت عمدتا همگن پیدا می‌کند و ظرفیت کمتری برای نوآوری و خلاقیت خواهد داشت [۳].

۳ـ۲ پیش‌قدر موقعیت

- یک مشکل اساسی و رایج در حوزه‌ی بازیابی اطلاعات مسئله‌ی پیش‌قدر موقعیت[۴] است. این مسئله در واقع بیانگر تمایل کاربران به توجه یا تعامل با مواردی از یک فهرست مرتب هستند که در جایگاه‌های خاصی از آن قرار گرفته‌اند؛ حتی اگر ارتباطشان با نیاز واقعی کاربر کمتر از سایر موارد باشد [۲۰].

- این مسئله چالش‌هایی در بحث ارزیابی عملکرد سامانه‌های توصیه‌گر نیز ایجاد می‌کند. میزان تأثیرگذاری پیشنهادهای یک سامانه‌ی توصیه‌گر معمولا به صورت ضمنی و بر اساس دنبال کردن روند کلیک‌های کاربران آن تعیین می‌شود. اما به دلیل وجود پیش‌قدر مذکور، احتمالات به دست آمده از تحلیل این داده‌ها در مورد تعامل کاربران با یک مورد پیشنهادی در لیست، لزوما بیانگر ارتباط مطلق آن مورد با نیاز واقعی کاربر نیست. مثلا اگر این موارد پیشنهادی اسناد خروجی‌داده‌شده‌ی یک سامانه‌ی بازیابی اطلاعات باشند، احتمال واقعی کلیک شدن یک سند پیشنهادی علاوه بر میزان ارتباط[۵] سند با پرسمان ورودی، به موقعیت سند مذکور در لیست نتایج پیشنهاد شده هم بستگی دارد. شواهدی از جمله آزمایش‌های تعقیب چشم کاربر وجود این نوع پیش‌قدر را تأیید می‌کنند. نتایج این گونه آزمایش‌ها نشان می‌دهد که کاربران کمتر به نتایج انتهایی لیست پیشنهادی توجه می‌کنند. لذا ارزیابی‌های صرفا مبتنی بر داده‌هایی مثل کلیک کاربران بدون توجه به اثر این نوع پیش‌قدر می‌توانند نتایجی گمراه‌کننده به همراه داشته باشند [۲۰، ۲۱].

[۳] Brand
[۴] Position bias
[۵] Relevance

۳ـ۳ پیش‌قدر مدل پیشین

از آنجا که تمرکز ما در انجام این مطالعه مسئله‌ی پیش‌قدر مدل پیشین[۶] بوده است و قصد داریم رویکردهای حل این مسئله با بهره‌گیری از داده‌های جمع‌آوری شده به صورت یکنواخت را بررسی کنیم، پژوهش‌های مرتبط قبلی پیرامون نوع اخیر پیش‌قدر را با تفصیل بیشتری بررسی می‌کنیم.

- در [۵] به این اشاره شده که مدل استقراریافته‌ی کنونی نمونه‌های آموزشی تولیدشده برای آینده را به طرز چشمگیری تحت تأثیر قرار می‌دهد و نویسندگان اثر منفی این چرخه‌ی بازخورد را مشاهده کرده‌اند. آنها که در کار خود یک سامانه‌ی توصیه‌گر روی شبکه‌ی اجتماعی Pinterest را مطالعه کرده‌اند، گزارش داده‌اند که پس از چند ماه داده‌های آموزشی عملا فقط درگیری با pin هایی را نشان می‌داده که مدل کنونی قبلا به آنها رتبه‌ی بالایی داده است.

  لذا برای غلبه بر این مشکل در داده های آموزشی، درصد اندکی از ترافیک را به مجموعه‌ای بدون پیش‌قدر از داده‌ها اختصاص داده‌اند تا برای این درصد اندک از درخواست‌ها نمونه‌ای تصادفی از همه‌ی نامزدهای موجود را بدون رتبه‌بندی پیشنهاد شوند. بدین ترتیب داده‌های آموزشی دیگر تحت تأثیر مدل پیشین قرار نخواهند گرفت. برای آنکه پیشنهادهای رتبه‌بندی نشده و کاملاً تصادفی تجربه و رضایت هر کاربر مشخص را خیلی تخریب نکند، هر کاربر فقط برای یک زیرمجموعه‌ی بسیار کوچک از پرسمان[۷]های ارسالی‌اش به سیستم با این نوع پیشنهادهای کاملاً تصادفی روبرو می‌شود.

- مدل‌های پیشرو[۸]ی موجود در یادگیری ماشین هنوز هم مسئله‌ی یک سامانه‌ی توصیه‌گر را به یکی از دو صورت زیر توصیف می‌کنند:

  ۱. یک مسئله‌ی یادگیری فاصله بین جفت‌های محصولات یا جفت‌های کاربران و محصولات که در آن خروجی‌ها را با استفاده از معیارهای پیش‌بینی پیوند یا تکمیل ماتریس مثل خطای میانگین مربعی و سطح زیر منحنی[۹] بسنجیم.

  ۲. یک مسئله‌ی پیش‌بینی محصول بعدی که رفتار کاربر را مدل‌سازی و تلاش می‌کند تا پیش‌بینی کند حرکت بعدی او چیست. در این رویکرد از معیارهای رتبه‌بندی مانند Precision@K

[۶] Previous model bias
[۷] Query
[۸] State-of-the-art
[۹] Area Under the Curve (AUC)

و Normalized Discounted Cumulative Gain (NDCG) استفاده می‌شود.

اما هر دو رویکرد مذکور، در مدل‌کردن ذات مداخله‌گر موضوع توصیه‌گری نقص دارند. این ذات مداخه‌گر به این معناست که ما در فرآیند توصیه صرفا قصد مدل‌کردن رفتار طبیعی کاربر را نداریم؛ بلکه در حقیقت قصد داریم رفتار او را نیز مبتنی بر یک هدف از پیش تعیین‌شده‌ی خارجی جهت‌دهی کنیم. ازجمله مثال‌های این مداخله‌گری، تلاش در جهت افزایش نوع خاصی از فعالیت کاربر مانند خرید برخی اجناس، مشاهده‌ی برخی فیلم‌ها، و یا اقدام به سفارش یک کارت اعتباری است. از یک دیدگاه ایده‌آل‌گرایانه، ارزیابی میزان افزایش فعالیت کاربر در جهت مطلوب، در مقایسه با شرایطی انجام می‌گیرد که هیچ توصیه‌ای به او نشان داده نشده باشد. [۱۸]

به بیان رایج در ادبیات علّی، نویسندگان [۱۸] قصد یافتن سیاست پیشنهاددهی درمانی[۱۰] بهینه‌ای را دارند که پاداش[۱۱] را با توجه به سیاست پیشنهاددهی کنترلی برای هر کاربر بیشینه می‌کند. این سیاست به نام اثر درمان درمانی منحصر به فرد[۱۲] نیز شناخته می‌شود. سیاست کنترلی، حالت کنونی سامانه را شکل می‌دهد و که این سامانه می‌تواند یک سامانه‌ی بدون توصیه‌گر یا دارای یک توصیه‌گر ابتدایی[۱۳] باشد که قصد بهبود آن با استفاده از بازخوردهای ثبت‌شده را داریم.

در [۱۸] نسخه‌ای از روش‌های ماتریسی با تغییرات ساده نسبت به انواع مرسوم آن معرفی شده است که از مجموعه‌ای کوچک از توصیه‌های انجام‌گرفته به صورت تصادفی (به بیان دیگر یکنواخت) برای تولید نمایش کاربران و محصولات بهره می‌برد. ادعای نویسندگان [۱۸] هم بر این است که فاصله‌ی بین جفت‌های کاربر و محصول در روش پیشنهادی‌شان، نسبت به رویکردهای سنتی، همخوانی بیشتری با ITE متناظر دارد.

- در مطالعه‌ای دیگر، نویسندگان [۱] چارچوبی عمومی به نام چارچوب استحصال دانش برای پیشنهاددهی خلاف واقع[۱۴] ارائه کرده‌اند تا به شکل مؤثری از داده‌های یکنواخت بهره بگیرند. این چارچوب چهار دسته روش دارد که شامل روش‌های استحصال مبتنی بر برچسب[۱۵]، مبتنی بر ویژگی[۱۶]، مبتنی بر نمونه[۱۷]، و مبتنی بر ساختار مدل[۱۸] می‌شود. هر دسته روش بر جنبه‌های

[۱۰] Treatment
[۱۱] Reward
[۱۲] Individual Treatment Effect (ITE)
[۱۳] Baseline
[۱۴] Knowledge Distillation Framework for Counterfactual Recommendation (KDCRec)
[۱۵] Label-based
[۱۶] Feature-based
[۱۷] Sample-based
[۱۸] Model structure-based

متفاوتی برای استخراج دانش مفید از داده‌های یکنواخت تمرکز می‌کند که در نهایت برای بهبود نتایج یادگیری داده‌های دارای پیش‌قدر استفاده خواهد شد.

فصل ۴

روش پیشنهادی

در این فصل ابتدا رویکردهایی را که برای حل مسئله‌ی پیش‌قدر در مدل پیشین با استفاده از داده‌های یکنواخت بررسی کرده‌ایم، مطرح می‌کنیم. ایده‌ی اصلی دو روش اول از راهبردهای معرفی‌شده در ماژول مبتنی بر برچسب[۱] ذیل چارچوب معرفی‌شده در [۱] است.

۴ـ۱ روش‌های مبتنی بر برچسب

در ادامه‌ی این بخش، مجموعه‌ی داده‌های دارای پیش‌قدر (غیریکنواخت) را با S_c و مجموعه‌ی داده‌های بدون پیش‌قدر (یکنواخت) را با S_t نمایش می‌دهیم. یک توصیف شهودی این است که هنگام آموزش یک مدل روی S_c، آن مدل از برچسب‌های منتسب[۲] از روی S_t برای اصلاح پیش‌قدر موجود در پیش‌بینی‌های خود استفاده می‌کند.

حال دو راهبرد اصلی پیشنهاد شده ذیل این روش‌ها را در ادامه معرفی می‌کنیم.

[۱] Label-based
[۲] Imputed labels

۴ـ۱ـ۱ راهبرد پل

نخستین راهبرد بررسی‌شده ذیل روش‌های مبتنی بر برچسب، راهبرد پل[۳] است که ابتدا به معرفی نمادگذاری استفاده‌شده در آن می‌پردازیم. یک زیرمجموعه‌ی نمونه‌برداری‌شده از همه‌ی داده‌های مشاهده‌نشده را با S_a نمایش می‌دهیم. مدل M_c یک مدل غیریکنواخت و مدل M_t یک مدل یکنواخت است که به ترتیب روی مجموعه‌داده[۴]های S_c و S_t آموزش می‌بینند. اما برای اصلاح پیش‌قدر مدل M_c از همان داده‌های مجموعه‌ی S_a به مثابه‌ی **پلی** استفاده می‌کنیم که بین دو مدل یکنواخت و غیریکنواخت قرار می‌گیرد. به این صورت که S_a مجموعه‌داده‌ای می‌شود که هر دو مدل M_t و M_c به طور مشترک روی آن آموزش داده می‌شوند (علاوه بر یک مجموعه‌داده‌ی مختص هر کدام که قبل‌تر ذکر شد) و لذا انتظار داریم که از این طریق، خروجی‌های دو مدل روی S_a به هم نزدیک شوند و کاهش پیش‌قدر مدل M_c از این طریق حاصل شود. تابع هدف[۵] نهایی مورد استفاده در این راهبرد به صورت

(۱ـ۴)

$$\min_{W_c,W_t} \frac{1}{|S_c|}\sum_{(i,j)\in S_c} l(y_{ij},\hat{y}^c_{ij})+\frac{1}{|S_t|}\sum_{(i,j)\in S_t} l(y_{ij},\hat{y}^t_{ij})+\frac{1}{|S_a|}\sum_{(i,j)\in S_a} l(\hat{y}^c_{ij},\hat{y}^t_{ij})+\lambda_c R(W_c)+\lambda_t R(W_t)$$

است که در آن W_c و W_T به ترتیب مجموعه‌ی پارامترهای مدل‌های M_C و M_t را نشان می‌دهند و تابع دو ورودی l هم هر تابع هزینه[۶]ی دلخواه می‌تواند باشد که در پیاده‌سازی ما تابع خطای میانگین مربعی به این منظور انتخاب شده‌است. نمادهای i و j در عبارت بالا به ترتیب بیانگر کاربر i و محصول j هستند. همچنین y_{ij} خروجی (امتیاز) واقعی کاربر i به محصول j، $\hat{y}^c_{ij}$ خروجی پیش‌بینی‌شده‌ی مدل M_c، و $\hat{y}^t_{ij}$ خروجی پیش‌بینی‌شده‌ی مدل M_t به ازای جفت ورودی کاربر i و محصول j هستند. در نهایت، R نماینده‌ی جمله‌ی منظم‌ساز[۷] و λ_t و λ_c پارامترهای منظم‌سازی هستند.

۴ـ۱ـ۲ راهبرد تصفیه

راهبرد دوم از این مجموعه روش‌ها، راهبرد تصفیه[۸] نام دارد. اینجا بر خلاف روش قبلی، دو مدل همزمان آموزش داده نمی‌شوند؛ بلکه مدل M_c روی مجموعه‌داده‌ی غیریکنواخت S_c آموزش داده می‌شود و بالتبع

[۳] Bridge strategy
[۴] Dataset
[۵] Objective function
[۶] Loss function
[۷] Regularization term
[۸] Refine strategy

نتایج این مدل دارای پیش‌قدر خواهد بود. در واقع نکته اینجاست که همه‌ی بازخوردهای مثبت و منفی جمع‌آوری‌شده از کاربران به ترتیب به عنوان برچسب خروجی ۱ و ۱ـ در نظر گرفته می‌شوند در حالی که این در تعیین مقادیر این برچسب‌ها باید یک توزیع ترجیح[۹] هم لحاظ شود. لذا در این روش قصد بر این است که با بهره‌گیری از S_t توزیع درست برچسب‌ها روی S_c بهتر استخراج و سپس *تصفیه* شود.

بدین منظور یک مدل M_t را پیش از شروع آموزش مدل اصلی M_c روی داده‌های S_t آموزش می‌دهیم که پس از آن، مدل M_t را اصطلاحا مدل پیش‌آموزش‌دیده[۱۰] می‌نامند. مدل M_t می‌تواند صرفا با یک تابع هزینه و یک جمله‌ی منظم‌ساز رایج (مثلا به ترتیب خطای میانگین مربعی و منظم‌ساز L۲) و بدون بهره‌گیری از ایده‌ی خاص دیگری آموزش ببیند. سپس از آن برای پیش‌بینی همه‌ی نمونه‌های موجود در S_c استفاده می‌کنیم. این برچسب‌های منتسب از طریق یک پارامتر وزن‌دهی قابل تنظیم[۱۱] به نام α با برچسب‌های اصلی پیش‌بینی‌شده روی S_c ترکیب می‌شوند که نهایتا مدل M_c با پیش‌قدر کمتری آموزش داده شود. همچنین برای جلوگیری از اختلاف توزیع بین برچسب‌های منتسب و برچسب‌های اصلی، برچسب‌های منتسب را با استفاده از تابعی مانند N نرمال‌سازی[۱۲] می‌کنیم. تابع هدف نهایی برای این راهبرد به صورت

$$\min_{W_c} \frac{۱}{|S_c|} \sum_{(i,j)\in S_c} l\left(y_{ij} + \alpha N\left(\hat{y}^t_{ij}\right), \hat{y}^c_{ij}\right) + \lambda_c R(W_c) \qquad (۲-۴)$$

است. توجه شود که مقدار α به یک تعبیر، اهمیت برچسب‌های منتسب را کنترل می‌کند.

۴ـ۲ مجموعه‌داده‌ها

در این مطالعه از دو دادگان به نام‌های Coat و Yahoo!R۳ برای انجام آزمایش‌ها استفاده شده است که در ادامه به معرفی هر کدام خواهیم پرداخت.

- دادگان Coat مجموعه‌ای از داده‌های ناموجود به شکل غیرتصادفی (MNAR) از مشتریان کت در یک فروشگاه لباس‌فروشی برخط را شبیه‌سازی می‌کند. برای تولید داده‌های آموزشی یک واسط

[۹] Preference distribution
[۱۰] Pre-trained model
[۱۱] Tunable
[۱۲] Normalize

ساده‌ی فروشگاه اینترنتی در اختیار افراد مشغول در تورکینگ مکانیکی آمازون قرار گرفته و از آنها خواسته شده کتی را که بیش از همه مایل به خرید آن هستند، بیابند. هر نفر ۲۴ مورد از کت‌هایی را که در حین جستجوهایش به آنها برخورده (موسوم به خودمنتخب[۱۳] و عملا متأثر از پیش‌قدر مدل پیشین که اینجا مدل پیشین به نوعی همان سلیقه‌ی قبلی کاربران است)، به علاوه‌ی ۱۶ مورد کت تصادفی (به عنوان داده‌های یکنواخت و بدون پیش‌قدر) را در مقیاس ۱ تا ۵ امتیازدهی کرده است. دادگان شامل امتیازدهی‌های ۲۹۰ نفر روی انباری از ۳۰۰ قلم کالاست [۱۲].

- دادگان Yahoo!R۳ شامل امتیازات جمع‌آوری برای تعدادی آهنگ از دو منبع مختلف است. منبع اول مربوط به تعاملات معمول کاربران با خدمات موسیقی Yahoo! است. اما منبع دوم شامل امتیازات داده‌شده به تعدادی آهنگ تصادفی است که گروه پژوهش Yahoo! طی یک نظرسنجی برخط آن را به انجام رسانده است. در این دادگان ۱۵۴۰۰ کاربر و ۱۰۰۰ آهنگ متمایز وجود دارد. از هر کاربر حداقل ۱۰ مورد امتیازدهی به آهنگ‌های خودمنتخب ثبت شده است. همچنین هر یک از ۵۴۰۰ کاربر نخست دقیقا به ۱۰ آهنگ تصادفی پیشنهادشده به آنها امتیاز داده‌اند. لذا در مجموع این دادگان شامل حدود ۳۰۰۰۰۰ امتیاز برای آهنگ‌های خودمنتخب و دقیقا ۵۴۰۰۰ امتیاز ثبت‌شده برای آهنگ‌های پیشنهادشده به صورت تصادفی به کاربران است. ضمنا تخصیص شناسه‌ی عددی به هر یک از کاربران و آهنگ‌ها نیز به طور کاملا تصادفی انجام گرفته است. داده‌های تصادفی (یکنواخت و بدون پیش‌قدر) در بازه‌ی ۲۲ اوت تا ۷ سپتامبر ۲۰۰۶ و داده‌های خودمنتخب (به نوعی متأثر از پیش‌قدر سلیقه‌ی پیشین کاربران) بین سال‌های ۲۰۰۲ تا ۲۰۰۶ جمع‌آوری شده‌اند [۲۲].

در ادامه به تفصیل مراحل و اجزای آزمایش‌های انجام‌گرفته روی این دو دادگان برای ارزیابی راهبردهای پیشنهادی حل مسئله‌ی پیش‌قدر مدل پیشین را شرح می‌دهیم.

[۱۳] Self-selected

۴ـ۳ آزمایش‌ها روی دادگان Coat

۴ـ۳ـ۱ جزئیات دادگان خام

دادگان خام Coat شامل چند فایل از جمله ویژگی‌های کاربران و کالاها، مقادیر شاخص تمایل[۱۴]، و امتیازات جمع‌آوری‌شده به روش یکنواخت و غیریکنواخت ـ به شرحی که پیش‌تر آمد ـ می‌شود که ما در آزمایش‌های خود از فایل‌های مربوط به داده‌های امتیازات یکنواخت و غیریکنواخت و ویژگی‌های کاربران و کالاها استفاده می‌کنیم. هر یک از دو فایل یکنواخت و غیریکنواخت در واقع یک ماتریس تنک[۱۵] شامل ۲۹۰ سطر و ۳۰۰ ستون است که درایه‌ی سطر iام و ستون jام آن، اگر ناصفر باشد، نشان‌دهنده‌ی امتیاز داده‌شده‌ی کاربر با شناسه‌ی i به کالای با شناسه‌ی j است که برای سادگی از این پس این موارد را به ترتیب کاربر iام و کالای jام نیز می‌نامیم. اگر هم این درایه صفر باشد، یعنی در روش جمع‌آوری داده‌ی مربوط به آن فایل (یکنواخت یا غیریکنواخت) کاربر iام با کالای jام روبرو نشده و امتیازی هم برای او ثبت نکرده است.

فایل ویژگی‌های کاربران یک ماتریس دودویی ۲۹۰ در ۱۴ است که هر سطر از آن مجموعه‌ای از ویژگی‌های کاربر آن سطر را به صورت یک بردار دودویی نمایش می‌دهد. در واقع هر بردار حاصل الحاق چند زیربردار یک‌ـبارز است که هر کدام از این زیربردارها مقدار یکی از ویژگی‌های کاربر را مشخص می‌کنند. مشابهاً فایل ویژگی‌های کالاها نیز یک ماتریس دودویی ۳۰۰ در ۳۳ است. توضیح تمام ویژگی‌های کاربران و کالاها نیز در فایل‌هایی جداگانه در همان مجموعه‌فایل‌های دادگان خام Coat ذکر شده است.

در دادگان Coat مجموعا فقط چهار ویژگی جنسیت (۲ مقدار)، بازه‌ی سنی (۶ مقدار)، محل زندگی (۳ مقدار)، و میزان علاقه به مد روز (۳ مقدار) برای کاربران ثبت شده است و ترکیب همه‌ی مقادیر ممکن این ویژگی‌ها طبق اصل ضرب حداکثر $۲ \times ۶ \times ۳ \times ۳ = ۱۰۸$ حالت مختلف خواهد داشت که البته توزیع وقوع این ۱۰۸ حالت هم یکنواخت نیست. لذا بسیاری از کاربران موجود در دادگان بردار نمایش ویژگی‌شان دقیقا یکسان خواهد بود؛ در حالی که امتیازدهی‌های متفاوتی به کالاهایی مختلف از آنها ثبت شده است. چنین وضعیتی کیفیت نتایج الگوریتم‌های یادگیری ماشین را کاهش می‌دهد (این موضوع به صورت عینی نیز در آزمایش‌های اولیه مشاهده شد)؛ چون الگوریتم با ورودی‌هایی

[۱۴] Propensity
[۱۵] Sparse

کاملا یکسان با خروجی‌های متفاوت روبرو می‌شود و به شکل منطقی نمی‌تواند ارتباط بین ورودی‌ها و خروجی‌های مسئله را مدل کند. این وضعیت حتی برای انسان‌ها هم می‌تواند پیچیده باشد؛ چرا که اگر ندانیم پشت هر کدام از این بردارها واقعا افراد مختلفی بوده‌اند و صرفا بخواهیم مجموعه مقادیر ویژگی‌ها را معیار تمایز نمونه‌ها قرار دهیم، در واقع انگار یک نمونه‌ی واحد بدون هیچ اطلاعات بیشتری در دفعات مختلف نظراتی کاملا متفاوت، و بعضا حتی متضاد روی کالاهایی یکسان، داشته است. لذا تحلیل این وضعیت و به دست آوردن مدلی برای پیش‌بینی انتخاب‌های هر فرد نسبت به کالاهای مختلف برای انسان هم می‌تواند چالش‌برانگیز و بعضا ناممکن باشد.

۴ـ۳ـ۲ پیش‌پردازش

با توجه به این توضیحات، نخستین گام در فرآیند پیش‌پردازش داده‌ها این است که ابتدا دو نمایش وان‌ـ هات، یکی از روی شناسه‌ی عددی کاربران و دیگری از روی شناسه‌ی عددی کالاها، به دست آوریم. سپس به ازای هر درایه از ماتریس امتیازات (خواه یکنواخت و خواه غیریکنواخت)، دو نمایش مذکور را برای کاربر و کالای متناظر با آن امتیاز به هم الحاق کرده و سپس حاصل را به بردار ویژگی‌های آن کاربر و آن کالا الحاق می‌کنیم. هر کدام از این بردارهای الحاقی یکی از سطرهای ماتریسی موسوم به ماتریس بافتار[۱۶] را تشکیل خواهند داد. در نهایت برچسب خروجی متناظر با هر یک از این سطرها که در واقع همان عدد امتیاز ثبت‌شده‌ی متناظر است را به انتهای هر سطر ضمیمه می‌کنیم و به این ترتیب ماتریسی با قالب مناسب از داده‌های خام اولیه به دست می‌آید.

برای پیش‌پردازش هر یک از دو مجموعه‌داده‌ی یکنواخت و غیریکنواخت، کافی است فایل خام مربوط به ماتریس هر کدام به عنوان ورودی اولیه مراحل بالا را طی کند. اما در مورد داده‌های مشاهده‌نشده، چون قصد داشتیم جفت‌های کاربرـ کالایی را به عنوان مشاهده‌نشده تلقی کنیم که در هیچ از یک دو مجموعه‌داده‌ی یکنواخت و غیریکنواخت به ازای آنها امتیازی ثبت نشده باشد، از این ایده استفاده کردیم که ماتریس‌های خام این فایل را با هم جمع کنیم و حاصل را به عنوان ورودی اولیه به تابع پیش‌پردازشگر بدهیم و البته معین کنیم که می‌خواهیم تابع را در حالت مشاهده‌نشده اجرا کنیم. با این توضیح که چون همه‌ی مقادیر هر یک از دو ماتریس داده‌ای خام نامنفی‌اند، وقتی آنها را درایه‌به‌درایه با هم جمع می‌کنیم، تنها درایه‌هایی از ماتریس حاصل صفر خواهند بود که در هر دو ماتریس صفر بوده‌اند و این یعنی دقیقا همان جفت‌هایی از کاربرـ کالا که در هر دوفایل امتیازی برایشان ثبت نشده است.

[۱۶] Context

پس از مرحله‌ی بالا، با توجه به یک آستانه‌ی از پیش‌تعیین‌شده، امتیازات خام که مقدار هر یک می‌توانست یکی از اعداد بازه‌ی ۱ تا ۵ بود، به صورت امتیازاتی دودویی ساده‌سازی می‌شوند؛ به این صورت که امتیازات بیشتر یا مساوی آستانه‌ی معین به عنوان امتیاز یک و امتیازات پایین‌تر از آن به عنوان امتیاز صفر در نظر گرفته می‌شوند.

پس از طی این مراحل، داده‌های خام در قالبی مناسب برای تغذیه به عنوان ورودی مدل‌ها در می‌آیند. اما باز هم این ماتریس‌های بزرگ به شکل مستقیم به مدل‌ها ورودی داده نمی‌شوند. بلکه ابتدا به سه بخش داده‌های آموزشی[۱۷]، ارزیابی[۱۸]، و آزمایشی[۱۹] تقسیم می‌شوند و هر دسته نیز به بسته[۲۰]های داده‌ای کوچکتری تقسیم می‌شود که در هر بسته تعداد از پیش‌تعیین‌شده‌ای نمونه وجود خواهد داشت. این بسته‌ها با نام‌گذاری مناسب و به تفکیک در قالب فایل‌های ماتریسی با پسوند npy ذخیره می‌شوند تا هنگام توسعه‌ی مدل‌ها، محتوای آنها خوانده و استفاده شود.

۴ـ۳ـ۳ مدل استفاده‌شده

برای یادگیری مسئله‌ی اصلی مورد بررسی ـ که همان پیش‌بینی بازخورد کاربران (مثبت یا منفی) در رویارویی با پیشنهاد محصولات مختلف است ـ از مدل شبکه‌های عصبی روبه‌جلو استفاده شده است. این ساختار با وجود اینکه از ساده‌ترین انواع شبکه‌های عصبی از نظر عملکرد و پیاده‌سازی است، به دلیل نتایج رضایت‌بخش آن همچنان در بسیاری از پژوهش‌های علمی حوزه‌ی یادگیری ماشین و علوم داده مورد استفاده قرار می‌گیرد. نتایج مطالعه‌ی حاضر نیز که در ادامه به آن پرداخته خواهد شد، مؤید این موضوع است.

بنابر اصول عمومی برنامه‌نویسی شیءگرا، مدل شبکه‌ی عصبی روبه‌جلوی مذکور را در قالب یک کلاس تعریف کرده‌ایم. برای نمونه‌سازی[۲۱] از این کلاس، متد سازنده[۲۲]ی آن سه ورودی *بعد داده‌های ورودی*، *لیستی شامل تعداد نورون‌ها در هر یک از لایه‌های پنهان شبکه*، و *تابع فعال‌ساز غیرخطی مورد استفاده در نورون‌های شبکه* را به عنوان پارامترهای موردنیاز برای تعیین معماری شبکه دریافت می‌کند. در نهایت هم خروجی شبکه یک تک‌مقدار عددی است که احتمال بازخورد مثبت کاربر به محصول را

[۱۷] Train data
[۱۸] Validation data
[۱۹] Test data
[۲۰] Batch
[۲۱] Instantiation
[۲۲] Constructor

به ازای هر نمونه از داده‌ها بیان می‌کند. طبعا هرچه این مقدار به صفر نزدیک‌تر باشد، یعنی پیش‌بینی مدل از بازخورد کاربر این است که محصول پیشنهادشده‌اش را نخواهد پسندید (بازخورد منفی) و هرچه خروجی شبکه به یک نزدیک‌تر باشد، بیانگر عکس این موضوع است (بازخورد مثبت).

۴ـ۳ـ۴ توابع هزینه و معیارهای ارزیابی

در فرآیند یادگیری بانظارت، برای آموزش دادن هر مدل روی داده‌های آموزشی نیاز به یک تابع هزینه داریم که معیاری از برازش مدل برای پیش‌بینی داده‌هاست. توابعی مانند آنتروپی متقاطع دودویی[۲۳] و خطای میانگین مربعی از نمونه‌های مشهور این نوع توابع‌اند که ما نیز در آموزش مدل‌هایمان از آنها استفاده کرده‌ایم. شرح این توابع در فصل ۲ آمده است.

از آنجا که هزینه‌ی لگاریتمی یک ***تابع هزینه*** و سطح زیر منحنی عملیاتی گیرنده یک ***معیار کارآیی*** است، حالت مطلوب در هر مسئله‌ی یادگیری، کمینه‌سازی اولی و بیشینه‌سازی دومی است. در این‌جا باید تفاوت‌های مهم دو معیار مذکور را با دقت بیشتری بررسی کنیم.

منحنی عملیاتی گیرنده

این منحنی نرخ اعلام درست برچسب مثبت برای موارد واقعا با برچسب مثبت[۲۴] را روی یک محور و نرخ اعلام اشتباه برچسب مثبت برای نمونه‌های واقعا با برچسب منفی[۲۵] را روی محوری دیگر نمایش می‌دهد. خود منحنی از به هم وصل کردن نقاطی به دست می‌آید که وضعیت دو نرخ مذکور را به ازای مقادیر مختلف آستانه[۲۶] برای تصمیم‌گیری مدل دسته‌بند دودویی نشان می‌دهند. لذا سطح زیر این منحنی معیاری از قدرت مدل در برقراری توازن در برچسب‌زنی درست نمونه‌هاست؛ چرا که معیارهای صرفا یک‌جانبه مانند دقت[۲۷] یا یادآوری[۲۸] می‌توانند از طریق انتساب تعدادی غیرمتعارف (بیش از اندازه زیاد یا کم) برچسب مثبت روی نمونه‌ها، به شکل مصنوعی و بدون بهبود واقعی کیفیت یادگیری مدل افزایش یابند اما سطح زیر منحنی عملیاتی گیرنده در واقع به طور همزمان به نسبت دسته‌بندی‌های درست در هر یک از دو نوع برچسب‌های مثبت و منفی توجه می‌کند.

[۲۳] Binary Cross Entropy (BCE)
[۲۴] True Positive Rate (TPR), recall, or sensitivity
[۲۵] False Positive Rate (FPR), or probability of false alarm
[۲۶] Threshold
[۲۷] Precision
[۲۸] Recall

البته یک نکته‌ی مهم این است که در مسائلی با توزیع نامتوازن برچسب‌های مثبت و منفی در میان داده‌ها (مثلا در شرایطی که فراوانی نمونه‌های مثبت بسیار کمتر است)، چون هر انتساب صحیح نمونه‌ها به هر یک از دو برچسب با اهمیت یکسان در نظر گرفته شده، تعداد فراوان نمونه‌های متعلق به یک دسته می‌تواند باعث شود که یک مدل حتی نه چندان تفکیک‌گر ــ که اغلب موارد همان دسته‌ی پرتعداد را به عنوان برچسب خروجی اعلام می‌کند ــ بتواند عدد بالایی از این معیار کسب کند؛ در حالی که واقعا تشخیص درست موارد با برچسب مثبت ــ اگرچه بسیار معدودند ــ برای ما اهمیت بیشتری نسبت به تشخیص موارد متعدد منفی داشته باشد. لذا یک نقطه‌ی ضعف عمومی معیار AUC در شرایط وجود دسته‌های نامتوازن در مسئله[۲۹] است.

هزینه‌ی لگاریتمی

در سوی دیگر، معیار هزینه‌ی لگاریتمی میزان **قطعیت**[۳۰] مدل را در مورد پیش‌بینی‌هایش ارزیابی می‌کند. به این معنا که این تابع هزینه با توجه به فرمول محاسبه‌ی آن ــ که در فصل ۲ نمونه‌ای از آن آورده شده است ــ در مواردی که مدل، با احتمال (به تعبیری اطمینان) خیلی بالا برچسب یک نمونه را اشتباه تشخیص می‌دهد، به مقدار بسیار زیادی رشد می‌کند (رشد تابع هزینه به معنای بدتر شدن عملکرد مدل است). لذا این معیار ارزیابی، به نوعی قدرت تعمیم‌پذیری مدل را نیز می‌سنجد؛ چرا که وقتی یک مدل نمونه‌های پیش روی خود را با قطعیت بالایی درست برچسب بزند، در مقایسه با مدلی که حتی ممکن است همان تعداد نمونه‌ی درست را پیش‌بینی کرده اما هر نمونه را با عدد احتمال شکننده‌تر و نامطمئن‌تری به دسته‌ی درست منتسب کرده، قابل اعتمادتر است و احتمالا در رویارویی با داده‌های جدید، عملکرد بهتری خواهد داشت.

۴ــ۳ــ۵ حلقه‌ی آموزش

حلقه‌ی آموزش[۳۱] یکی از مهم‌ترین بخش‌های کد اصلی نوشته‌شده برای اجرای آزمایش‌های مطلوب روی داده‌هاست. این بخش از کد در واقع شامل حلقه‌هایی تودرتوست که هر کدام از آنها مسئول پیمایش روی مجموعه‌ای از پیش‌تعیین‌شده از یکی از ابرپارامتر[۳۲]های برنامه هستند. لذا به این ترتیب در هر گام

[۲۹] Class imbalance problem
[۳۰] Certainty
[۳۱] Training loop
[۳۲] Hyperparameter

از اجرای درونی‌ترین حلقه در واقع با یک آرایش[۳۳] خاص از همه‌ی ابرپارامترهای موجود در برنامه ــ که به وضوح حداقل در مقدار یکی از ابرپارامترها با سایر آرایش‌ها متمایز است ــ روبرو هستیم. به ازای هر یک از این آرایش‌ها، آموزش دادن مدل‌ها به کمک داده‌های آموزشی و ارزیابی انجام می‌گیرد که در ادامه به تفصیل به جزئیات این فرآیند خواهیم پرداخت.

از نظر ساختاری و در یک دید سطح بالا، حلقه‌ی آموزش به دو بخش اصلی آموزش کلاسیک[۳۴] و آموزش چرخه‌ی بازخورد [۳۵] تقسیم می‌شود.

تفاوت مهم این دو نوع رویکرد آموزش این است که در نوع کلاسیک همه‌ی داده‌های آموزشی که قرار است مدل روی آنها آموزش ببیند، در قالب یک بارگیر داده[۳۶] واحد به مدل داده می‌شوند و مدل نیز به شیوه‌ی متداول و تک‌به‌تک روی هر یک از بسته‌های داده‌ای موجود در بارگیر داده آموزش می‌بیند. باقی داده‌های موجود هم مطابق معمول به دو بخش ارزیابی و آزمایشی تقسیم می‌شوند که برای تنظیم[۳۷] ابرپارامترها در خلال آموزش و سنجش نهایی عملکرد مدل استفاده خواهند شد.

اما در آموزش چرخه‌ی بازخورد، چنانکه از نام آن برمی‌آید، سعی داریم مفهوم چرخه‌ی بازخورد در یادگیری تقویتی را با نوع خاصی از فرآیند آموزش مدل‌های یادگیری، شبیه‌سازی کنیم. به این صورت که برای به دست آوردن بارگیر داده در این نوع آموزش، اصطلاحا ابتدا یک مجموعه‌داده‌ی پیشین داریم که البته با هر بار دریافت بارگیر داده مقدار آن به‌روز می‌شود.

در بدنه‌ی تابع تولید کننده‌ی بارگیر داده، ابتدا بهترین مدل آموزش‌دیده تاکنون، از محلی که قبلا در آن ذخیره‌شده بارگذاری می‌شود. سپس پیش‌بینی این مدل روی داده‌های پیشین به دست می‌آید. این نتایج ابتدا به ترتیب نزولی (بر اساس مقدار احتمال بازخورد مثبت کاربر به محصول) مرتب و سپس تعداد از پیش‌تعیین‌شده‌ای از ابتدای این لیست مرتب‌شده (این تعداد از پیش‌تعیین‌شده، خود یکی از ابرپارامترهای قابل تنظیم در مسئله است که در بخش‌های دیگر این متن ــ از جمله در فصل نتایج ــ از آن با عنوان «تعداد بالاترین نمونه‌های منتخب برای چرخه‌ی بازخورد» یاد شده است) انتخاب می‌شوند. در نهایت این لیست جایگزین همان داده‌های پیشین می‌شود که پیش‌تر راجع به آن صحبت کردیم.

در یک سامانه‌ی توصیه‌گر، یک چرخه‌ی بازخورد واقعی زمانی شکل می‌گیرد که پس از هر بار پیش‌بینی مدل یادگیری روی نمونه‌داده‌ها، محصولاتی با بیشترین امتیاز از این پیش‌بینی به کاربران پیشنهاد شوند؛

[۳۳] Setting
[۳۴] Warm start training
[۳۵] Feedback loop training
[۳۶] Dataloader
[۳۷] Tune

سپس هر یک از کاربران به محصولات پیشنهاد شده به خودشان امتیازی می‌دهند و این امتیازهای جمع‌آوری‌شده به عنوان اطلاعاتی جدید در اختیار مدل‌ها قرار می‌گیرد تا بر اساس آن پیش‌بینی‌های بعدی خود را اصلاح و به بازخوردهای واقعی کاربران نسبت به محصولات پیشنهادی نزدیک‌تر کنند.

در هر یک از دو نوع ساختار آموزشی مذکور مدل‌های مربوط به هر کدام از راهبردهای پل و تصفیه به صورت جداگانه توسعه داده می‌شوند که در ادامه هر یک را بررسی خواهیم کرد.

فرآیند یادگیری در راهبرد پل

همان‌طور که در ۴ـ۱ـ۱ دیدیم، ایده‌ی راهبرد پل شامل آموزش همزمان دو مدل یکنواخت و غیریکنواخت می‌شود که در تابع آموزش توسعه‌داده‌شده نیز همین موضوع به چشم می‌خورد؛ یعنی دو شبکه‌ی عصبی متناظر با مدل‌های یکنواخت و غیریکنواخت، همزمان و طبق تابع هزینه‌ی راهبرد پل روی بسته‌های داده‌ای با پیش‌قدر و بدون پیش‌قدر آموزش می‌بینند. در این فرآیند، مطابق رویکرد رایج در یادگیری بانظارت، پس از آموزش مدل روی نمونه‌داده‌های آموزشی، عملکرد مدل روی نمونه‌داده‌های ارزیابی سنجیده می‌شود و اگر نتایج مدل روی این دسته از داده‌ها بهتر از بهترین مدل ذخیره‌شده تا آن زمان باشد (با معیاری مانند سطح زیر منحنی عملیاتی گیرنده[۳۸])، مدل جدید جایگزین آخرین مدل ذخیره‌شده‌ی قبلی می‌شود.

یک نکته‌ی قابل ذکر در بحث یادگیری این راهبرد، نیاز به یک نوع بارگیر داده‌ی خاص، مطابق با تابع هزینه‌ی این راهبرد و آموزش همزمان دو مدل یکنواخت و غیریکنواخت طی آن، است که آن را پیاده‌سازی کردیم.

در بارگیر داده‌های رایج و آماده در کتابخانه‌های یادگیری ماشین، به جای آنکه اندازه‌ی هر یک از بسته‌های داده به عنوان یک پارامتر ورودی داده شود و بر اساس آن، مجموعه‌داده‌ی خام به تعدادی بسته با اندازه‌ی یکسان تقسیم می‌شود (مگر احیانا آخرین بسته، آن هم به دلیل بخش‌پذیر نبودن تعداد نمونه‌ها در دادگان بر طول هر بسته).

اما در بارگیر داده‌ی مورد استفاده‌ی ما برای راهبرد پل، هدف مساوی بودن **تعداد بسته‌ها**یی است که از هر یک از مجموعه‌داده‌های با پیش‌قدر، بدون پیش‌قدر، و مشاهده‌نشده می‌آیند و نه تعداد نمونه‌های موجود در هر بسته. چرا که در این راهبرد می‌خواهیم در هر گام از بهینه‌سازی تابع هزینه‌ی مدل، یک

[۳۸] Area Under Curve (AUC) of Receiver Operating Characteristic (ROC)

بسته داده از هر یک از انواع سه‌گانه‌ی مذکور بگیریم و با استفاده از این سه بسته داده، یک مرحله از آموزش مدل را پیش ببریم. لذا واضح است که نباید در هیچ مرحله‌ای بسته‌های داده‌ای آماده‌شده از هیچ یک از این سه نوع زودتر از انواع دیگر تمام شوند.

البته واضح است که چون تعداد کل نمونه‌های موجود در مجموعه‌داده‌های این سه نوع داده‌ی مذکور با هم برابر نیست (از جمله به دلیل پرهزینه‌بودن جمع‌آوری داده‌های بدون پیش‌قدر طبق بحث‌های پیشین)، برابری تعداد بسته‌ها در بارگیر داده، باعث نامساوی شدن تعداد اعضای هر بسته در انواع داده‌ای مختلف می‌شود اما این موضوع دغدغه‌ی ما در این راهبرد نیست.

فرآیند یادگیری در راهبرد تصفیه

چنان که پیش‌تر در ۴ـ۱ـ۲ از آن صحبت شد، راهبرد تصفیه خود به دو بخش پیش‌آموزش شبکه‌ی یکنواخت و آموزش شبکه‌ی غیریکنواخت تقسیم می‌شود. ابتدا مدل نخست ـ که یکنواخت نامیده می‌شود ـ روی داده‌های بدون پیش‌قدر آموزش می‌بیند و به کمک مجموعه‌داده‌های ارزیابی بهترین وضعیت این مدل طی آموزش ذخیره می‌گردد.

سپس بهترین مدل یکنواخت ذخیره‌شده طی فرآیند پیش‌آموزش به همراه داده‌های با پیش‌قدر و ابرپارامترهای مربوطه، به عنوان ورودی‌های فرآیند آموزش شبکه‌ی بعدی استفاده می‌شوند که این شبکه‌ی اخیر را غیریکنواخت می‌نامیم. در فرآیند آموزش شبکه‌ی غیریکنواخت، به ازای هر نمونه‌ی با پیش‌قدر، ضریبی از پیش‌بینی نرمال‌سازی‌شده‌ی مدل یکنواخت پیش‌آموزش‌دیده با پیش‌بینی مدل غیریکنواخت روی همان نمونه ترکیب می‌شود. بر این اساس مدل غیریکنواخت، مطابق همان تابع هزینه‌ی مذکور در ۴ـ۱ـ۲ آموزش می‌بیند و بهترینِ آن از نظر عملکرد روی نمونه‌های ارزیابی ذخیره می‌گردد.

فصل ۵

نتایج

در این فصل نتایج عددی و نمودارهای این پروژه که مربوط به آزمایش‌های انجام‌شده روی دادگان‌های Coat و Yahoo!R۳ هستند، ارائه خواهند شد.

یک نکته‌ی کلی لازم به ذکر این است که در تمام آزمایش‌های مربوط به جستجوی ابرپارامترها در این فصل، بهترین مقدار حاصل از هر مجموعه‌ی مورد جستجو به کمک سنجش عملکرد مدل روی داده‌های ارزیابی به ازای هر یک از مقادیر آن مجموعه استخراج شده است.

۵ـ۱ نتایج روی دادگان Coat

۵ـ۱ـ۱ آزمایش‌های ترکیب کلاسیک و چرخه‌ی بازخورد

در این بخش دو آزمایش مرتبط به هم را انجام داده‌ایم که در آزمایش اول ابرپارامترهای مختص راهبردهای پل و تصفیه را صرفا با روش آموزش کلاسیک تنظیم می‌کنیم و در آزمایش دوم با ثابت نگه داشتن بهترین آرایش یافت‌شده از ابرپارامترهای مذکور، ابرپارامتر تعداد بالاترین نمونه‌های منتخب برای چرخه‌ی بازخورد را با ترکیب هر دو نوع آموزش کلاسیک و چرخه‌ی بازخورد تنظیم می‌کنیم.

در آزمایش نخست، مجموعه‌ی مقادیر {۰/۰۰۰۱, ۰/۰۰۱} برای هر یک از ابرپارامترهای λ_t و λ_c در راهبرد پل و λ_t در مدل پیش‌آموزش‌دیده و λ_c در آموزش مدل اصلی راهبرد تصفیه، و مجموعه‌ی مقادیر {۰/۵, ۰/۷, ۰/۹} برای ابرپارامتر α، مورد جستجو قرار گرفتند. ابرپارامتر نسبت داده‌های مشاهده‌نشده

به داده‌های مشاهده‌شده در فرآیند آموزش مقدار ثابت ۱ بوده است. بهترین مقدار یافت‌شده برای ابرپارامترهای هر یک از مدل‌ها از قرار زیر است:

راهبرد پل)

$$\lambda_t = ۰/۰۰۱, \lambda_c = ۰/۰۰۰۱ \quad (۵-۱)$$

راهبرد تصفیه)

$$\lambda_t = ۰/۰۰۱ \text{ (مدل پیش‌آموزش‌دیده)} , \lambda_c = ۰/۰۰۱, \alpha = ۰/۹ \text{ (مدل اصلی)} \quad (۵-۲)$$

جدول ۵ـ۱ حاوی نتایج داده‌های آزمایشی برای مدل‌هایی است که با همین مقادیرِ بهترین آموزش دیده‌اند.

جدول ۵ـ۱: نتایج مدل‌ها روی داده‌های آزمایشی دادگان Coat با استفاده از ۱۰٪ داده‌های یکنواخت در آموزش و آستانه‌ی ۴ برای دودویی‌کردن بازخوردها.

نام مدل	AUC	هزینه‌ی لگاریتمی
شبکه‌ی یکنواخت پل	۰/۶۹۱۳	۰/۱۷۹۲
شبکه‌ی غیریکنواخت پل	۰/۸۰۵۵	۰/۱۶۸۲
شبکه‌ی تصفیه (مدل غیریکنواخت نهایی)	۰/۷۹۹۰	۰/۵۸۶۸

در آزمایش دوم که فرآیند آموزش در آن شامل هر دو نوع کلاسیک و چرخه‌ی بازخورد می‌شود، مجموعه‌ی مقادیر {۹۲, ۱۸۴, ۳۰۶, ۴۶۰, ۹۲۰, ۱۳۹۲} برای ابرپارامتر تعداد بالاترین نمونه‌های منتخب برای چرخه‌ی بازخورد مورد جستجو قرار گرفتند که جزئیات نتایج مدل‌ها برای این آزمایش در جدول ۵ـ۲ آمده است.

جدول ۵ـ۲: نتایج مدل‌ها روی داده‌های آزمایشی دادگان Coat با استفاده از ۱۰٪ داده‌های یکنواخت در آموزش و آستانه‌ی ۴ برای دودویی‌کردن بازخوردها، به قصد تنظیم تعداد بالاترین نمونه‌های منتخب در چرخه‌ی بازخورد.

		تعداد بالاترین نمونه‌های منتخب در چرخه‌ی بازخورد					
		۹۲	۱۸۴	۳۰۶	۴۶۰	۹۲۰	۱۳۹۲
معیار ارزیابی مدل	AUC شبکه‌ی یکنواخت پل	۰/۶۶۶۹	۰/۶۸۳۷	۰/۶۷۷۴	۰/۶۷۷۶	۰/۶۸۵۴	۰/۶۸۷۷
	AUC شبکه‌ی غیریکنواخت پل	۰/۵۹۵۴	۰/۶۴۶۱	۰/۶۷۲۵	۰/۶۸۹۹	۰/۷۰۴۱	۰/۶۷۷۰
	AUC شبکه‌ی تصفیه (مدل غیریکنواخت نهایی)	۰/۷۰۵۹	۰/۶۸۶۵	۰/۶۹۳۸	۰/۶۸۲۴	۰/۷۰۶۰	۰/۷۰۶۶
	هزینه‌ی لگاریتمی شبکه‌ی یکنواخت پل	۰/۱۹۱۵	۰/۱۷۴۹	۰/۱۸۳۲	۰/۲۰۰۰	۰/۲۰۳۶	۰/۱۹۰۲
	هزینه‌ی لگاریتمی شبکه‌ی غیریکنواخت پل	۰/۱۹۷۷	۰/۲۰۰۴	۰/۱۸۳۹	۰/۱۷۹۱	۰/۱۹۲۳	۰/۱۸۷۰
	هزینه‌ی لگاریتمی شبکه‌ی تصفیه (مدل غیریکنواخت نهایی)	۰/۳۰۸۲	۰/۲۹۵۰	۰/۲۸۳۴	۰/۳۱۵۹	۰/۴۰۲۸	۰/۴۷۶۶

۵ـ۱ـ۲ آزمایش‌های صرفا کلاسیک

نتایج سه آزمایش این بخش در جدول‌های ۵ـ۳، ۵ـ۴، و ۵ـ۵ آمده و همگی صرفا مربوط به نوع آموزش کلاسیک هستند. مجموعه‌ی مقادیر $\{۰/۰۰۰۱, ۰/۰۰۱\}$ برای هر یک از ابرپارامترهای λ_t و λ_c در راهبرد پل و λ_t در مدل پیش‌آموزش‌دیده و λ_c در آموزش مدل اصلی راهبرد تصفیه، و مجموعه‌ی مقادیر $\{۰/۵, ۰/۷, ۰/۹\}$ برای ابرپارامتر α، مورد جستجو قرار گرفتند. ابرپارامتر نسبت داده‌های مشاهده‌نشده به داده‌های مشاهده‌شده در فرآیند آموزش نیز مقدار ثابت ۱ بوده است.

در پیش‌پردازش نخست داده‌ها، بهترین مقدار یافت‌شده برای ابرپارامترهای هر یک از مدل‌ها از قرار زیر است:

راهبرد پل)

$$\lambda_t = ۰/۰۰۱, \lambda_c = ۰/۰۰۰۱ \quad (۵ـ۳)$$

راهبرد تصفیه)

$$\lambda_t = ۰/۰۰۱ \text{ (مدل پیش‌آموزش‌دیده)} , \lambda_c = ۰/۰۰۰۱, \alpha = ۰/۷ \text{ (مدل اصلی)} \quad (۵ـ۴)$$

جدول ۵ـ۳ حاوی نتایج داده‌های آزمایشی برای مدل‌هایی است که با همین مقادیرِ بهترین آموزش دیده‌اند.

جدول ۵ـ۳: نتایج مدل‌ها روی داده‌های آزمایشی دادگان Coat با استفاده از ۱۰٪ داده‌های یکنواخت در آموزش و آستانه‌ی ۳ برای دودویی‌کردن بازخوردها.

نام مدل	AUC	هزینه‌ی لگاریتمی
شبکه‌ی یکنواخت پل	۰/۶۷۳۶	۰/۴۶۰۲
شبکه‌ی غیریکنواخت پل	۰/۷۶۵۷	۰/۴۰۹۹
شبکه‌ی تصفیه (مدل غیریکنواخت نهایی)	۰/۷۶۸۶	۰/۵۴۳۳

در پیش‌پردازش دوم، بهترین ابرپارامترهای یافت‌شده از قرار زیر و جدول ۵ـ۴ حاوی نتایج داده‌های آزمایشی است.

راهبرد پل)

$$\lambda_t = ۰/۰۰۱, \lambda_c = ۰/۰۰۱ \quad (۵\_۵)$$

راهبرد تصفیه)

$$\lambda_t = ۰/۰۰۱ \text{ (مدل پیش‌آموزش‌دیده)} \ , \lambda_c = ۰/۰۰۱, \alpha = ۰/۷ \text{ (مدل اصلی)} \quad (۶\_۵)$$

جدول ۵ـ۴: نتایج مدل‌ها روی داده‌های آزمایشی دادگان Coat با استفاده از ۶۰٪ داده‌های یکنواخت در آموزش و آستانه‌ی ۳ برای دودویی‌کردن بازخوردها.

نام مدل	AUC	هزینه‌ی لگاریتمی
شبکه‌ی یکنواخت پل	۰/۶۸۹۲	۰/۴۸۳۲
شبکه‌ی غیریکنواخت پل	۰/۷۱۹۸	۰/۴۶۸۲
شبکه‌ی تصفیه (مدل غیریکنواخت نهایی)	۰/۷۲۵۳	۰/۶۳۳۰

در پیش‌پردازش سوم، بهترین ابرپارامترهای یافت‌شده از قرار زیر و جدول ۵ـ۵ حاوی نتایج داده‌های آزمایشی است.

راهبرد پل)

$$\lambda_t = ۰/۰۰۱, \lambda_c = ۰/۰۰۰۱ \quad (۷\_۵)$$

راهبرد تصفیه)

$$\lambda_t = ۰/۰۰۱ \text{ (مدل پیش‌آموزش‌دیده)} \ , \lambda_c = ۰/۰۰۰۱, \alpha = ۰/۹ \text{ (مدل اصلی)} \quad (۸\_۵)$$

جدول ۵ـ۵: نتایج مدل‌ها روی داده‌های آزمایشی دادگان Coat با استفاده از ۶۰٪ داده‌های یکنواخت در آموزش و آستانه‌ی ۴ برای دودویی‌کردن بازخوردها.

نام مدل	AUC	هزینه‌ی لگاریتمی
شبکه‌ی یکنواخت پل	۰/۶۵۲۹	۰/۱۸۲۲
شبکه‌ی غیریکنواخت پل	۰/۷۰۵۸	۰/۱۸۶۹
شبکه‌ی تصفیه (مدل غیریکنواخت نهایی)	۰/۷۲۲۷	۰/۴۲۱۱

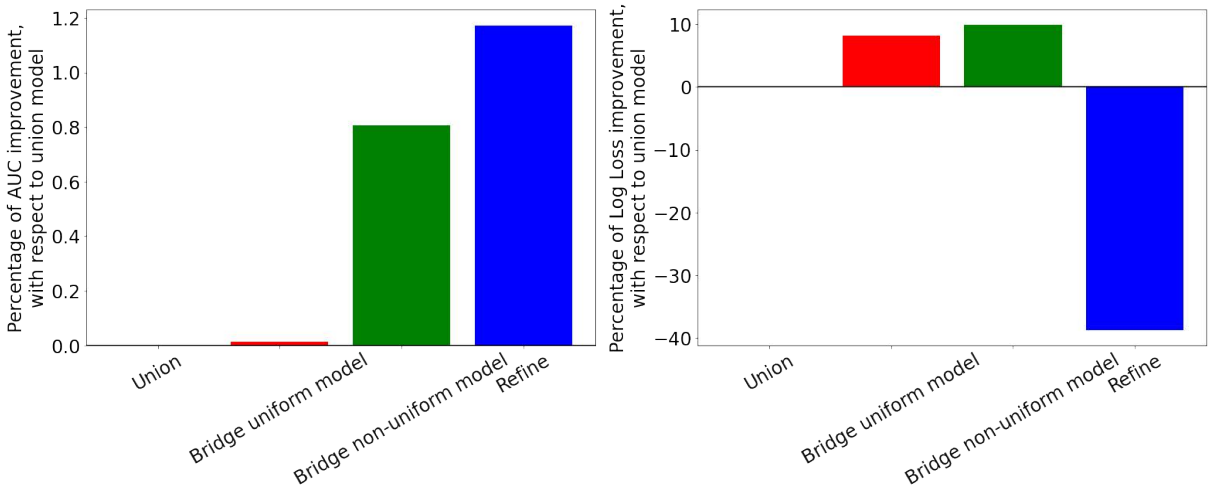
۵ـ۱ـ۳ نتایج نهایی

در جدول ۵ـ۶ نتایج آخرین آزمایش روی دادگان Coat با وسیع‌ترین بازه‌ی مورد جستجو برای ابرپارامترهای هدف ذکر شده است که در آن هر یک از شبکه‌های سه‌گانه‌ی مورد بررسی در نتایج قبلی، با یک مدل مرجع[۱] مقایسه شده‌اند. این مدل مرجع در واقع یک شبکه‌ی عصبی با یک تابع هزینه‌ی کاملا ساده (به طور خاص آنتروپی متقاطع دودویی) است که صرفا به صورت یکجا روی اجتماع داده‌های یکنواخت و غیریکنواخت آموزش می‌بیند و هیچ ایده‌ی خاصی در پیاده‌سازی آن به کار گرفته نشده است.

جدول ۵ـ۶: درصد بهبود مدل‌ها نسبت به مدل اجتماع روی داده‌های آزمایشی دادگان Coat با استفاده از ۶۰٪ داده‌های یکنواخت در آموزش و آستانه‌ی ۴ برای دودویی‌کردن بازخوردها (دقت شود که بهبود loss log در اثر کاهش آن و بهبود AUC در اثر افزایش آن رخ می‌دهد و درصد بهبود برای هر معیار با لحاظ کردن این موضوع گزارش شده است).

نام مدل	AUC	هزینه‌ی لگاریتمی
اجتماع (تغذیه‌ی همزمان مدل با داده‌های یکنواخت و غیریکنواخت)	٪۰/۰	٪۰/۰
شبکه‌ی یکنواخت پل	٪۰/۰۱۵	٪۸/۱۷۴
شبکه‌ی غیریکنواخت پل	٪۰/۸۰۷	٪۹/۹۵۲
شبکه‌ی تصفیه (مدل غیریکنواخت نهایی)	٪۱/۱۷۲	ـ٪۳۸/۷۶۷

[۱] Baseline



شکل ۵ـ۱: مقایسه‌ی مدل‌ها روی داده‌های آزمایشی دادگان Coat با استفاده از ۶۰٪ داده‌های یکنواخت در آموزش و آستانه‌ی ۴ برای دودویی‌کردن بازخوردها.

۵ـ۲ نتایج روی دادگان Yahoo!R۳

۵ـ۲ـ۱ آزمایش‌های صرفا کلاسیک

در چهار آزمایش این بخش که نتایج آن در جدول‌های ۵ـ۷، ۵ـ۸، ۵ـ۹، و ۵ـ۱۰ آمده و همگی صرفا مربوط به نوع آموزش کلاسیک هستند، مجموعه‌ی مقادیر $\{۰/۰۰۰۱, ۰/۰۱\}$ برای هر یک از ابرپارامترهای λ_t و λ_c در راهبرد پل و λ_t در مدل پیش‌آموزش‌دیده و λ_c در آموزش مدل اصلی راهبرد تصفیه، مجموعه‌ی مقادیر $\{۰/۵, ۰/۷, ۰/۹\}$ برای ابرپارامتر α ، و مجموعه‌ی مقادیر $\{۱/۱, ۲, ۳\}$ برای ابرپارامتر نسبت داده‌های مشاهده‌نشده به داده‌های مشاهده‌شده در فرآیند آموزش، مورد جستجو قرار گرفتند.

در پیش‌پردازش نخست داده‌ها، بهترین مقدار یافت‌شده برای ابرپارامترهای هر یک از مدل‌ها از قرار زیر است:

راهبرد پل)

$$\text{نسبت داده‌های مشاهده‌نشده به مشاهده‌شده} = ۱/۱, \lambda_c = ۰/۰۰۰۱, \lambda_t = ۰/۰۰۰۱ \quad (۵ـ۹)$$

راهبرد تصفیه)

$$\lambda_t = ۰/۰۰۰۱ \text{ (مدل پیش‌آموزش‌دیده)} \quad , \lambda_c = ۰/۰۰۰۱, \alpha = ۰/۵ \text{ (مدل اصلی)} \qquad (۵-۱۰)$$

جدول ۵ـ۷ حاوی نتایج داده‌های آزمایشی برای مدل‌هایی است که با همین مقادیرِ بهترین آموزش دیده‌اند.

جدول ۵ـ۷: نتایج مدل‌ها روی داده‌های آزمایشی دادگان Yahoo!R۳ با استفاده از ۱۰٪ داده‌های یکنواخت در آموزش و آستانه‌ی ۳ برای دودویی‌کردن بازخوردها.

نام مدل	AUC	هزینه‌ی لگاریتمی
شبکه‌ی یکنواخت پل	۰/۵۶۶۹	۰/۳۰۳۹
شبکه‌ی غیریکنواخت پل	۰/۶۰۷۹	۰/۲۹۴۹
شبکه‌ی تصفیه (مدل غیریکنواخت نهایی)	۰/۷۲۶۰	۰/۶۲۰۶

در پیش‌پردازش دوم، بهترین ابرپارامترهای یافت‌شده از قرار زیر و جدول ۵ـ۸ حاوی نتایج داده‌های آزمایشی است.

راهبرد پل)

$$\lambda_t = ۰/۰۰۰۱, \lambda_c = ۰/۰۰۰۱, \text{نسبت داده‌های مشاهده‌نشده به مشاهده‌شده} = ۱/۱ \qquad (۵-۱۱)$$

راهبرد تصفیه)

$$\lambda_t = ۰/۰۰۰۱ \text{ (مدل پیش‌آموزش‌دیده)} \quad , \lambda_c = ۰/۰۰۰۱, \alpha = ۰/۹ \text{ (مدل اصلی)} \qquad (۵-۱۲)$$

جدول ۵ـ۸: نتایج مدل‌ها روی داده‌های آزمایشی دادگان Yahoo!R۳ با استفاده از ۶۰٪ داده‌های یکنواخت در آموزش و آستانه‌ی ۳ برای دودویی‌کردن بازخوردها.

نام مدل	AUC	هزینه‌ی لگاریتمی
شبکه‌ی یکنواخت پل	۰/۶۱۴۲	۰/۳۰۰۸
شبکه‌ی غیریکنواخت پل	۰/۶۰۶۸	۰/۳۰۲۹
شبکه‌ی تصفیه (مدل غیریکنواخت نهایی)	۰/۷۵۳۹	۰/۷۵۱۶

در پیش‌پردازش سوم، بهترین ابرپارامترهای یافت‌شده از قرار زیر و جدول ۵ـ۹ حاوی نتایج داده‌های آزمایشی است.

راهبرد پل)

$$\lambda_t = ۰/۰۰۰۱, \lambda_c = ۰/۰۱, \text{نسبت داده‌های مشاهده‌نشده به مشاهده‌شده} = ۱/۱ \qquad (۵ـ۱۳)$$

راهبرد تصفیه)

$$\lambda_t = ۰/۰۰۰۱ \text{ (مدل پیش‌آموزش‌دیده)} \ , \lambda_c = ۰/۰۰۰۱, \alpha = ۰/۵ \text{ (مدل اصلی)} \qquad (۵ـ۱۴)$$

جدول ۵ـ۹: نتایج مدل‌ها روی داده‌های آزمایشی دادگان Yahoo!R۳ با استفاده از ۱۰٪ داده‌های یکنواخت در آموزش و آستانه‌ی ۴ برای دودویی‌کردن بازخوردها.

نام مدل	AUC	هزینه‌ی لگاریتمی
شبکه‌ی یکنواخت پل	۰/۵۷۲۸	۰/۱۲۳۴
شبکه‌ی غیریکنواخت پل	۰/۶۱۶۵	۰/۱۴۰۴
شبکه‌ی تصفیه (مدل غیریکنواخت نهایی)	۰/۷۷۳۲	۰/۴۶۳۷

در پیش‌پردازش چهارم، بهترین ابرپارامترهای یافت‌شده از قرار زیر و جدول ۵ـ۱۰ حاوی نتایج داده‌های آزمایشی است.

راهبرد پل)

$$\lambda_t = ۰/۰۰۰۱, \lambda_c = ۰/۰۱, \text{نسبت داده‌های مشاهده‌نشده به مشاهده‌شده} = ۱/۱ \qquad (۵ـ۱۵)$$

راهبرد تصفیه)

(۱۶ـ۵) (مدل اصلی) $\alpha = ۰/۵, \lambda_c = ۰/۰۰۰۱$, $\lambda_t = ۰/۰۰۰۱$ (مدل پیش‌آموزش‌دیده)

جدول ۵ـ۱۰: نتایج مدل‌ها روی داده‌های آزمایشی دادگان Yahoo!R۳ با استفاده از ۶۰٪ داده‌های یکنواخت در آموزش و آستانه‌ی ۴ برای دودویی‌کردن بازخوردها.

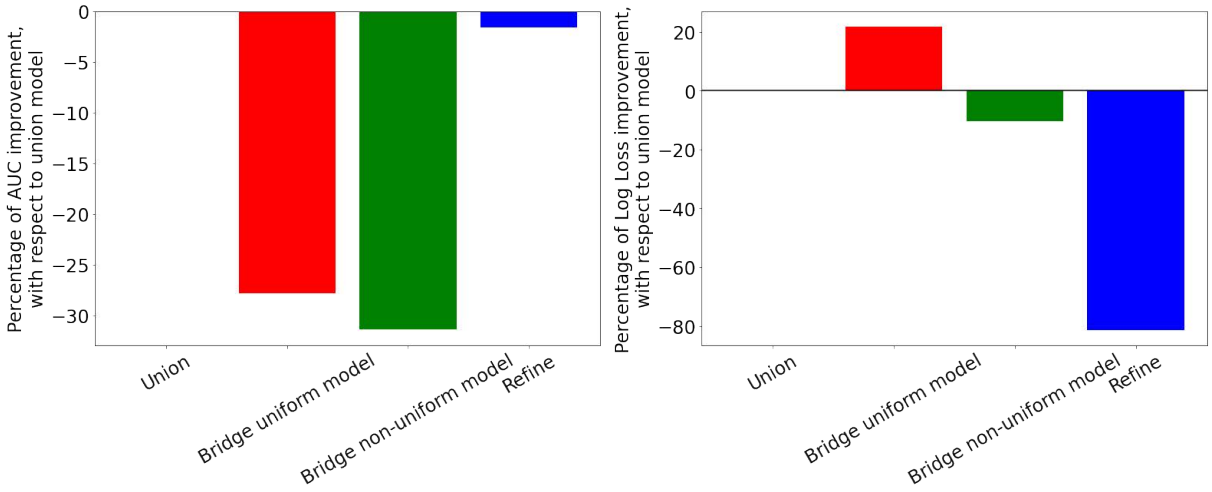
نام مدل	AUC	هزینه‌ی لگاریتمی
شبکه‌ی یکنواخت پل	۰/۶۵۱۳	۰/۱۲۱۰
شبکه‌ی غیریکنواخت پل	۰/۵۹۸۳	۰/۱۶۰۷
شبکه‌ی تصفیه (مدل غیریکنواخت نهایی)	۰/۷۹۲۱	۰/۴۹۷۵

۵ـ۲ـ۲ نتایج نهایی

در جدول ۵ـ۱۱ نتایج آخرین آزمایش انجام‌شده روی دادگان Yahoo!R۳ آمده است که مشابه توضیحات مذکور در بخش ۵ـ۱ـ۳ برای این بخش نیز صادق است.

جدول ۵ـ۱۱: درصد بهبود مدل‌ها نسبت به مدل اجتماع روی داده‌های آزمایشی دادگان Yahoo!R۳ با استفاده از ۶۰٪ داده‌های یکنواخت در آموزش و آستانه‌ی ۴ برای دودویی‌کردن بازخوردها.

نام مدل	AUC	هزینه‌ی لگاریتمی
اجتماع (تغذیه‌ی همزمان مدل با داده‌های یکنواخت و غیریکنواخت)	٪۰/۰	٪۰/۰
شبکه‌ی یکنواخت پل	ـ٪۲۷/۸۱۲	٪۲۱/۸۴۹
شبکه‌ی غیریکنواخت پل	ـ٪۳۱/۳۸۵	ـ٪۱۰/۳۱۷
شبکه‌ی تصفیه (مدل غیریکنواخت نهایی)	ـ٪۱/۵۹۴	ـ٪۸۱/۴۰۹



شکل ۵ـ۲: مقایسه‌ی مدل‌ها روی داده‌های آزمایشی دادگان Yahoo!R۳ با استفاده از ۶۰٪ داده‌های یکنواخت در آموزش و آستانه‌ی ۴ برای دودویی‌کردن بازخوردها.

۵ـ۳ تحلیل نتایج آزمایش‌ها بر اساس معیارهای ارزیابی

همان‌طور که از محتوای جدول‌های این فصل می‌توان دریافت، معیارهای ارزیابی اصلی مورد استفاده در آزمایش‌ها یکی AUC و دیگری هزینه‌ی لگاریتمی است. با دقت در کلیت نتایج آزمایش‌ها می‌توان دید که مدل‌های توسعه‌یافته بر اساس راهبرد تصفیه در اکثریت قریب به اتفاق آزمایش‌ها عملکرد بهتری از نظر معیار اول، یعنی سطح زیر منحنی عملیاتی گیرنده، داشته‌اند. اما در مقابل، معمولا مقدار هزینه‌ی لگاریتمی برای مدل‌های تصفیه به میزان قابل توجهی بیشتر از مدل‌های پل در آزمایش‌های یکسان بوده است.

۵ـ۳ـ۱ مقایسه‌ی عملکرد مدل‌ها

در مسئله‌ی مورد مطالعه‌ی ما مشکل نامتوازن بودن برچسب‌های ممکن برای داده‌ها به میزان قابل توجهی وجود دارد؛ زیرا به طور کلی حجم نظرات حاکی از رضایت زیاد (مثلا نمره‌ی ۴ یا ۵ از مقایس ۵ که آنها را به برچسب مثبت تعبیر می‌کنیم) کاربران نسبت به کالاهای پیشنهادی به آنها بسیار کمتر از حجم نظرات کم‌رضایت (مثلا نمره‌ی ۱ تا ۳ از مقیاس ۵ که آنها را به برچسب منفی تعبیر می‌کنیم) است.

لذا با توجه به تفاوت دو معیار ارزیابی AUC و هزینه‌ی لگاریتمی که در بخش ۴ـ۳ـ۴ مفصلاً به

آن پرداخته شد، و نیز با توجه به اینکه برتری مدل‌های راهبرد تصفیه در AUC نسبت به مدل‌های راهبرد پل خفیف است اما برتری مدل‌های پل در هزینه‌ی لگاریتمی نسبت به مدل‌های تصفیه در اغلب موارد چشمگیر است، می‌توان گفت که راهبرد پل در برآیند دو معیار، عملکرد قابل قبول‌تری در آزمایش‌ها نشان داده است.

فصل ۶

جمع‌بندی

در این فصل ضمن ارائه‌ی یک جمع‌بندی کلی از روند پیشبرد این پروژه، به راهبردهای ممکن دیگر برای حرکت در راستای این مسئله نیز اشاره می‌کنیم.

در این مطالعه دو راهبرد به نام‌های پل و تصفیه را که در [۱] معرفی شده‌اند و از روش‌های مبتنی بر برچسب برای حل مسئله‌ی پیش‌قدر مدل پیشین‌اند، بررسی کردیم. نحوه‌ی پیاده‌سازی اصلی این راهبردها در مقاله‌ی مذکور مبتنی بر روش‌های تجزیه‌ی ماتریسی با رتبه‌ی پایین و استفاده از شبکه‌های عصبی خودکدگذار بود. بخش عمده‌ای از فعالیت ما در این پروژه به مطالعه‌ی دقیق و جزئی [۱] و [۱۸]، و سپس ارائه‌ی یک پیاده‌سازی جدید برای دو راهبرد پل و تصفیه با استفاده از شبکه‌های عصبی روبه‌جلو اختصاص داشته است.

۶ـ۱ مطالعه‌ی منابع و پیاده‌سازی موجود از دو راهبرد هدف

همان‌طور که پیش‌تر به آن اشاره شد، ما اجرای این پروژه را با مطالعه‌ی منابع مرتبط به مسئله‌ی پیش‌قدر در داده‌های جمع‌آوری‌شده برای سامانه‌های توصیه‌گر آغاز کردیم. سپس به طور دقیق‌تر روی روش‌های پیشنهاد شده در [۱] متمرکز شدیم و باز از بین چند دسته روش مختلف موجود در این مقاله، به سراغ بخش روش‌های مبتنی بر برچسب رفتیم. در این روش‌ها، ایده‌ی کلی این است که یک مدل روی حجم کمی داده که بدون پیش‌قدر جمع‌آوری شده آموزش می‌بیند (موسوم به مدل یکنواخت). این در حالی است که مدل دیگر روی حجم بسیار بیشتری از داده‌های دارای پیش‌قدر آموزش می‌بیند (مدل

غیریکنواخت). حال یا مدل یکنواخت پیش ازمدل غیریکنواخت و یا موازی با آن (با استفاده از یک تابع هدف توأم) آموزش می‌بیند. در دو حالت، هدف این است که مدل غیریکنواخت با استفاده از برچسب‌های پیش‌بینی‌شده‌ی مدل یکنواخت، به اصلاح پیش‌قدر موجود در نتایج خود (ناشی از پیش‌قدر داده‌های در اختیارش) بپردازد. به این ترتیب با کمک حجم اندکی از داده‌های یکنواخت که جمع‌آوری‌شان پرهزینه است (به دلیل لطمه به رضایت کاربران)، عملکرد مدل اصلی که مجبور است روی حجم زیادی داده‌ی با پیش‌قدر آموزش ببیند، بهبود داده می‌شود.

در این مرحله پیاده‌سازی اصلی‌ای را که نویسندگان [۱] برای راهبردهای پیشنهادی‌شان ارائه کرده بودند، به دقت مطالعه کردیم تا با کسب دید کافی از آن، به توسعه‌ی پیاده‌سازی متناسب با نیاز خودمان بپردازیم.

۶ـ۲ پیش‌پردازش داده‌ها

پس از مطالعات مقدماتی مذکور، دو دادگان Coat و Yahoo!R۳ را که به ترتیب مربوط به نظرات کاربران در مورد پوشاک و موسیقی هستند، برای انجام آزمایش‌ها انتخاب نمودیم. این دو دادگان در مقالات پیشین موضوع نیز مورد استفاده‌ی دیگران قرار گرفته بودند؛ از جمله در خود [۱] که دادگان عمومی (جدا از دادگان تجاری منتشر نشده) آن همین Yahoo!R۳ بوده است.

پس از بررسی ساختار خام هر یک از این دو دادگان، پیش‌پردازش‌های متناسب برای هر کدام را انجام دادیم. برای مثال بازخورد (همان برچسب خروجی هر نمونه) های موجود در هر دادگان را که از مقیاس ۱ تا ۵ بودند، بر اساس یک آستانه‌ی معین مثل ۴ یا ۵ به صورت دودویی (مثبت یا منفی) درآوردیم که یعنی هر بازخورد با مقدار امتیاز بیشتر از آستانه، بازخورد مثبت و سایر بازخوردها منفی تلقی شوند. همچنین برای استخراج داده‌های مشاهده‌نشده نیز، متناسب با هر دادگان، پیش‌پردازش مناسب را انجام دادیم. این دسته از داده‌های تولیدی برچسب خروجی ندارند و در هر دادگان، جفت‌هایی از کاربرـ محصول را شامل می‌شوند که به ازای آنها هیچ بازخورد ثبت‌شده‌ای در اختیار نداریم.

از تفاوت‌های ساختار خام دو دادگان می‌توان به نحوه‌ی ذخیره‌سازی بازخوردها اشاره کرد که در دادگان Coat به دلیل کم بودن تعداد نمونه‌ها، همه‌ی اطلاعات در قالب یک ماتریس تنک (به دلیل حجم زیاد جفت‌های بدون بازخورد ثبت‌شده) ذخیره شده است که درایه‌های مربوط به جفت‌های بدون بازخورد، مقدار صفر دارند. اما در دادگان بزرگ Yahoo!R۳ فقط جفت‌های کاربرـ محصول دارای

بازخورد به صورت ذکر شناسه‌ی کاربر، شناسه‌ی محصول، و بازخورد کاربر به آن محصول در سطرهایی مجزا گردآوری شده است.

۶ـ۳ پیاده‌سازی مدل‌ها و انجام آزمایش‌ها

در گام بعدی به طراحی پیاده‌سازی مورد نظر خود از راهبردهای پل و تصفیه در قالب شبکه‌های عصبی روبه‌جلو پرداختیم. در این مرحله، توابع هزینه و جزئیات مذکور در [۱] برای دو راهبرد پل و تصفیه را در فرآیند یادگیری شبکه‌های عصبی طراحی‌شده‌ی خود لحاظ کردیم.

در این بخش، با ازمون و خطا روی ابرپارامترهای عمومی یک شبکه‌ی عصبی مانند تعداد لایه‌ها و نورون‌های هر یک، اندازه‌ی بسته‌های داده‌ای ورودی برای تغذیه‌ی شبکه، و ... معماری‌ای با بهترین نتایج را یافتیم. سپس با استفاده از این مدل، برای یافتن بهترین آرایش از ابرپارامترهای ویژه‌ی هر مدل، جستجوی مشبکه‌ای انجام دادیم که توضیح آن در فصل ۲ ذکر شده است. البته در آزمایش‌های مختلف از پیش‌پردازش‌های مختلفی برای هر دادگان استفاده کرده‌ایم که در مؤلفه‌هایی از قبیل آستانه‌ی دودویی‌سازی بازخوردها و درصدی از داده‌های یکنواخت که برای آموزش استفاده‌شده‌اند، متفاوت بودند.

در چرخه‌ی آموزش پیشنهادی‌مان از دو نوع آموزش متفاوت استفاده شده که در اولی به شیوه‌ی متداول یادگیری بانظارت عمل کردیم. این شیوه شامل تقسیم داده‌ها به سه بخش آموزشی، ارزیابی، و آزمایشی می‌شود و در آزمایش‌ها از این شیوه با نام *کلاسیک* یاد کرده‌ایم. اما در نوع دیگر، سعی کردیم چرخه‌ی *بازخورد* موجود در سامانه‌های توصیه‌گر واقعی را شبیه‌سازی کنیم و از این شیوه هم به نام همان مفهومی یاد کرده‌ایم که قصد شبیه‌سازی‌اش را داشته است.

برخی دیگر از آزمایش‌ها نیز به تنظیم ابرپارامترهای خاص‌تر مانند تعداد بالاترین نمونه‌های منتخب برای نوع آموزش چرخه‌ی بازخورد، با فرض استفاده از یک آرایش بهترینِ قبلا یافت‌شده برای همه‌ی ابرپارامترهای دیگر، اختصاص یافته است.

در نهایت نیز در دو آزمایش گسترده‌تر (یکی به ازای هر دادگان)، به بررسی عملکرد مدل‌های پیاده‌سازی‌شده‌ی خود در مقایسه با یک مدل کاملاً ساده پرداختیم که صرف روی اجتماعی از داده‌های یکنواخت و غیریکنواخت موجود آموزش دیده است و به همین دلیل هم در جدول مربوط به این دو آزمایش، یعنی جدول‌های ۵ـ۶ و ۵ـ۱۱ این مدل را «مدل اجتماع» نامیده‌ایم.

۶ـ۴ مسیرهای پیش رو

همان طور که درآغاز پایان‌نامه هم اشاره کردیم، مسئله‌ی پیش‌قدر در سامانه‌های توصیه‌گر ابعاد و ریشه‌های مختلفی دارد؛ در حالی که تمرکز ما در این مطالعه صرفا تمرکز بر یکی از انواع آن، یعنی پیش‌قدر ناشی از مدل پیشین، بوده است. همچنین خط مشی کلی ما برای رویارویی با این مسئله، استفاده از داده‌های یکنواخت بوده و از بین مجموعه‌ی روش‌های بهره‌مند از این رویکرد نیز، مطالعه‌ی عمیق‌تر ما به دو راهبرد خاص معرفی‌شده در [۱] اختصاص داده شده که نهایتا هم منجر به ارائه‌ی یک پیاده‌سازی متفاوت از پیاده‌سازی نویسندگان مقاله‌ی اصلی شد.

یکی از مسیرهای ممکن برای ادامه‌دادن موضوع این پایان‌نامه، می‌تواند مطالعه در مورد مجسم‌سازی[۱] یا تفسیرپذیری اطلاعات مفیدی باشد که از مجموعه‌داده‌های جمع‌آوری‌شده بدون پیش‌قدر (یکنواخت) حاصل می‌شود که اساس راهبردهای پل و تصفیه نیز بر استفاده از این اطلاعات مفید استوار است. همچنین ارتباط احتمالی موجود بین اندازه‌ی مجموعه‌داده‌ی یکنواخت مورد استفاده و کارآیی مدل‌های توسعه‌یافته به کمک آن یک موضوع جالب توجه دیگر است. از دیدگاه مطالعات نظری نیز پی بردن به چرایی تأثیر مثبت استفاده از داده‌های یکنواخت در جهت حل مسئله‌ی پیش‌قدر و ارائه‌ی شواهد نظری محکم‌تر برای آن می‌تواند مفید باشد؛ چرا که اگر چارچوب (های) نظری محکمی در این باره معرفی شود، می‌تواند به طراحی راهبردهای استحصال دانش مؤثرتر برای توسعه‌ی مدل‌های قدرتمندتر نیز کمک کند [۱].

[۱] Visualization

مراجع

[1] D. Liu, P. Cheng, Z. Dong, X. He, W. Pan, and Z. Ming. A general knowledge distillation framework for counterfactual recommendation via uniform data. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 831–840, 2020.

[2] Aggregate knowledge raises. https://venturebeat.com/2006/12/10/aggregate-knowledge-raises-5m-from-kleiner-on-a-roll/. Accessed: 2022-03-21.

[3] H. Abdollahpouri, R. Burke, and B. Mobasher. Controlling popularity bias in learning-to-rank recommendation. In *Proceedings of the eleventh ACM conference on recommender systems*, pages 42–46, 2017.

[4] R. Cañamares and P. Castells. Should i follow the crowd? a probabilistic analysis of the effectiveness of popularity in recommender systems. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 415–424, 2018.

[5] D. C. Liu, S. Rogers, R. Shiau, D. Kislyuk, K. C. Ma, Z. Zhong, J. Liu, and Y. Jing. Related pins at pinterest: The evolution of a real-world recommender system. In *Proceedings of the 26th international conference on world wide web companion*, pages 583–592, 2017.

[6] J. J. Heckman. Sample selection bias as a specification error. *Econometrica: Journal of the econometric society*, pages 153–161, 1979.

[7] B. M. Marlin and R. S. Zemel. Collaborative prediction and ranking with non-random missing data. In *Proceedings of the third ACM conference on Recommender systems*, pages 5–12, 2009.

[8] A. Agarwal, I. Zaitsev, X. Wang, C. Li, M. Najork, and T. Joachims. Estimating position bias without intrusive interventions. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 474–482, 2019.

[9] X. Wang, N. Golbandi, M. Bendersky, D. Metzler, and M. Najork. Position bias estimation for unbiased learning to rank in personal search. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 610–618, 2018.

[10] L. Yang, Y. Cui, Y. Xuan, C. Wang, S. Belongie, and D. Estrin. Unbiased offline recommender evaluation for missing-not-at-random implicit feedback. In *Proceedings of the 12th ACM conference on recommender systems*, pages 279–287, 2018.

[11] A. Swaminathan and T. Joachims. Batch learning from logged bandit feedback through counterfactual risk minimization. *The Journal of Machine Learning Research*, 16(1):1731–1755, 2015.

[12] T. Schnabel, A. Swaminathan, A. Singh, N. Chandak, and T. Joachims. Recommendations as treatments: Debiasing learning and evaluation. In *international conference on machine learning*, pages 1670–1679. PMLR, 2016.

[13] M. Grbovic, V. Radosavljevic, N. Djuric, N. Bhamidipati, J. Savla, V. Bhagwan, and D. Sharp. E-commerce in your inbox: Product recommendations at scale. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1809–1818, 2015.

[14] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[15] F. Vasile, E. Smirnova, and A. Conneau. Meta-prod2vec: Product embeddings using side-information for recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 225–232, 2016.

[16] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.

[17] B. Hidasi, M. Quadrana, A. Karatzoglou, and D. Tikk. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pages 241–248, 2016.

[18] S. Bonner and F. Vasile. Causal embeddings for recommendation. In *Proceedings of the 12th ACM conference on recommender systems*, pages 104–112, 2018.

[19] Concepts of mcar, mar and mnar. https://stefvanbuuren.name/fimd/sec-MCAR.html. Accessed: 2022-05-04.

[20] A. Collins, D. Tkaczyk, A. Aizawa, and J. Beel. A study of position bias in digital library recommender systems. *arXiv preprint arXiv:1802.06565*, 2018.

[21] N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey. An experimental comparison of click position-bias models. In *Proceedings of the 2008 international conference on web search and data mining*, pages 87–94, 2008.

[22] R3 - yahoo! music ratings for user selected and randomly selected songs, version 1.0. https://webscope.sandbox.yahoo.com/catalog.php?datatype=r. Accessed: 2022-05-04.

Abstract

The rapid development of internet-based businesses, such as online shopping websites, has made a massive amount of data available about users' activities and transactions. Trying to utilize this big data has motivated such businesses to employ Recommendation Systems as one of the key elements of their success in the market. The primary intent of the recommendation systems is to personalize the advertisements provided to each user. This way, it is supposed that the probability of performing the desired action (e.g., click, purchase, etc.) by the user on the suggested item would be maximized. One of the significant problems with increasing the performance of recommendation systems is the existence of bias within the training data fed to these systems. This problem is mainly known in the literature as The Previous Model Bias. One approach to address this issue is using the user feedback data, which is collected by suggesting random items without incorporating the users' history at all. Although it is evident that if all the training data of the system is collected randomly, the biased data problem no longer exists, doing so is practically impossible; because it seriously hurts the users' overall satisfaction. Therefore, only a tiny portion of all the training data is obtained from this so-called "uniform data". In this study, we start with a brief introduction to the various types of bias problems in recommendation systems and the relevant suggested solutions in the literature. Then, we explain the details of two label-based strategies called "Bridge" and "Refine" which both use uniform data in the training procedure but in different ways. Moreover, our proposed implementation for these two methods using feedforward neural networks and the results of the corresponding experiments are provided, respectively.

Keywords: Recommendation systems, Biased data problem, Uniform data

Sharif University of Technology

Department of Computer Engineering

B.Sc. Thesis

On the solutions of biased data problem in recommendation systems using uniform data

By:

Amir Abbas Asgari

Supervisors:

Dr. Seyed Abbas Hosseini- Dr. Hamid Reza Rabiee

July 2022