

Newpixie: روش جدید مبتنی بر استفاده از قوانین وابستگی

قوی برای سیستم توصیه گر Pixie

نیما علی‌نیا*^۱ دانشجوی دکتری، سجاد راحتی^۱ دانشجوی کارشناسی ارشد، زینب جعفری طادی^۱ دانشجوی کارشناسی ارشد، مسعود جنگ زرگری^۱ دانشجوی کارشناسی ارشد، سید مهدی وحیدی پور^۱ استادیار

گروه هوش مصنوعی_ دانشکده برق و کامپیوتر_ دانشگاه کاشان_ ایران

چکیده: در دهه‌های اخیر، با پیشرفت فناوری اطلاعات و ارتباطات، نیاز به ساماندهی و بهینه‌سازی فرآیندهای اطلاعاتی و ارتباطی بین افراد و منابع مختلف، بیش از پیش اهمیت یافته است و یکی از روش‌های کارآمد در این زمینه، استفاده از سیستم‌های توصیه‌گر است. این سیستم‌ها می‌توانند به صورت برون‌خط و با آماده کردن لیستی از بهترین توصیه‌ها یا به صورت برخط و بلادرنگ در بهبود تجربه کاربری نقش مهمی را ایفا کنند. از جمله چالش‌های سیستم‌های توصیه‌گر، برقراری تعادل بین سرعت ارائه توصیه‌ها و کیفیت آن‌هاست. این مقاله، با ترکیب پردازش برون‌خط و برخط، به بررسی و ساخت الگوریتمی توصیه‌گر به نام Newpixie می‌پردازد که به دنبال ارائه توصیه‌هایی بر اساس علاقمندی‌های مشترک بین کاربران است؛ در Newpixie، از گرافی مقیاس‌پذیر و دوبخشی متشکل از پین‌ها و بوردها که شامل تعدادی از پین‌های مرتبط به هم در یک موضوع خاص هستند، استفاده می‌شود. در فاز برون‌خط این الگوریتم، با استفاده از قوانین وابستگی موجود در داده‌کاوی، ارتباطات قوی بین پین‌ها استخراج شده و بصورت یال مجازی در گراف اعمال می‌شوند. در فاز برخط نیز با قدم‌زنی تصادفی سوگیرانه نسبت به پیوند قوی بین جفت آیت‌های موجود در بوردها، توصیه‌هایی مناسب برای کاربران فراهم خواهد شد. نتایج این الگوریتم بر مبنای سه معیار ارزیابی، با روش پایه pixie مقایسه شده است. آزمایش‌ها نشان داده است که در الگوریتم Newpixie، سرعت ارائه توصیه‌ها تقریباً ۱۷٪، کیفیت توصیه‌ها نیز تقریباً ۳۴٪ و معیار ارزیابی recall@k در پیش‌بینی پیوند، تقریباً ۲۰٪ نسبت به روش پایه pixie، بهبود یافته است.

واژه‌های کلیدی: توصیه‌گر، پیترست، گراف دوبخشی، پین، برد.

*نویسنده مسئول، ایمیل نویسنده: nima.alinia@grad.kashanu.ac.ir

پست‌های الکترونیک: sajadrahaty2@gmail.com ، nima.alinia@grad.kashanu.ac.ir ،

zeinabjafari911380@gmail.com ، M.G.Zargari@grad.kashanu.ac.ir ، vahidipour@kashanu.ac.ir

Newpixie: A new method based on the use of strong dependency rules for Pixie recommender system

Nima Alinia*¹, Ph.D. student, Sajjad Rahati¹, Master's student, Zainab Jafari Tadi¹, Master's student, Masoud Jang Zargari¹, Master's student, Seyed Mehdi Vahidipour¹, Assistant Professor

¹Department of Artificial Intelligence-Faculty of Electricity and Computer-Kashan University-Iran

Abstract:

In recent decades, with the advancement of information and communication technology, the need for organizing and optimizing informational and communicational processes among individuals and various resources has gained increased importance. One effective approach in this field is the use of recommender systems. These systems can significantly enhance user experience either offline, by preparing a list of the best recommendations, or online, in real-time. One of the challenges of recommender systems is achieving a balance between the speed of delivering recommendations and their quality. This paper presents the development and evaluation of a recommender algorithm named "Newpixie," which aims to provide recommendations based on common interests among users. The Newpixie algorithm utilizes a scalable, bipartite graph comprising pins and boards, where each board contains a number of related pins on a specific topic. In the offline phase of the algorithm, strong associations between pins are extracted using association rules in data mining and applied as virtual edges in the graph. In the online phase, biased random walks on the strong links between pairs of items within the boards are employed to generate suitable recommendations for users. The performance of the Newpixie algorithm is evaluated against the baseline pixie method based on three evaluation metrics. Experimental results indicate that the Newpixie algorithm improves recommendation speed by approximately 17%, recommendation quality by about 34%, and the recall@k metric in link prediction by roughly 20% compared to the baseline pixie method.

Keywords: recommendation system, Pinterest, bipartite graph, pin, board.

* Corresponding author, author's email : Nima Alinia , nima.alinia@grad.kashanu.ac.ir.

author's email: nim.alinia@grad.kashanu.ac.ir , sajadrahaty2@gmail.com , zeinabjafari911380@gmail.com ,
M.G.Zargari@grad.kashanu.ac.ir , vahidipour@kashanu.ac.ir.

تعداد کاربران فعال ماهانه پینترست به بیش از ۴۹۰ میلیون می‌رسد و بیش از میلیاردها عکس و فیلم مختلف در آن به اشتراک گذاشته شده است.^۶ پینترست برای بهبود ارائه خدمات به کاربران از یک سیستم توصیه‌گر به نام pixie استفاده می‌کند که به صورت بلادرنگ^۷ در کسری از ثانیه تعداد زیادی عکس و فیلم به آن‌ها توصیه خواهد کرد؛ این توصیه با توجه به عکس‌ها و فیلم‌هایی است که در پروفایل کاربر قرار دارد [۳].

در روش pixie ارتباط موجود در میان پین‌ها و بوردها به صورت یک گراف دو بخشی مقیاس‌پذیر^۸ نشان داده می‌شود. یک بخش از گره‌های این گراف، شامل پین‌ها و بخش دیگر شامل بوردها است. این الگوریتم پس از بررسی دقیق ارتباط بین گره‌های گراف، توصیه‌های شخصی را به کاربران ارائه می‌دهد [۴].

درواقع، کاری که الگوریتم توصیه‌گر pixie انجام می‌دهد، یک نوع پیش‌بینی پیوند^۹ در گراف است. در عمل پیش‌بینی پیوند، با استفاده از پیوندهایی که در گراف وجود دارند، پیوندهایی که ممکن است در آینده ایجاد شوند، پیش‌بینی می‌شوند [۵و۶].

چالش اصلی الگوریتم‌های بلادرنگ مانند pixie، ایجاد تعادل بین سرعت و کیفیت توصیه‌های ارائه شده به کاربر است؛ یعنی از طرفی توصیه‌ها باید در سریع‌ترین زمان ممکن به کاربر ارائه شود و از طرف دیگر باید بیشترین مطابقت را با علایق و نیازهای کاربر داشته باشد [۳].

در این مقاله، با هدف افزایش سرعت و همچنین حفظ کیفیت توصیه‌ها، الگوریتمی با نام Newpixie ارائه شده است که توسعه‌ای بر الگوریتم pixie است. مبنای الگوریتم Newpixie، ترکیب پردازش برون‌خط و پردازش برخط است. در پردازش برون‌خط، داده‌ها با استفاده از روش‌های داده‌کاوی تحلیل و

سیستم‌های توصیه‌گر، ابزارهای هوشمندی هستند که با استفاده از الگوریتم‌ها و مدل‌های یادگیری ماشین، به تحلیل سابقه فعالیت‌ها و علاقمندی‌های کاربران می‌پردازند تا به آن‌ها محتوا، محصول یا خدمات مناسبی پیشنهاد دهند. این سیستم‌ها نه تنها به بهبود تجربه کاربران کمک می‌کنند بلکه می‌توانند در بهبود فرآیندهای تصمیم‌گیری و افزایش بهره‌وری در مواردی از جمله خرید برخط^۱، مطالعه مقالات علمی، تماشای فیلم و موارد دیگر نیز کارا باشند [۱و۲].

با رشد روز افزون تعداد کاربران و حجم اطلاعات موجود در شبکه‌های اجتماعی، چالش‌هایی مانند سرعت ارائه توصیه در سیستم‌های توصیه‌گر مطرح می‌شود [۲]. یکی دیگر از چالش‌های این سیستم‌ها، ارائه توصیه‌هایی متناسب با هر تغییر رفتار و علاقمندی کاربران در هر بازه می‌باشد. از این رو آن‌ها می‌توانند با استفاده از سیستمی پویا، بهترین پیشنهادات ممکن را دریافت کنند. همین موضوع می‌تواند ذهن کاربر را به سمت سیستم‌هایی، از نوع برون‌خط^۲، که با توجه به سابقه فعالیت کاربران کاربردی‌ترین توصیه‌ها را ارائه می‌کند، سوق دهد. لازم به ذکر است که بیشتر کاربردهای توصیه‌گرها در سیستم‌های برخط است. پینترست^۳ یک شبکه اجتماعی است که کاربران می‌توانند در پروفایل خود عکس‌ها و فیلم‌هایشان را به اشتراک گذاشته و/یا عکس‌ها و فیلم‌های دیگران را ذخیره کنند. هریک از این عکس‌ها و فیلم‌ها، پین^۴ نامیده می‌شوند. بوردها مجموعه‌ای از پین‌ها هستند که توسط کاربران برای ساماندهی به محتواهای مورد علاقه‌شان ایجاد می‌شوند. هر کاربر می‌تواند پین‌های موجود در بوردهای سایر کاربران را در بوردهای خود ذخیره کند و یا آن‌ها را بپسندد.

^۶ <https://blog.hootsuite.com/pinterest-statistics-for-business/>

^۷ real time

^۸ scalable

^۹ link prediction

^۱ online

^۲ offline

^۳ <https://www.pinterest.com>

^۴ pin

^۵ board

ایجاد شده است. این ایده، در قدم‌زنی تصادفی، برای شمارش بازدید هر گره با شروع تصادفی از دیگر گره‌ها استفاده شده است.

(۳) در الگوریتم **Newpixie**، به هرکدام از بین‌های مجموعه پرس‌وجو^{۱۱}، متناسب با اهمیتشان وزنی اختصاص داده شده است و با این کار، کل سابقه رفتاری کاربران حفظ شده است.

۲ مفاهیم پایه

در این بخش از مقاله، به بررسی دقیق الگوریتم پیکسی و اصول اساسی آن پرداخته می‌شود. این بحث شامل توضیحی واضح و دقیق از چگونگی کارکرد الگوریتم مذکور می‌باشد. علاوه بر این، تحلیلی از قواعد وابستگی مورد استفاده در داده‌کاوی با هدف بررسی چگونگی استخراج ویژگی‌ها و شناسایی ارتباطات معنادار بین گره‌های درون گراف، ارائه خواهد شد.

۲.۱ روش pixie

pixie به عنوان یک روش برای ارائه بهترین توصیه‌های شخصی به کاربران با استفاده از مجموعه داده‌های پینترست و بر مبنای قدم‌زنی تصادفی پیاده‌سازی شده است. در اجرای این روش، از ترکیب چندین قدم‌زنی تصادفی مستقل استفاده می‌شود به گونه‌ای که بعد از مشاهده هر پین در سابقه فعالیت‌های کاربر به آن پین پاداشی تعلق می‌گیرد، این موضوع منجر به استفاده بهینه از سابقه فعالیت‌های کاربر در ارائه بهترین توصیه‌ها به او خواهد شد. روش **pixie** از یک معیار همگرایی خاص برای جلوگیری از توقف نامناسب استفاده می‌کند به نوعی باید گفت این روش با ساخت یک گراف دوبخشی از ارتباطات کاربران و آیتم‌ها به اسم **G**، در نظر دارد که بعد از هر قدم‌زنی تصادفی در این گراف تعداد مشخصی از پین‌ها یعنی np را به میزان مشخصی از بازدید یعنی n_p برساند و سپس الگوریتم را خاتمه داده و از یک آرایه‌ای شامل تعداد بازدید پین‌ها به عنوان خروجی استفاده کند [۳].

پردازش می‌شوند و قانون‌های قوی موجود بین پین‌ها استخراج می‌شوند. این قانون‌ها، بعنوان یال‌های مجازی بین پین‌ها در نظر گرفته شده و به گراف دوبخشی اولیه اضافه می‌شوند. در پردازش برخلاف با تکیه بر قدم‌زنی تصادفی^{۱۲} و یال‌های مجازی که در مرحله برون‌خط به گراف اولیه اضافه شده‌اند، به صورت سریع، بهینه و متناسب با علاقمندی‌ها و فعالیت جدید کاربر، توصیه‌هایی به او داده می‌شود. عمل قدم‌زنی تصادفی روی گراف، از یک گره دلخواه روی گراف شروع می‌شود، سپس یکی از همسایگان آن گره، بصورت تصادفی انتخاب می‌شود و این عمل تا رسیدن به یک معیار توقف از پیش تعیین شده، ادامه یافته و یک دنباله متناهی از گره‌ها ایجاد می‌شود [۷].

الگوریتم **Newpixie** نسبت به گراف ورودی مستقل است؛ چرا که می‌تواند مستقل از اندازه ورودی توصیه‌هایی را به کاربران متناسب با علاقمندی‌های مشترک آن‌ها و مشابهت بوردهایشان ارائه دهد. آزمایش‌ها نشان می‌دهد که الگوریتم **Newpixie** از سرعت بالاتری نسبت به روش پایه برخوردار است ضمن اینکه کیفیت در این روش نیز به معیارهایی مانند تعداد گام‌های قدم‌زنی تصادفی وابسته است.

مهمترین نوآوری‌ها و مزایای روش پیشنهادی نسبت به روش پایه در این مقاله به شرح زیر است:

- (۱) الگوریتم **Newpixie**، ابتدا بوردها را به صورت عمومی توصیه کرده و سپس پین‌های ذخیره شده در آن‌ها را به صورت شخصی به کاربران خود ارائه می‌کند. باید بدانید که این روش از ارائه توصیه‌های برخلاف و برون‌خط به صورت همزمان برخوردار است و با استخراج قوانین وابستگی بین داده‌ها، بهترین توصیه‌ها را بر مبنای سابقه فعالیت کاربران ایجاد می‌کند.
- (۲) در الگوریتم **Newpixie**، با استفاده از استخراج قوانین وابستگی، میان بعضی از گره‌های گراف، یال مجازی

^{۱۱} query

^{۱۲} Random walk

۲,۲ قوانین وابستگی در داده کاوی

قوانین وابستگی در داده کاوی، به عنوان یکی از ابزارهای کلیدی در کشف و استخراج الگوها از میان داده‌های عظیم، نقش مهمی ایفا می‌کنند. این روش از تحلیل، امکان شناسایی ارتباطات و وابستگی‌های بین آیتم‌ها یا بین‌های مختلف موجود در یک مجموعه داده را فراهم می‌آورد و به عنوان پلی بین داده‌های خام و اطلاعات مفید عمل می‌کند، به طوری که می‌تواند برای ایجاد سیستم‌های توصیه‌گری که پیشنهاد‌های مبتنی بر تجزیه و تحلیل داده‌ها ارائه می‌دهند، استفاده شود. [۹].

در حقیقت قوانین وابستگی از نوع "اگر و آنگاه" هستند، که این به معنای شناسایی الگوهایی است که در آنها حضور یک مجموعه از آیتم‌ها (پین‌ها) به صورت معناداری با حضور مجموعه‌ای دیگر از آیتم‌ها همراه است. به عنوان مثال، در یک برنامه کاربردی مانند پینترست، استفاده از این قوانین نشان می‌دهد کاربرانی که پین A را می‌پسندند، اغلب به پین B نیز علاقمند هستند و این می‌تواند منجر به کارآمد بودن سیستم توصیه‌گر و ارائه بهترین توصیه‌ها برای کاربران باشد.

قانون‌های وابستگی قوی بین پین‌ها بر اساس تعاملات قبلی پین‌ها استخراج می‌شود. پین p با پین q وابستگی قوی دارد، اگر در بوردهایی که پین p حضور دارد پین q نیز با یک احتمالی حضور داشته باشد. در این مقاله، برای بدست آوردن این قوانین وابستگی از الگوریتم FPGrowth [۱۰] استفاده شده است.

۳ کارهای مرتبط

سیستم‌های توصیه‌گر، یک زمینه تحقیقاتی گسترده هستند که به خوبی و در بسیاری از جهات مطالعه شده‌اند. در اینجا، کاربرد چند سیستم توصیه‌گر روی شبکه‌های اجتماعی با گراف‌های بزرگ آورده شده است.

در الگوریتم pixie، برای هر کاربر، مجموعه‌ای از پین‌های پرس‌وجو ساخته می‌شود که نشان‌دهنده سابقه فعالیت کاربران در پینترست می‌باشد که با Q تعریف می‌شود. هر پین عضو این مجموعه با $q \in Q$ نشان داده می‌شود. به هر یک از پین‌هایی موجود در مجموعه Q ، متناسب با زمان ارتباط کاربران با آن پین، وزنی نسبت داده می‌شود که میزان اهمیت آن پین در سابقه فعالیت کاربران را مشخص خواهد کرد. این موضوع باعث می‌شود که علاوه بر پین‌های فعلی، پین‌های قبلی نیز در ارائه بهترین توصیه‌ها به کاربران موثر باشند. این روش از امتیاز و پاداشی که به هر یک از پین‌ها متناسب با معیارهای مشخص نسبت می‌دهد به عنوان عاملی برای تقویت خود استفاده می‌کند و این عامل منجر به کارآمد بودن آن می‌شود [۸ و ۳].

یکی دیگر از نوآوری‌های کاربردی در pixie، تمایل توصیه‌ها به سمت ویژگی‌های خاص دریافت شده از سمت کاربر است؛ به گونه‌ای که متناسب با زبان و علاقمندی‌های خاص کاربر، توصیه‌ای شخصی‌سازی شده به او ارائه می‌شود. روش pixie نه تنها ارتباط چندگانه پین‌ها با پین فعلی کاربر را نادیده نمی‌گیرد بلکه در حقیقت این الگوریتم به پین‌هایی که در تعداد دفعات بیشتری با پین‌های موجود در پرس‌وجو‌های کاربر ارتباط دارند، امتیازی داده و از این امتیازها نیز برای تقویت سیستم توصیه‌کننده خود استفاده می‌کند. pixie، برای خود یک شرط توقف نیز قائل می‌شود تا با توقف زودهنگام در زمانی که هر پین به مقدار معینی بازدید رسیده باشد اجرای الگوریتم متوقف شود. [۳].

اگرچه pixie، با تمامی ایده‌ها و مزیت‌هایی که دارد می‌تواند توصیه‌هایی شخصی به کاربران خود ارائه دهد، اما امکان ارائه توصیه‌های اجتماعی و عمومی به گروهی از کاربرانی که علاقمندی‌های مشترکی دارند و در مجموعه‌های پرس‌وجو‌یشان شباهت زیادی پیدا می‌شود وجود ندارد؛ چرا که این روش تنها به دنبال ارائه توصیه‌های شخصی متناسب با علاقمندی‌ها کاربران و سابقه فعالیت آن‌ها می‌باشد.

مجموعه $P \cup B$ نیز معرف مجموعه گره‌های گراف G بوده که در این گراف دویخشی ارتباط بین پین‌ها و بوردها یا پین‌ها با پین‌ها را نشان داده است. اگر کاربری پین p را در برد b ذخیره کند، یک یال $e \in E$ بین پین $p \in P$ و برد $b \in B$ به وجود می‌آید. همچنین از $E(p)$ برای نشان دادن گره‌های برد متصل به پین p و از $E(b)$ برای پین‌های متصل به برد b استفاده می‌شود و فرض بر این است که گراف G متصل است که در عمل نیز چنین است. اما یال‌های E_p حاصل از ارتباطات میان پین‌ها با خودشان در این گراف دو بخشی است که این ارتباط با یال‌ها مجازی نشان داده می‌شود.

در الگوریتم Newpixie، تمرکز روی ارتباطات بین پین‌ها می‌باشد و قدم‌زنی تصادفی به دنبال بدست آوردن تعداد تکرار پین‌های مشخص در ارتباط با یکدیگر و در بوردهای متفاوت است تا بتواند توصیه‌ای بر روی پین‌هایی که به صورت مشترک بین کاربران متفاوت استفاده شده است، بدست آورد.

در ادامه، ابتدا الگوریتم قدم‌زنی ساده که در روش pixie معرفی شده توضیح داده می‌شود. در مرحله بعد، الگوریتم قدم‌زنی توسعه یافته مورد استفاده قرار می‌گیرد. این قسمت در بخش ۳/۲ توضیح داده می‌شود. لازم به ذکر است، این الگوریتم در روش پایه pixie وجود دارد و عیناً در این قسمت نیز ارائه شده است.

۴،۱ الگوریتم ساده قدم زدن تصادفی از یک پین

پرس و جو

الگوریتم قدم‌زنی تصادفی از جمله پرکاربرترین الگوریتم‌ها در سیستم‌های توصیه محسوب می‌شود. ورودی اول این الگوریتم قدم‌زنی، گراف دو بخشی G و ورودی دوم یک پین پرس و جو به نام q است. این الگوریتم با توجه به گراف G و با شروع از q تعدادی قدم‌زنی تصادفی انجام داده و تعداد دفعاتی که هر پین p از گراف G مشاهده شده است را ذخیره می‌کند (شبه کد ۱).

اگر اوال و همکاران [۱۱] یک سیستم توصیه‌گر برای شخصی‌سازی صفحه اصلی^{۱۲} لینکدین برای هر کاربر ارائه دادند. این مطالعه، با چالش‌های مدیریت تعداد زیادی از کاربران و آیتم‌ها و اطمینان از ارائه توصیه‌های بلادرنگ روبرو بود. روش پیشنهادی، شامل استفاده از ترکیب فیلترسازی مشارکتی و روش‌های مبتنی بر محتوا با بهره‌گیری از منابع محاسباتی قابل توجه بود. با این حال، این روش با مشکلات مقیاس‌پذیری در مواجهه با افزایش تعداد کاربران و آیتم‌ها و هزینه‌های بالای محاسباتی برای حفظ توصیه‌های بلادرنگ مواجه بود.

کاوینگتون و همکاران [۱۲] یکی دیگر از سیستم‌های توصیه‌گر را معرفی کردند که به مسئله ارائه توصیه‌های ویدیویی به کاربران YouTube می‌پرداخت. این مطالعه، با چالش‌های مدیریت یک کاتالوگ وسیع از ویدئوها و نیاز به توصیه‌های بلادرنگ به کاربران روبرو بود.

گوئل و همکاران [۱۳] سیستم توصیه‌گر "Who-to-Follow" را معرفی کردند که به مسئله توصیه کاربران برای دنبال کردن در توئیتر می‌پرداخت. این مطالعه، با چالش‌هایی همچون ارائه توصیه‌های بلادرنگ و مدیریت یک گراف اجتماعی بزرگ روبرو بود و روش پیشنهادی آن نیز شامل استفاده از الگوریتم‌های مبتنی بر گراف مانند PageRank شخصی‌سازی شده بود. با این حال، این روش با محدودیت‌هایی مانند کند بودن الگوریتم‌های گراف در گراف‌های بسیار بزرگ و چالش‌های مداوم در حفظ عملکرد بلادرنگ مواجه بود.

۴ الگوریتم پیشنهادی Newpixie

در این روش از یک مجموعه داده بزرگ متشکل از ارتباطات کاربران و آیتم‌ها استفاده می‌کند با استفاده از این مجموعه داده‌ها می‌توان گراف‌هایی دو بخشی و بدون جهت مانند $G = (P, B, E, E_p)$ ساخت. در اینجا P نشان‌دهنده مجموعه‌ای از پین‌ها و B نشان‌دهنده مجموعه‌ای از بوردهاست.

^{۱۲} home feed

۴,۲ الگوریتم توسعه یافته قدمزنی تصادفی از یک پین پرس وجود

الگوریتم pixie به دلیل تمرکز بر روی سرعت پاسخدهی به کاربران و بلادرنگ بودن توصیه‌ها، وجود ارتباطات بین پین‌های تکرار شده در بوردهای مختلف را نادیده می‌گیرد. در این مقاله، برای پوشش دادن به این نقص، با در نظر گرفتن ارتباطات بین پین‌های تکرار شده در بوردها و تاثیر آن‌ها در قدمزنی تصادفی روشی جدید به نام Newpixie ارائه شده است. جزئیات اجرای این الگوریتم در شبه کد ۲ نشان داده شده است.

در این الگوریتم، سعی شده است تا از شباهت‌های علایق کاربران با یکدیگر استفاده شود و با توجه به این شباهت‌ها، با سرعت بیشتر، توصیه‌های جذاب‌تر و با کیفیت‌تری به کاربران ارائه شود. یکی از راه‌های یافتن شباهت علایق کاربران با یکدیگر، استفاده از قوانین وابستگی در داده‌کاوی است.

شبه کد ۲ الگوریتم توسعه یافته قدمزنی تصادفی از یک پین q

الگوریتم Newpixie، شامل دو مرحله پردازش می‌باشد: (۱) پردازش برون‌خط که شامل پیش پردازش داده‌ها، استخراج قانون‌های وابستگی قوی بین پین‌ها و ذخیره کردن این قانون‌ها می‌باشد. (۲) پردازش برخط که با کمک قانون‌های وابستگی ذخیره شده در مرحله برون‌خط، یک قدمزنی تصادفی سوگیرانه روی گراف G اجرا می‌شود.

در الگوریتم ساده قدمزنی تصادفی pixie (شبه کد ۱)، بعد از مشاهده هر پین p ، مقدار $V_q[p]$ یکی زیاد می‌شود؛ در حالی که با توجه به شبه کد ۲، در روش پیشنهادی این مقاله علاوه بر اضافه شدن مقدار بازدید، با استفاده از قوانین وابستگی قوی که در مرحله پیش پردازش به دست آمده، $V_q[p]$ برای پین‌هایی که با پین p وابستگی دارند نیز به اندازه δ افزایش می‌یابد و این مقدار افزایش بازدید به اینصورت است:

در اجرای الگوریتم قدمزنی تصادفی pixie بر روی گراف دوبخشی، در ابتدا پین ورودی q به عنوان پین جاری و نقطه شروع قدمزنی تصادفی در نظر گرفته شده است. سپس برد b از میان $E(q)$ که شامل تمام بوردهای متصل به پین q هستند انتخاب خواهد شد. در ادامه پین p از میان $E(b)$ که شامل پین‌های متصل به برد b هستند، انتخاب شده و p را بعنوان پین جاری در نظر گرفته و این مراحل تکرار می‌شود. طول هر قدمزنی تصادفی با مقدار α تعیین می‌شود و تعداد کل گام‌ها در این الگوریتم نیز با مقدار N تعیین می‌گردد. در اینجا شرط توقف با مقدار مشخصی بازدید بر روی تعداد مشخصی از پین‌ها اعمال می‌شود؛ یعنی اگر تعداد پین n_p بار مشاهده شوند، الگوریتم متوقف می‌شود که این شرط، منجر به افزایش سرعت الگوریتم خواهد شد (خط ۱۵ از شبه کد ۱). در انتهای اجرای این الگوریتم می‌توان لیستی از تعداد بازدید هر پین بدست آورد [۳].

الگوریتم ساده قدمزنی تصادفی sRW

ورودی:

q : پین پرس وجود، E : مجموعه یال‌های گراف G ، u : ویژگی‌های کاربران، α : احتمال راه‌اندازی مجدد قدمزنی، N : تعداد کل گام‌های قدمزنی تصادفی، n_p : حداکثر تعداد پین‌های بازدید شده، n_v : حداکثر تعداد بازدید هر پین

خروجی:

V : لیستی از تعداد بازدید هر پین

```
totSteps = 0, v = 0
nHighVisited = 0
repeat
  currPin = q
  currSteps = SampleWalkLength( $\alpha$ )
  for i = [1 : currSteps] do
    currBoard =
      E(currPin) [PersonalizedNeighbor (E, U) ]
    currPin =
      E(currBoard) [PersonalizedNeighbor (E, U) ]
    V[currPin] = V[currPin] + 1
    if V [currPin] =  $n_v$  then
      nHighVisited = nHighVisited + 1
      totSteps = totSteps + currSteps
  until totSteps  $\geq$  N or nHighVisited  $>$   $n_p$ 
return V
```

شبه کد ۱ الگوریتم ساده قدمزنی تصادفی از یک پین q [۳]

الگوریتم توسعه یافته قدمزنی تصادفی از یک `newRW` ورودی:

q : بین پرس و جو، E : مجموعه یال‌های گراف G ، u : ویژگی‌های کاربر، α : احتمال راه‌اندازی مجدد قدمزنی، N : تعداد کل گام‌های قدمزنی تصادفی، n_p : حداکثر تعداد بین‌های بازدید شده، n_v : حداکثر تعداد بازدید هر بین، S_q : مجموعه بین‌هایی که با بین q یال مجازی دارند. $gamma$: یک مقدار *Real*

خروجی:

V : لیستی از تعداد بازدید هر بین

```
totSteps = 0, V = 0
nHighVisited = 0
repeat
  currPin = q
  currSteps = SampleWalkLength(_)
  for i = [1 : currSteps]
    currBoard =
      E(currPin) [PersonalizedNeighbor (E_v, U)]
    currPin =
      E(currBoard) [PersonalizedNeighbor (E_v, U)]
    V[currPin] = V[currPin] + 1
  for all s ∈ S_q do
    if gamma ≤ 1/log(Ap)
      delta = gamma
    else
      delta = 1/log(Ap)
      V[s] = V[s] + delta
    if V[currPin] == n_v then
      nHighVisited = nHighVisited + 1
      totSteps = totSteps + currSteps
  until totSteps ≥ N or nHighVisited > n_p
return V
```

ضریب مقیاس‌پذیری S_q باید برای هر بین $q \in Q$ به صورت مجزا محاسبه شود. مقدار $|E(q)|$ درجه بین q در گراف است و C برابر با بیشینه درجه بین‌های موجود در گراف است [۳].

با توجه به N_q ، به بین‌های دارای درجه بالاتر تعداد گام‌های بیشتری برای قدمزنی تصادفی اختصاص داده می‌شود و بین‌های با درجه کمتر نیز تعداد گام‌های کافی را دریافت می‌کنند.

شبه کد ۳ الگوریتم قدمزنی تصادفی چندگانه روی مجموعه پرس و جو Q

برای هر بین $q \in Q$ ، عملیات قدمزنی تصادفی با طول N_q انجام شده و تعدادی بین مرتبط با q توصیه می‌شوند. بعد از اتمام قدمزنی تصادفی برای تمامی بین‌های q ، با استفاده از $v_q[p]$ می‌توان تعداد دفعات مشاهده شدن بین p را در قدم زدن‌های تصادفی شروع شده از بین q مشاهده کرد (شبه کد ۳). در الگوریتم `pixie`، با استفاده از رابطه زیر به بین‌های p که با تعداد بالاتری

$$\delta = \begin{cases} \gamma, & \gamma \leq \frac{1}{\log(Ap)} \\ \frac{1}{\log(Dp)}, & \gamma > \frac{1}{\log(Ap)} \end{cases} \quad (1)$$

در این رابطه، γ یک مقدار ورودی بین ۰ و ۱ است، Ap تعداد بین‌هایی را نشان می‌دهد که با بین p وابستگی قوی دارند و Dp نیز درجه بین p در گراف را نشان می‌دهد. در نهایت، اجرای این الگوریتم باعث می‌شود که مقدار بازدید تمامی بین‌ها تکرار شده در بوردهای مختلف نیز بدست آمده و از این خروجی برای ارائه توصیه‌های عمومی به کاربران استفاده شود.

۴.۳ الگوریتم قدمزنی مبتنی بر مجموعه‌ای از بین‌های

پرس و جو

Q ، مجموعه پرس و جو ی هر کاربر است که شامل تمامی سابقه فعالیت او می‌باشد؛ بنابراین برای هر بین q که در رابطه $q \in Q$ صادق باشد از w_q برای نشان داده وزن متعلق به آن بین در مجموعه پرس و جو استفاده می‌شود. الگوریتم قدمزنی تصادفی `pixie`، یک مقدار شمارشی $v_q[p]$ با شروع قدمزنی تصادفی از بین q تا رسیدن به بین p به دست می‌آورد. در این الگوریتم پایه، طول قدم زدن تصادفی برای هر بین q برابر با مقدار N است. اما در واقع برای بین‌های q که درجه یا تعداد همسایگان بیشتری در گراف دارند، نیاز است که تعداد گام‌های بیشتری برداشته شود تا بتوان با کیفیت بیشتری بین‌های مرتبط با آن را پیدا کرد.

در این الگوریتم به ازای هر بین q بجای استفاده از N برای تعداد گام‌ها، از مقدار N_q استفاده می‌شود که این مقدار N_q از فرمول زیر به دست می‌آید:

$$N_q = w_q N \frac{S_q}{\sum_{r \in Q} S_r} \quad (2)$$

N_q تعداد گام‌های قدم زدن تصادفی در الگوریتم است که از بین q شروع می‌شود [۳]. S_q نیز ضریب مقیاس‌پذیری بین q است که از رابطه زیر به دست می‌آید:

$$S_q = |E(q)| \cdot (C - \log |E(q)|) \quad (3)$$

پس از دریافت ورودی‌ها، با توجه به گراف تغییر یافته، الگوریتم NewRWM (شبه‌کد ۳) روی مجموعه بین‌های پرس‌وجو اجرا می‌شود. درون این الگوریتم،

<p>الگوریتم NewRWM</p> <p>ورودی:</p> <p>Q: مجموعه بین‌های پرس‌وجو، E: مجموعه یال‌های گراف G، u: ویژگی‌های کاربران، α: احتمال راه‌اندازی مجدد قدم‌زنی، N تعداد کل گام‌های قدم‌زنی تصادفی، n_p: حداکثر تعداد بین‌های بازدید شده، n_v: حداکثر تعداد بازدید هر بین، S_q: مجموعه بین‌هایی که با بین q، یال مجازی دارند. γ: یک مقدار <i>Real</i> خروجی: V: لیستی از تعداد بازدید هر بین</p>
<pre> for all $q \in Q$ do $N_q = Eq \cdot 2$ $V_q = \text{newRW}(q, E, U, \alpha, N_q, S_q, \gamma)$ for all $p \in G$ do $V[p] = \left(\sum_{q \in Q} \sqrt{V[q]} \right)^2$ Return V </pre>

برای هر بین پرس‌وجو، الگوریتم newRW (شبه‌کد ۲) فراخوانی و اجرا می‌شود. در ادامه با استفاده از رابطه ۴، مقدار دقیق مشاهده شدن هر بین، محاسبه می‌شود. خروجی این الگوریتم، مجموعه بین‌هایی است که بیشتر از بقیه بین‌ها مشاهده شده‌اند.

در نهایت، خروجی مرحله برخط، بعنوان خروجی الگوریتم Newpixie، در نظر گرفته می‌شود و بعنوان بهترین توصیه‌ها، به کاربر پیشنهاد می‌شود.

از بین‌های ورودی q مرتبط هستند، پاداش داده می‌شود و به اصطلاح تقویت می‌شوند:

$$V[p] = \left(\sum_{q \in Q} \sqrt{V[q]} \right)^2 \quad (4)$$

که در آن $V[p]$ مقدار کلی مشاهده شدن بین p برای مجموعه پرس‌وجو Q است. الگوریتم با استفاده از N_q ، تعداد گام‌های قدم‌زنی تصادفی برای هر بین q را کاهش می‌دهد اما به این مقدار بسنده نکرده و تلاش می‌کند تا این مقدار را تا جایی که ممکن است کاهش دهد. برای این مهم، از دو مقدار صحیح n_p و n_v استفاده می‌کند و این مفهوم را بیان می‌کند که عملیات قدم‌زدن تصادفی با شروع از بین q زمانی پایان می‌یابد که حداقل تعداد n_p بین، به تعداد n_v بار مشاهده شده باشند.

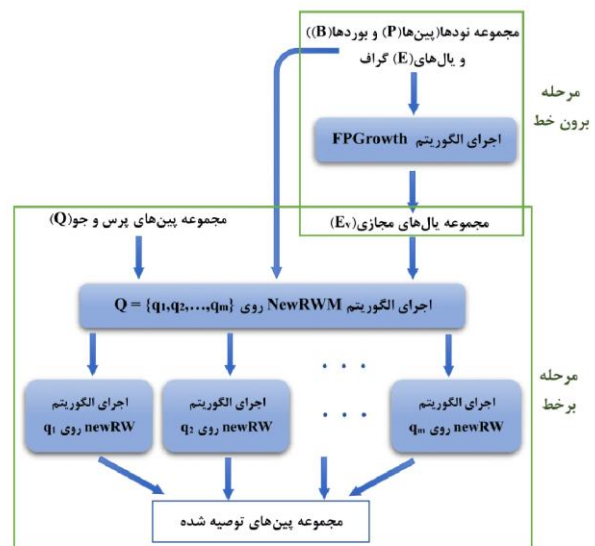
۴،۴ مروری بر الگوریتم پیشنهادی Newpixie

در این بخش، با توجه به شکل ۱، مراحل الگوریتم پیشنهادی Newpixie، به صورت کلی و گام به گام توضیح داده شده است. Newpixie از دو مرحله کلی تشکیل شده است:

(۱) مرحله برون‌خط: ورودی‌های این مرحله، مجموعه گره‌ها (بین‌ها و بوردها) و یال‌های گراف دوبخشی می‌باشند. در این مرحله، با استفاده از الگوریتم FPGrowth، قانون‌های وابستگی قوی بین بین‌های گراف، استخراج شده و ذخیره می‌شوند. این قوانین وابستگی استخراج شده، بعنوان یکی از ورودی‌های مرحله بعد در نظر گرفته می‌شوند.

(۲) مرحله برخط: این مرحله سه ورودی دارد: الف) مجموعه گره‌ها (بین‌ها و بوردها) و یال‌های گراف دوبخشی. ب) قوانین وابستگی استخراج شده در مرحله قبل که بعنوان یال‌های مجازی، به گراف اولیه اضافه می‌شوند. ج) مجموعه بین‌های پرس‌وجو که با توجه به آنها قرار است توصیه‌هایی به کاربر داده شود.

برای مقایسه الگوریتم پیشنهادی این مقاله و الگوریتم پایه pixie سه آزمایش مختلف طراحی و انجام شده است. در آزمایش اول که با هدف مقایسه سرعت این دو الگوریتم طراحی شده است، از تعداد گام‌های طی شده در قدم زنی تصادفی برای ارزیابی سرعت هر الگوریتم استفاده می‌شود. در آزمایش دوم برای مقایسه کیفیت دو الگوریتم، برای هر برد یک نام دسته^{۱۳} در نظر گرفته شده است به این معنی که برد و پین‌های موجود در آن نیز از یک دسته محسوب می‌شوند در این آزمایش از معیار شاخص جاکارد^{۱۴} برای ارزیابی کیفیت توصیه‌های ارائه شده توسط هر الگوریتم استفاده می‌شود.



شکل ۱ مراحل اجرای الگوریتم Newpixie

آزمایش سوم با هدف استفاده از یکی از معیارهای ارزیابی رایج در سیستم‌های توصیه‌گر طراحی شده است. برای این منظور، الگوریتم توصیه‌گر پیشنهادی به گونه‌ای تغییر داده شد که برای هر مجموعه از پین‌های پرس و جو، علاوه بر مجموعه پین‌ها، یک مجموعه از بوردها نیز توصیه شوند. با استفاده از این تغییر، دو الگوریتم از جهت پیش‌بینی لینک و با استفاده از معیار ارزیابی $recall@k$ مورد مقایسه قرار گرفته‌اند. این آزمایش‌ها روی پلتفرم Colab شرکت گوگل^{۱۵} اجرا شده است.

آزمایش‌ها روی مجموعه داده پیترست مربوط به سال ۲۰۱۵ انجام شده است که شامل ۸۸۰۰۰۰ پین، ۴۶۰۰۰۰ برد، ۲۶۵۰۰۰۰ یال و ۵۶۴ دسته برای بوردها می‌باشد.

در ادامه این بخش، در قسمت اول، معیارهای ارزیابی استفاده شده و همچنین روش مقایسه الگوریتم پیشنهادی این مقاله و الگوریتم pixie مورد بحث قرار می‌گیرد. در قسمت دوم، مقایسه سرعت بر اساس پارامتر γ و نتایج آن بررسی می‌شود، در قسمت سوم، کیفیت دو الگوریتم بر اساس شاخص جاکارد مورد مقایسه قرار می‌گیرد و در قسمت چهارم، دو الگوریتم بر اساس

۵ آزمایش‌ها

سیستم‌های توصیه‌گر رایج، معمولاً روی یک گراف دوبخشی کار می‌کنند (در پیترست، یک بخش شامل پین‌ها و بخش دیگر شامل بوردها است). سیستم توصیه‌گر این مقاله که به طور اختصاصی، برای پیترست طراحی شده است، از دو جهت متفاوت از سایر سیستم‌های توصیه‌گر است: (۱) در سیستم‌های توصیه‌گر رایج، که معمولاً روی گراف دو بخشی کاربر (در اینجا پین) و محصول (در اینجا برد) کار می‌کنند، برای هر کاربر (پین)، چندین محصول (برد) پیشنهاد می‌شود، اما در سیستم توصیه‌گر مورد نیاز برای پیترست، باید برای هر پین (کاربر)، چندین پین (کاربر) مشابه پیشنهاد شود. (۲) در سیستم‌های توصیه‌گر رایج، برای هر پین، یک مجموعه از بوردها پیشنهاد می‌شوند اما در این مقاله برای یک مجموعه از پین‌ها، یک مجموعه از پین‌های مشابه پیشنهاد می‌شود. با توجه به تفاوت‌هایی که بیان شد، معیارهای ارزیابی موجود برای سیستم‌های توصیه‌گر رایج، روی سیستم توصیه‌گر این مقاله قابل ارزیابی نیستند.

^{۱۵} <https://colab.research.google.com/>

^{۱۳} category

^{۱۴} jaccard index

معیار پیش‌بینی لینک و با استفاده از معیار $\text{recall}@k$ مقایسه می‌شوند.

۵,۱ معیارهای ارزیابی و روش مقایسه

الگوریتم pixie دارای سرعت بالایی در ارائه توصیه به کاربران با توجه به اولویت و علاقمندی‌های آن‌ها می‌باشد. به عنوان یکی از پارامترهای موثر بر سرعت بالای این الگوریتم می‌توان به ویژگی توقف زودهنگام اشاره کرد. توقف زود هنگام به دو شیوه در الگوریتم pixie اجرا شده است: (۱) استفاده از N_q به جای N (۲) استفاده از دو مقدار n_p و n_v بدین گونه که اگر حداقل تعداد n_p پین در قدم‌زنی تصادفی حداقل n_v بار دیده شوند، قدم‌زنی پایان یابد. شرط پایان یافتن قدم‌زنی تصادفی در الگوریتم Newpixie نیز مشابه الگوریتم pixie است.

برای ارزیابی و مقایسه سرعت الگوریتم‌های pixie و Newpixie ، از جمع کل گام‌های برداشته شده در قدم‌زنی‌های تصادفی برای هر مجموعه پرس و جو استفاده می‌شود. هر چه تعداد گام‌های برداشته شده برای رسیدن به بهترین توصیه‌ها متناسب با یال‌های مجازی بین پین‌ها کمتر باشد، الگوریتم سرعت بیشتری دارد.

معیار بعدی ارزیابی این دو الگوریتم، کیفیت توصیه‌های پیشنهاد شده به کاربران است. برای این منظور، در ابتدا، با توجه به نام دسته‌ای که به هر مورد اختصاص داده شده است، به پین‌های موجود در آن مورد نیز همان نام دسته اختصاص داده می‌شود. از آنجا که هر پین ممکن است در چند مورد مختلف حضور داشته باشد پس هر پین ممکن است در چندین دسته مختلف نیز باشد. حال برای هر کدام از مجموعه پین‌های تولید شده توسط مجموعه پرس و جو و الگوریتم پایه و الگوریتم توسعه یافته قدم‌زنی تصادفی یک مجموعه مورد ایجاد خواهد شد. برای مقایسه این سه مجموعه مورد از شاخص جاکارد بصورت زیر استفاده می‌شود:

$$\text{jacard} = \frac{A \cap B}{A \cup B} \quad (۵)$$

به این شکل که در ابتدا شاخص جاکارد را برای مجموعه بوردهای بدست آمده از مجموعه پین‌های پرس و جو و الگوریتم pixie پایه محاسبه کرده و سپس با مقدار شاخص جاکارد مربوط به مجموعه بوردهای بدست آمده از مجموعه پین‌های پرس و جو و الگوریتم pixie توسعه یافته مورد مقایسه قرار می‌دهد.

در سیستم توصیه‌گر این مقاله، برای هر پین p ، تعداد دفعات مشاهده شده در قدم‌زنی‌های تصادفی ($v[p]$)، ذخیره می‌شود و در نهایت، k پین با بالاترین $v[p]$ پیشنهاد می‌شوند. برای اینکه سیستم توصیه‌گر بتواند مورد را نیز پیشنهاد دهد. بوردهای b مرتبط با هر پین p پیشنهادی، شناسایی شده و تعداد دفعات مشاهده هر مورد ($z[p]$) با توجه به $x[p]$ وزن می‌گیرد. در نهایت، k تا از بوردهایی که بالاترین $z[p]$ را دارند توصیه می‌شوند.

یکی از روش‌های رایج برای ارزیابی سیستم‌های توصیه‌گر، پیش‌بینی پیوند است. بدین گونه که، درصدی از پیوند (یال‌های) بین کاربر و محصول، در گراف، حذف می‌شوند و سیستم توصیه‌گر باید این پیوندهای حذف شده را به درستی به کاربر توصیه کند. یکی از پر استفاده ترین معیارهای ارزیابی در روش پیش‌بینی پیوند، $\text{recall}@k$ است که از رابطه زیر محاسبه می‌شود:

$$\text{recall}@k = \frac{|P_Q \cap R_Q|}{|P_Q|} \quad (۶)$$

در رابطه ۶، R_Q ، مجموعه بوردهای توصیه شده توسط سیستم توصیه‌گر، برای مجموعه پین‌های پرس و جو Q می‌باشد و P_Q ، مجموعه بوردهای حذف شده برای پین‌های مجموعه Q می‌باشد. با توجه به اینکه روش pixie کاملاً برخط بوده و الگوریتم Newpixie شامل دو مرحله برون‌خط و برخط است؛ لذا در مرحله برون‌خط الگوریتم Newpixie ، با بهره‌گیری از قوانین وابستگی در داده کاوی، جفت پین‌های دارای وابستگی قوی با استفاده از FPGrowth استخراج خواهند شد.

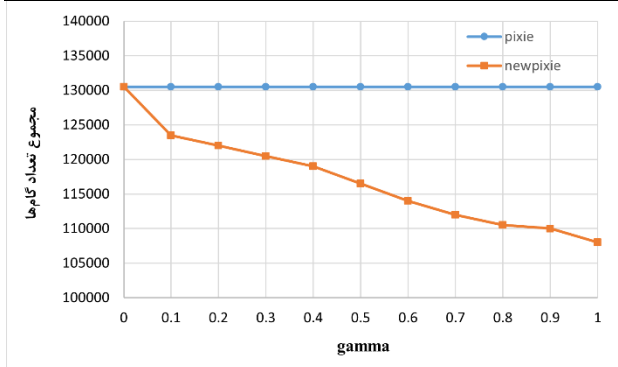
می‌یابد. با توجه به اینکه مقدار مجموع تعداد گام‌های طی شده و سرعت اجرای الگوریتم نسبت عکس دارند، لذا این بدین معناست که با افزایش مقدار γ ، مجموع تعداد گام‌های طی شده توسط الگوریتم **Newpixie**، کاهش یافته و در نتیجه، سرعت این الگوریتم، به تدریج افزایش می‌یابد. سرعت **Newpixie** در زمانی که مقدار γ برابر با ۱ باشد به بیشینه مقدار خود می‌رسد که تقریباً ۱۷٪ بیشتر از سرعت الگوریتم **pixie** است.

برای مقایسه دو الگوریتم **pixie** و **Newpixie**، ابتدا یک مجموعه پین پرس و جو بصورت تصافی انتخاب شده و برای هر پین، یک وزن، متنظر با آن نیز در نظر گرفته می‌شود به طوری که پین‌های با الویت بالاتر، وزن بیشتری دارند.

از آن‌جا که روش قدم زنی تصادفی در هر بار اجرا، دنباله متفاوتی از پین‌ها را استخراج می‌کند، لذا برای مقایسه دو الگوریتم **pixie** و **Newpixie** دنباله یکسانی از پین‌ها انتخاب می‌شود. سپس یک دنباله قدم زنی تصادفی به طول N روی گراف دوبخشی G ایجاد شده و هر الگوریتم روی آن اجرا می‌شود. به این صورت، امکان مقایسه دو الگوریتم روی دنباله یکسانی از پین‌ها ایجاد خواهد شد.

جدول ۱ معرفی پارامترها و مقدارشان

مقدار	پارامتر	
۵۰۰۰۰۰	N	پارامترهای مشابه دو الگوریتم
۵۰	n_p	
۲۰۰	n_v	
۰٫۳	α	
۰٫۰۰۰۲	min_sup	پارامترهای استفاده شده فقط در Newpixie
۰٫۶	min_conf	



شکل ۲ نمودار ارزیابی سرعت توصیه‌ها

۵٫۳ آزمایش دوم: ارزیابی کیفیت بر اساس γ
پس از ارزیابی سرعت دو الگوریتم، کیفیت توصیه‌های ارائه شده توسط هر کدام از الگوریتم‌های **pixie** و **Newpixie** در این مقاله مورد بررسی قرار گرفته است.

جدول ۱، پارامترهای استفاده شده در آزمایش‌ها و مقدارشان را نمایش می‌دهد. همانگونه که مشاهده می‌شود پارامترهای مشابه استفاده شده در هر دو الگوریتم شامل N ، تعداد کل قدم‌زنی‌های تصادفی، n_p و n_v ، پارامترهای مورد استفاده برای توقف زود هنگام و α ، طول هر گام در قدم‌زنی تصادفی، هستند. پارامترهای \min_conf و \min_sup ، به ترتیب نمایانگر حداقل پشتیبان و حداقل اطمینان هستند و در مرحله برون‌خط الگوریتم **Newpixie** و توسط الگوریتم **FPGrowth** استفاده می‌شوند.

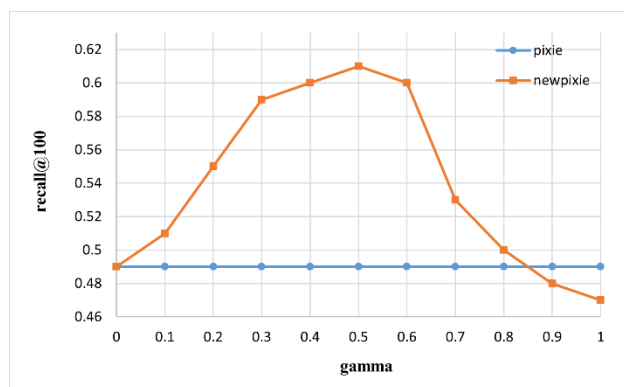
۵٫۲ آزمایش اول: ارزیابی سرعت بر اساس γ

در این آزمایش مجموع تعداد گام‌های برداشته شده برای ایجاد مجموعه پین‌های مناسب برای توصیه، نسبت به مقدار γ بررسی شده است.

با توجه به نمودار شکل ۲، مشاهده می‌شود که به دلیل وابسته نبودن الگوریتم **pixie** به پارامتر γ ، با افزایش این معیار، تعداد گام‌های طی شده توسط الگوریتم **pixie** ثابت می‌ماند ولی تعداد گام‌های طی شده توسط الگوریتم **Newpixie** کاهش

شکل ۴، نمودار مقایسه نتایج عملیات پیش‌بینی پیوند، با استفاده از معیار ارزیابی $\text{recall}@100$ روی الگوریتم‌های pixie و Newpixie را نشان می‌دهد.

همانگونه که مشاهده می‌شود، با توجه به اینکه مقدار gamma تأثیری روی الگوریتم pixie ندارد، لذا مقدار $\text{recall}@100$ آن در طول نمودار، ثابت مانده است. اما برای الگوریتم Newpixie، مشاهده می‌شود که با افزایش مقدار gamma و رسیدن آن به مقدار ۰,۶، نمودار یک روند صعودی را طی کرده‌است و مقدار $\text{recall}@100$ را تقریباً تا ۲۰٪ بهبود داده‌است. با عبور gamma از مقدار ۰,۶، نمودار یک روند نزولی را طی کرده‌است و در مقادیر ۰,۹ و ۱ نتایج ضعیفتری را نسبت به pixie کسب کرده است.



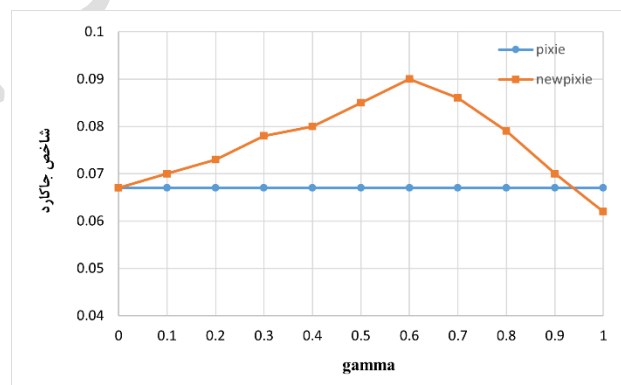
شکل ۴ نمودار مقایسه پیش‌بینی پیوند

۶ نتیجه‌گیری

برقراری تعادل بین سرعت و کیفیت توصیه‌های ارائه‌شده به کاربر از چالش‌های اصلی سیستم‌های توصیه‌گر است. در این مقاله الگوریتمی بر مبنای الگوریتم قدم‌زنی تصادفی پایه pixie، به نام الگوریتم قدم‌زنی تصادفی Newpixie معرفی شده است که هدف آن افزایش سرعت ارائه توصیه‌ها با توجه به در نظر گرفتن کیفیت آن‌هاست. این الگوریتم در دو مرحله اجرا می‌شود: (۱) مرحله برون‌خط (۲) مرحله برخط. در مرحله برون‌خط، با استفاده از پیش‌پردازش و بررسی قوانین داده کاوی در مجموعه‌های دوتایی از پین‌های موجود در بوردها، جفت پین‌هایی که در بوردهای زیادی تکرار شده‌اند را به عنوان همسایه‌های مجازی در نظر گرفته و یک ارتباط فرضی بین آن‌ها

با توجه به اینکه تغییرات مقدار gamma در الگوریتم pixie تأثیری ندارد؛ لذا در شکل ۳ مشاهده می‌شود که با افزایش مقدار این معیار، الگوریتم pixie مشمول هیچ تغییری در مقدار شاخص جاکارد خود نخواهد شد؛ در حالی که مقدار شاخص جاکارد برای الگوریتم Newpixie در نقطه gamma برابر ۰,۶ به نقطه اوج خود می‌رسد، که نسبت به الگوریتم pixie، ۳۴٪ افزایش یافته است. در ادامه، با افزایش gamma ، مقدار شاخص جاکارد به تدریج کاهش یافته و در نهایت در مقدار gamma برابر با ۱، کمتر از الگوریتم pixie می‌شود.

با توجه به نتایج این بخش، چنین برداشت می‌شود که گرچه با افزایش مقدار گاما، سرعت الگوریتم قدم‌زنی توسعه داده شده در این مقاله افزایش می‌یابد، اما با عبور از مقدار گامای ۰,۶، کیفیت توصیه‌های پیشنهاد شده به کاربر افت پیدا خواهد کرد.



شکل ۳ نمودار ارزیابی کیفیت توصیه‌ها

۵,۴ آزمایش سوم: ارزیابی پیش‌بینی پیوند

برای بالابردن اعتبار آزمایش‌های انجام شده و نشان دادن کارایی الگوریتم پیشنهادی، از معیار ارزیابی $\text{recall}@k$ استفاده شده‌است. این معیار، یکی از پرکاربردترین معیارهای ارزیابی سیستم‌های توصیه‌گر است که روی عملیات پیش‌بینی پیوند محاسبه می‌شود. برای این منظور، تغییری در الگوریتم Newpixie داده شد تا در این الگوریتم، پس از دریافت مجموعه پین‌های پرس‌وجوی Q ، علاوه بر توصیه کردن یک مجموعه از پین‌ها، یک مجموعه از بوردها نیز توصیه شود تا عملیات پیش‌بینی پیوند، مورد ارزیابی قرار گیرد.

[8] C. C. Aggarwal, "Recommender Systems," Cham: Springer International Publishing, 2016. doi: 10.1007/978-3-319-29659-3.

[9] D. Putri, N. Yuhandri, and R. Hardianto, "Application of the FP-Growth algorithm in consumer purchasing pattern analysis," *UPI YPTK Journal of Computer Science and Information Technology*, pp. 44–49, Apr. 2024, doi: 10.35134/jcsitech.v10i2.99.

[10] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," *SIGMOD Record*, vol. 29, no. 2, pp. 1–12, May 2000, doi: 10.1145/335191.335372.

[11] D. Agarwal et al. "Personalizing LinkedIn Feed," *In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '15)*. ACM, New York, NY, USA, pp. 1651–1660, Aug. 2015, doi: 10.1145/2783258.2788614.

[12] P. Covington, J. Adams, and E. Sargin. "Deep Neural Networks for YouTube Recommendations," *In Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)*. ACM, New York, NY, USA, pp. 191–198, Sep. 2016, doi: 10.1145/2959100.2959190.

[13] A. Goel, P. Gupta, J. Sirois, D. Wang, A. Sharma, and S. Gurumurthy, "The Who-To-Follow System at Twitter: Strategy, algorithms, and revenue impact," *Interfaces*, vol. 45, no. 1, pp. 98–107, Feb. 2015, doi: 10.1287/inte.2014.0784.

در گراف، برقرار می‌کند. در مرحله برخط، با استفاده از گراف تغییر یافته در مرحله برون‌خط و با اجرای قدم‌زنی تصادفی، با سرعت، توصیه‌هایی با کیفیت مناسب را به کاربر ارائه می‌دهد. سه آزمایش مختلف روی Newpxie انجام شده است. نتایج نشان می‌دهد که Newpixie نتایج بهتری را از نظر سرعت و دقت، نسبت به الگوریتم pixie داشته است.

منابع

۷

[1] X. Li, L. Sun, M. Ling, and Y. Peng, "A survey of graph neural network based recommendation in social networks," *Neurocomputing*, vol. 549, p. 126441, 2023, doi: 10.1016/j.neucom.2023.126441.

[2] Z. Fayyaz, M. Ebrahimian, D. Nawara, A. Ibrahim, and R. Kashef, "Recommendation systems: algorithms, challenges, metrics, and business opportunities," *Applied Sciences*, vol. 10, no. 21, p. 7748, Nov. 2020, doi: 10.3390/app10217748.

[3] C. Eksombatchai, P. Jindal, JZ. Liu, Y. Liu, R. Sharma, C. Sugnet, M. Ulrich, and J. Leskovec. "Pixie: A system for recommending 3+ billion items to 200+ million users in real-time." *In Proceedings of the 2018 world wide web conference*, pp. 1775–1784, Jan. 2018, doi: 10.1145/3178876.3186183.

[4] Y. Li, K. Liu, R. Satapathy, S. Wang, and E. Cambria, "Recent Developments in Recommender Systems: A survey [Review article]," *IEEE Computational Intelligence Magazine*, vol. 19, no. 2, pp. 78–95, May 2024, doi: 10.1109/mci.2024.3363984.

[5] S. M. Vahidipour, A. Mohamadi, "Link prediction in scientific networks using machine learning and weighted graphs," *Soft Computing Journal*, Jan. 2024, doi: 10.22052/scj.2024.253476.1181 [In Persian]

[6] S. M. Vahidipour, R. Karami, "Link prediction framework using graph neural network based on subgraph," *Soft Computing Journal*, Jan. 2024, doi: 10.22052/scj.2024.253458.1179 [In Persian]

[7] M. Shams, GH. Hesamian, "Random walk on connected graphs and its application in electronic networks," *Soft Computing Journal*, Oct. 2023, doi: 10.22052/scj.2024.248446.1106 [In Persian]