سال هفتم، شماره‌های ۲۵ و ۲۶، پاییز و زمستان ۱۳۹۴ فصلنامه علمی- پژوهشی

صص: ۶۷- ۸۶ فناوری اطلاعات و ارتباطات ایران

ارائهٔ الگوریتمی مبتنی بر یادگیری جمعی به منظور یادگیری رتبه‌بندی در بازیابی اطلاعات

الهام قنبری* آزاده شاکری**

*دانشجو دکتری، دانشکده مهندسی برق و کامپیوتر، پردیس دانشکده‌های فنی دانشگاه تهران، تهران

**استادیار، دانشکده مهندسی برق و کامپیوتر، پردیس دانشکده‌های فنی دانشگاه تهران، تهران

تاریخ دریافت:۹۳/۰۱/۲۴ تاریخ پذیرش:۹۴/۰۳/۲۵

چکیده

یادگیری رتبه‌بندی که یکی از روش‌های یادگیری ماشین برای مدل کردن رتبه‌بندی است، امروزه کاربردهای بسیاری به خصوص در بازیابی اطلاعات، پردازش زبان طبیعی و داده‌کاوی دارد. فعالیت یادگیری رتبه‌بندی را می‌توان به دو بخش تقسیم کرد. یکی سیستم یادگیری مورد استفاده و دیگری سیستم رتبه‌بندی. در سیستم یادگیری، یک مدل رتبه‌بندی بر اساس داده‌های ورودی ساخته می‌شود. در بخش سیستم رتبه‌بندی، از این مدل ساخته شده برای پیش‌بینی رتبه‌بندی استفاده می‌شود. در این مقاله یک الگوریتم مبتنی بر یادگیری جمعی به منظور یادگیری رتبه‌بندی اسناد پیشنهاد می‌شود که به صورت تکراری یادگیرهای ضعیفی بر روی درصدی از داده‌های آموزشی که توزیع آنها بر اساس یادگیر قبلی تغییر یافته است، می‌سازد و جمعی از یادگیرهای ضعیف را برای رتبه‌بندی تولید می‌کند. این الگوریتم سعی می‌کند با ساختن رتبه‌بند بر روی درصدی از داده‌ها، دقت را افزایش و زمان آموزش را کاهش دهد. با ارزیابی بر روی مجموعه داده LETOR3 دیده می‌شود که الگوریتم پیشنهادی بهتر از الگوریتم‌های دیگری در این زمینه که مبتنی بر یادگیری جمعی هستند، عمل می‌کند.

واژه‌های کلیدی: یادگیری در ایجاد رتبه‌بندی، یادگیری رتبه‌بندی در بازیابی اطلاعات، یادگیری ماشین، یادگیری جمعی

۱- مقدمه

مسائل رتبه‌بندی امروزه در پژوهش‌ها از جایگاه ویژه‌ای برخوردارند و کاربردهای متنوعی به خصوص در بازیابی اطلاعات و موتورهای جستجو دارند. مسائل رتبه‌بندی را می‌توان به دو دسته کلی تقسیم کرد[1]

۱. ایجاد رتبه‌بندی[1]: برای یک درخواست وارد شده، یک لیست مرتب‌شده از پیشنهادات مبتنی بر خصوصیات آن درخواست ارائه می‌شود.

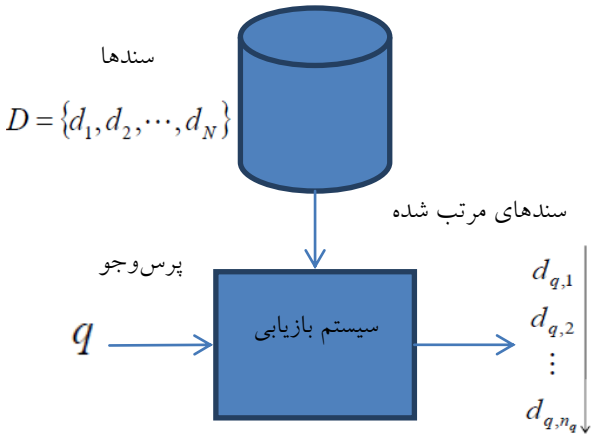
۲. تجمیع رتبه‌بندی[2]: برای یک درخواست وارد شده، از چندین لیست مرتب شده از پیشنهادات استفاده کرده و یک لیست مرتب شده جدید از پیشنهادات ارائه می‌شود.

بازیابی اطلاعات جزو مسائل دسته اول، یعنی ایجاد رتبه‌بندی محسوب می‌شود. در بازیابی اطلاعات، با ورود هر پرس‌وجوی q از طرف کاربر،سیستم بازیابی اسنادی که

1. Ranking creation

2. Ranking aggregation

نویسندهٔ عهده‌دار مکاتبات: آزاده شاکری shakery@ut.ac.ir



شکل۱: مراحل بازیابی اطلاعات [1]

مرتبط با پرس‌وجو هستند را از مجموعه D استخراج می‌کند. منظور از اسناد مرتبط، سندهایی هستند که شامل کلمات موجود در پرس‌وجو می‌باشند. بعد از استخراج این اسناد، سیستم بازیابی آن‌ها را رتبه‌بندی می‌کند و درصدی از اسناد با رتبه بالا را به عنوان خروجی به کاربر نشان می‌دهد. مراحل عنوان شده در شکل ۱ دیده می‌شود.

سیستم بازیابی عنوان شده، به چند طریق می‌تواند شبیه‌سازی شود. یک روش استفاده از روش‌های سنتی بازیابی اطلاعات که به روش‌های غیر یادگیری معروفند، مانند BM25 و یا مدل‌های زبانی می‌باشد و روش دیگر که امروزه مورد توجه بسیاری از پژوهشگران قرار گرفته است، استفاده از الگوریتم‌های یادگیری ماشین برای رتبه‌بندی است. روش‌های دسته دوم سعی می‌کنند بر اساس مجموعه‌ای از داده‌های آموزشی که نشان دهنده رتبه‌بندی مجموعه‌ای از درخواست‌ها است، یک مدل رتبه‌بندی برای مرتب‌سازی درخواست‌های جدید ارائه کنند. این روش‌ها تحت عنوان یادگیری رتبه‌بندی دسته‌بندی می‌شوند. برای یادگیری رتبه‌بندی دو تعریف وجود دارد. در تعریف عام، هر روش موجود در یادگیری ماشین برای رتبه‌بندی را یادگیری رتبه‌بندی می‌نامند[1]. در تعریف خاص، هر روش یادگیری ماشین که برای ساختن یک مدل رتبه‌بندی به منظور ایجاد رتبه‌بندی و یا تجمیع آن مورد استفاده قرار گیرد را یادگیری رتبه‌بندی می‌نامند[1]

در یادگیری رتبه‌بندی، بخش ایجاد رتبه‌بندی از اهمیت بیشتری برخوردار است، به گونه‌ای که اکثر مطالعات انجام شده در این بخش صورت گرفته است و اکثر الگوریتم‌های پیشنهادی مبتنی بر یادگیری با نظارت[3] هستند.

الگوریتم‌هایی که تا کنون برای یادگیری رتبه‌بندی ارائه شده‌اند، به سه بخش روش‌های مبتنی بر نقطه[4]، مبتنی بر جفت[5] و مبتنی بر لیست[6] تقسیم می‌شوند [1]روش‌های مبتنی بر نقطه و جفت، مسائل رتبه‌بندی را با تغییر در نحوه داده‌ها به مسائل دسته‌بندی تبدیل می‌کنند، در حالی که روش‌های مبتنی بر لیست، بدون تغییری در داده‌های ورودی، با استفاده از بهینه کردن یک تابع هدف سعی در رتبه‌بندی پرس‌وجوی جدید می‌نمایند.

در این مقاله الگوریتمی جدید در بخش ایجاد رتبه‌بندی ارائه شده است. این الگوریتم در دسته روش‌های مبتنی بر لیست قرار می‌گیرد و سعی می‌کند با کمینه کردن حد بالایی از خطای ایجاد شده به وسیله تعریف تابع هدف، به دقت بالا در رتبه‌بندی دست پیدا کند. الگوریتم جدید مشابه الگوریتم AdaRank0 [2] به صورت تکراری یادگیرهایی ضعیف بر روی داده‌های آموزشی که توزیع آنها بر اساس یادگیر قبلی تغییر یافته است، می‌سازد تا بتواند در مجموع

3. Supervised learning
4. Pointwise
5. Pairwise
6. Listwise

تابع خطا را کمینه کند. ایده جدید روش ارائه شده در این مقاله این است که در ساخت یادگیرهای ضعیف، از تمام مجموعه پرس‌وجو و مستندات مرتبط به آنها استفاده نمی‌شود، بلکه درصدی از آنها که وزن بالاتری دارند مورد استفاده قرار می‌گیرند. استفاده از درصدی از داده‌ها علاوه بر این که باعث افزایش دقت می‌شود، سبب می‌شود زمان اجرای الگوریتم به طور قابل ملاحظه‌ای کاهش پیدا کند و نیاز نباشد در هر تکرار، یادگیری بر روی تمام پرس‌وجوها انجام شود. با بررسی انجام شده بر روی هفت مجموعه داده در LETOR3 [3]دیده می‌شود که الگوریتم پیشنهادی هم از لحاظ دقت و هم از لحاظ سرعت نسبت به الگوریتم-های قبلی بهتر عمل می‌کند.

در ادامه مقاله در بخش دوم ابتدا مروری بر روی الگوریتم-های یادگیری رتبه‌بندی انجام می‌شود، سپس در بخش سوم و چهارم الگوریتم پیشنهادی به همراه نتایج پیاده‌سازی و ارزیابی این الگوریتم ارائه می‌شود. در بخش پنجم نتیجه‌گیری و کارهای آتی ارائه خواهند شد.

۲- مروری بر کارهای گذشته

فعالیت یادگیری رتبه‌بندی به دو سیستم یادگیری و سیستم رتبه‌بندی تقسیم می‌شود [1] در سیستم یادگیری، با بهره‌گیری از مجموعه‌ای از پرس‌وجوها و اسناد مرتبط به آن‌ها، یک مدل رتبه‌بند ساخته می‌شود و در سیستم رتبه‌بندی از این مدل برای رتبه‌بندی اسناد برای پرس‌وجوهای جدید بهره برده می‌شود.

در سیستم یادگیری، مجموعه پرس‌وجوهای $\mathcal{Q}$ و اسناد $\mathcal{D}$ به صورت داده آموزشی به شکل $S = \{(q_i, \mathcal{D}_i), y_i\}_{i=1}^{m}$ وارد سیستم می‌شوند. به عبارتی هر پرس‌وجوی $q_i \in \{q_1, q_2, \dots, q_m\}$ به همراه اسناد مرتبط $\mathcal{D}_i = \{d_{i,1}, d_{i,2}, \dots, d_{i,ni}\}$ به عنوان یک داده آموزشی محسوب می‌گردد. در این رابطه y_i برابر با درجه ارتباط اسناد $\mathcal{D}_i$ به پرس‌وجوی q_i است، که در ساده‌ترین حالت این ارتباط به صورت برچسب "مرتبط" یا "نامرتبط" مشخص می‌گردد. در این مرحله برای ساخت یک مدل یادگیر، از زوج پرس‌وجو و اسناد مرتبط $(q_i, \mathcal{D}_i)$، بردار ویژگی‌های x_i استخراج می‌شود. به عبارتی داده‌های آموزشی به شکل $\acute{S} = \{x_i, y_i\}_{i=1}^{m}$ تغییر می‌یابد. در انتها بر اساس این داده‌ها مدل رتبه‌بندی ساخته می‌شود.

در سیستم رتبه‌بندی، بر اساس مدل ساخته شده، مجموعه مستندات مرتبط D_{m+1} در پاسخ به برای پرس‌وجوی جدید q_{m+1} رتبه‌بندی می‌شوند.

در بخش یادگیری، اکثر الگوریتم‌های یادگیری ماشین که در دسته‌بندی داده‌ها مورد استفاده قرار می‌گیرند می‌توانند در یادگیری رتبه‌بندی مورد استفاده واقع شوند. یکی از مهم‌ترین الگوریتم‌های یادگیری در رتبه‌بندی که در هر سه دسته الگوریتم‌های مبتنی بر نقطه، جفت و لیست کاربرد دارد، استفاده از الگوریتم‌های یادگیری جمعی است؛ یعنی برای دسته‌بندی داده‌ها از چندین دسته‌بند و ترکیب آنها استفاده می‌کنند. این الگوریتم‌ها به علت بهره‌گیری از چندین دسته‌بند، در اکثر مواقع نتایج دقیق‌تر و مقاوم‌تری تولید می‌کنند، لذا استفاده از این الگوریتم‌ها در یادگیری رتبه‌بندی مورد توجه واقع شده است.

در ادامه به معرفی برخی از الگوریتم‌های مهم در یادگیری رتبه‌بندی در هر سه دسته موجود (مبتنی بر نقطه، جفت و لیست) با تمرکز بر روی یادگیری جمعی پرداخته می‌شود.

۲-۱- روش‌های مبتنی بر نقطه

یکی از ساده‌ترین روش‌ها در رتبه‌بندی اسناد مرتبط به پرس‌وجو، استفاده مستقیم از روش‌های موجود در یادگیری ماشین است. این روش‌ها، تحت عنوان روش‌های مبتنی بر نقطه شناخته می‌شوند. در این دسته روش‌ها، مسائل رتبه-بندی به مسائل دسته‌بندی و یا رگرسیون تبدیل می‌شوند. خوبی این روش در این است که روش‌های شناخته شده بسیاری در زمینه دسته‌بندی و رگرسیون موجود است که می‌تواند مورد استفاده قرار گیرد.

یکی از نقاط ضعف این روش‌ها در این است که فرض گروه‌بندی اسناد برای یک پرس‌وجو نادیده گرفته می‌شود. به عبارتی در این روش‌ها هدف تشخیص درجه ارتباط هر سند به پرس‌وجو مستقل از اسناد دیگر است.

در [6]مسائل رتبه‌بندی به صورت دسته‌بندی چندکلاسه در نظر گرفته شده است و الگوریتم McRank بر این مبنا

ارائه شده است. در این الگوریتم اسناد به صورت منفرد در کلاس‌های مختلف که در ساده‌ترین حالت دو کلاس "مرتبط" و "نامرتبط" می‌باشد، قرار می‌گیرند. در این الگوریتم از معیار DCG برای ارزیابی استفاده می‌شود، به این صورت که دسته‌بند مناسب، دارای امتیاز بالایDCG خواهد بود. در این روش از ایده امید ریاضی برای تبدیل احتمال کلاس‌ها به امتیاز رتبه‌بندی استفاده می‌شود. احتمالات برای هر کلاس توسط الگوریتم درخت بوستینگ در امتداد گرادیان محاسبه می‌گردد.

در [7]مطالعه بر روی دسته‌بندی به منظور تخصیص امتیازدهی مناسب به هر شی ورودی انجام گرفته است. از این امتیازدهی می‌توان برای رتبه‌بندی نیز استفاده نمود. این الگوریتم با نام Prank، الگوریتمی برخط است که رتبه‌بند خود را براساس مدل‌های موازی پرسپترون[۷] می‌سازد. هر مدل توانایی جدا نمودن امتیازات دو همسایگی را از یکدیگر دارا می‌باشد. تعداد همسایگی‌ها در این روش همان تعداد درجه ارتباط اسناد است، که در ساده‌ترین حالت اسناد در دو همسایگی مرتبط یا نامرتبط قرار می‌گیرند.

الگوریتم MART الگوریتم دیگری در این حوزه از خانواده الگوریتم‌های یادگیری جمعی است، که خروجی آن یک ترکیب خطی وزن‌دار از مجموعه درخت‌های رگرسیون می‌باشد. هر درخت رگرسیون با هدف کمینه کردن تابع خطا در امتداد کاهش گرادیان ساخته می‌شود[11].

الگوریتم OC SVM[۸] یک روش مبتنی بر ماشین بردار پشتیبان با حاشیه[۹] بزرگ برای دسته‌بندی ترتیبی است[4]در این الگوریتم سعی می‌شود ابرصفحاتی[۱۰] به صورت موازی برای جداسازی امتیازدهی همسایه‌ها یاد گرفته شود. هر همسایگی به صورت یک درجه ارتباط بین اسناد در نظر گرفته می‌شود. برای تعریف حاشیه مناسب بین همسایگی‌ها، دو روش در نظر گرفته شده است: یکی در نظر گرفتن حاشیه مساوی برای امتیازدهی به همسایه‌ها و بیشینه کردن این حاشیه و دیگری در نظر گرفتن حاشیه‌های متفاوت برای هر همسایه و بیشینه کردن مجموع این حاشیه‌ها.

۲-۲- روش‌های مبتنی بر جفت

در روش‌های مبتنی بر جفت، هدف یافتن ترتیب بین هر دو سند برای یک پرس‌وجو است. این روش‌ها بین هر جفت سد با دادن امتیاز بیشتر به سندی که مرتبط‌تر است، اسناد را رتبه‌بندی می‌کنند. برای مثال فرض کنید برای یک پرس‌وجو، سه سند d_1 ، d_2 و d_3 موجود باشد. در این روش‌ها ترتیب دو به دو اسناد مشخص می‌شود. اگر این ترتیب به صورت $d_1 > d_2$ ، $d_1 > d_3$ و $d_2 < d_3$ تشخیص داده شود، رتبه‌بندی نهایی به صورت $d_1 > d_3 > d_2$ بدست می‌آید.

در الگوریتم‌های مبتنی بر جفت، رایج‌ترین تابع خطا به صورت ترتیب نادرست دو به دو اسناد در نظر گرفته می‌شود. یکی از اولین الگوریتم‌های موجود در این زمینه برای یادگیری رتبه‌بندی، RankingSVM است.[8] در این الگوریتم برای دو به دو اسناد یک دسته‌بند بر اساس ماشین بردار پشتبان ساخته می‌شود. سپس از این دسته‌بندها برای رتبه‌بندی اسناد به صورت جفت بهره برده می‌شود.

الگوریتم RankNet با بهره‌گیری از یک تابع احتمالی ساده به نام آنتروپی متقابل[۱۱]برای محاسبه تابع خطا و استفاده از گرادیان در امتداد نزولی[۱۲] سعی می‌کند تا یک مدل بهینه مبتنی بر شبکه عصبی طراحی می‌کند[9]این الگوریتم به عنوان اولین الگوریتم مبتنی بر یادگیری رتبه‌بندی مورد استفاده موتورهای جستجو واقع شده است.

الگوریتم LambdaRank مشابه با الگوریتم RankNet تعریف می‌شود که از شبکه عصبی برای ساخت مدل رتبه‌بند خود استفاده می‌کند. این الگوریتم گرادیان تابع خطا را با عنوان تابع Lambda تعریف می‌کند. این تابع بر اساس تغییر رتبه اسناد در یک لیست مرتب به منظور بهینه کردن کارایی رتبه‌بند تعریف می‌شود. به عبارتی این روش برای بهینه کردن تابع lambda از شبکه عصبی بهره می‌برد[10]

7. Perceptron
8. Ordinal Classification SVM
9. Margin
10. Hyperplanes
11. CrossEntropy
12. Gradient Descent

الگـــوریتم LambdaMART در راســـتای الگـــوریتم LambdaRank ارائه شد. این الگـوریتم بـه جـای سـاخت مـدل رتبـه‌بنـد از طریـق شـبکه عصـبی، از روش درختـان بوستیگ و یا بـه عبـارتی از الگـوریتم MART بهـره مـی-بـرد[12].کـارایی و دقـت ایـن روش در مقایسـه بـا روش LambdaRank بالاتر است.

[13] با استفاده از درخت بوستیگ در امتداد گرادیان سـعی می‌کند در هر مرحله درختی بسازد کـه کمتـرین خطـا را از نظر مشخص کردن ترتیب بین دو سند داشته باشـد. سـپس این مدل به صورت خطی با مدل‌های قبلی ترکیب می‌شـود. در مرحله بعد از مدل ساخته شده برای یافتن ترتیب مجـدد بین اسناد استفاده می‌شود و تابع خطا محاسبه می‌شود. ایـن روند به صورت تکراری صورت می‌گیـرد تـا در نهایـت مـدل رتبه‌بند نهایی ساخته شود.

الگـوریتم دیگـر، الگـوریتم RankBoost اسـت [14]کـه مبتنی بر روش بوستینگ عمل می‌کند. این الگوریتم در هـر مرحله سعی می‌کند رتبه‌بند ضعیفی بسازد که کمترین خطا را از نظر تشخیص نادرست ترتیب بین جفـت اسـناد داشـته باشد. بدین منظور در هر مرحله وزن اسناد را به صورت دوبه دو تغییر می‌دهد. سپس با ترکیب این رتبه‌بندها، مدل کلـی ساخته می‌شود. رتبه‌بند ضعیف در هر مرحلـه مـی‌توانـد بـر اساس انتخاب مقدار بهترین ویژگی و یـا انتخـاب حـدی بـر روی ویژگی‌ها منظور شود.

الگـوریتم Ranking Forest الگـوریتمی اسـت کـه بـرای رتبه‌بندی ارائـه شـده اسـت و نـوعی از الگـوریتم یـادگیری جمعی است که نتایج مجموعه‌ای از درختـان تصـادفی را بـا هم ترکیب می‌کند. این الگـوریتم عمـل نمونـه‌گیـری بـرای ساخت رتبه‌بند را به صورت بگینگ انجام می‌دهد تـا ارتبـاط بین دو درخت رتبه‌بند را کاهش دهد[15]

چهارالگوریتم انتهایی، الگوریتم‌هـایی هسـتند کـه در دسـته الگوریتم‌های مبتنی بر یادگیری جمعی قرار می‌گیرند.

۲-۳- روش‌های مبتنی بر لیست

در روش‌های مبتنی بر لیست، رتبـه‌بنـدی اسـناد بـرای هـر پرس‌وجو به صورت کامل انجام می‌پـذیرد. در ایـن روش بـر خلاف روش‌های مبتنی بر نقطه و مبتنـی بـر جفـت کـه بـه ترتیب هدف پیش‌بینی ارتباط تک سند و پیش‌بینی ترتیـب بین جفت اسناد است، کل اسناد مرتبط به یک پرس‌وجو بـه عنوان داده آموزشی محسوب می‌شود. لـذا ایـن روش‌هـا بـه سوی بهینه‌سازی مستقیم یکی از معیارهای بازیابی اطلاعات بـر روی تمـام پـرس‌وجوهـا مـی‌رونـد. از ایـن جهـت بـه الگوریتم‌های موجود در این روش‌ها، الگـوریتم‌هـای بهینـه-سازی می‌گویند. البته این روش‌ها مشـکلات خـاص خـود را دارد، به این علت که اکثر معیارهای ارزیابی، توابعی ناپیوسته هستند. به همین دلیل در این دسته از روش‌هـا سـعی مـی-شود تا تخمینی از این توابع ارائه شود و بهینه‌سازی بـر روی آن صورت گیرد.

یکی از مهمترین الگوریتم‌های این دسـته، الگـوریتم SVM MAP [16] است که از بهینه‌سازی مستقیم معیار MAP برای ساخت مدل یادگیری استفاده می‌کند، به ایـن صـورت که با استفاده از تابع HINGE حد بالایی از خطا را کـه بـر اساس معیار MAPمحاسبه شده است بهنیه می‌کنـد. بـرای بهینه سازی این تـابع خطـای جدیـد از الگـوریتم یـادگیری ماشین بردار پشتیبان بهره برده می‌شود. این ایده مـی‌توانـد گسترش پیدا کند، به گونـه‌ای کـه بـرای هـر حـد بـالایی از خطایی که با استفاده از معیارهـای بازیـابی اطلاعـات ایجـاد شود می‌توان از الگوریتم ماشین بردار پشتیبان استفاده نمود.

الگوریتم SoftRank الگـوریتم دیگـری اسـت کـه یـک راه تخمینی (نرم) برای محاسبه توزیع رتبه‌بندی ارائه مـی‌دهـد. به این صورت کـه امتیـاز اسـناد را بـه صـورت یـک متغیـر تصادفی از نوع گوسی در نظر می‌گیرد. توزیع ایـن امتیـازات برای هر سند به عنوان مبنایی برای تعریف معیارهای بازیابی اطلاعات به صورت احتمـالی مـی‌شـود ، بـه گونـه‌ای کـه بـا استفاده از این توزیع می‌توان بـه طـور تخمینـی معیارهـایی مانند NDCG را محاسبه کـرد[17]. خـوبی ایـن روش در این است که در اکثر معیارهـای ارزیـابی کـه توانـایی بهینـه سازی مستقیم را ندارند، می‌توان از این روش استفاده نمود.

الگـوریتم ListNet [18] از یـک مـدل مبتنـی بـر شـبکه عصبی و معیار KL به عنوان تابع خطـا بـرای سـاخت مـدل رتبه‌بندی استفاده می‌کند، به این صورت که در ابتدا احتمال جایگشت‌های رتبه‌بندی و یا به عبارتی احتمال k تـای برتـر یک لیست از اشیا محاسبه می‌شود و سپس با بهـره‌گیـری از

این احتمال و با استفاده از روش KL، تفاوت بین لیست رتبه‌بندی ساخته شده از طریق یادگیری و لیست رتبه‌بندی واقعی محاسبه می‌شود و با روش گرادیان نزولی لیست بهینه بدست می‌آید.

الگوریتم بعدی AdaRank است[2] این الگوریتم مبتنی بر بوستیگ است، به این صورت که به صورت تکراری یادگیرهای ضعیفی بر روی داده‌های آموزشی که توزیع آنها بر اساس یادگیر قبلی تغییر یافته است، می‌سازد و جمعی از یادگیرهای ضعیف را برای رتبه‌بندی تولید می‌کند. یادگیر ضعیف در هر مرحله با انتخاب یک ویژگی که با هدف کمینه کردن تابع خطا (مبتنی بر یکی از معیارهای MAP یا NDCG) صورت می‌گیرد ساخته می‌شود. تفاوت این روش با روش‌های جمعی دسته‌بندی، استفاده از معیارهای ارزیابی به عنوان تابع خطا و روش انتخاب تابع یادگیر ضعیف است.

۳- الگوریتم پیشنهادی

در بین الگوریتم‌های مبتنی بر نقطه، جفت و لیست الگوریتم‌های موجود در دسته مبتنی بر لیست کارایی بهتری دارند. در این مقاله با الهام از الگوریتم‌های موجود در این دسته و با بهره‌گیری از یادگیری جمعی، الگوریتمی ارائه خواهد شد که سعی در یادگیری رتبه‌بندی مستندات دارد.

این الگوریتم الهام گرفته از الگوریتم AdaBoost [19]مبتنی بر بوستیگ و الگوریتم R-AdaBoost[20]می‌باشد. در الگوریتم پیشنهادی مشابه الگوریتم AdaBoost به صورت تکراری یادگیرهای ضعیفی بر روی داده‌های آموزشی که توزیع آنها بر اساس یادگیر قبلی تغییر یافته است، ساخته می‌شود، با این تفاوت که در الگوریتم AdaBoost سعی در ساخت دسته بند است، ولی در الگوریتم پیشنهادی هدف ساخت یک تابع رتبه‌بندی است که بتواند مستندات را در پاسخ به یک پرس‌وجوی خاص مرتب کند. در نهایت با ترکیب این یادگیرهای ضعیف یک تابع رتبه‌بندی ساخته می‌شود.

در مرحله ساخت رتبه‌بندهای ضعیف، مشابه الگوریتم R-AdaBoost که در ساخت دسته‌بندهایش از درصدی از داده‌ها استفاده می‌شود، به جای استفاده از کل داده‌های آموزشی، از درصدی از آنها، مثلا از ۲۰٪ داده‌های آموزشی که وزن بالاتری دارند، استفاده می‌شود. این پارامتر با R نشان داده می‌شود و درصد انتخاب تصادفی داده‌ها را بیان می‌کند. این ویژگی سبب می‌شود رتبه‌بندهای جدید تمرکز خود را بیشتر بر روی داده‌های با وزن بالاتر قرار دهند (داده‌هایی که به درستی رتبه‌بندی نشده‌اند) و لذا دقت را بالاتر خواهد برد. در الگوریتم AdaRank در ساخت هر رتبه‌بند ضعیف از همه داده‌های ورودی استفاده می‌شود و این علاوه بر زمان‌بر بودن الگوریتم، سبب می‌شود قابلیت یادگیری محلی از بین برود.

روش پیشنهادی مشابه الگوریتم AdaRank مبتنی بر لیست است، لذا داده‌های ورودی آن به صورت برداری از ویژگی‌ها و لیست مرتب شده متناظر با آنها به صورت $\acute{S} = \{x_i, y_i\}_{i=1}^{m}$ خواهد بود. هدف ساخت یک تابع رتبه-بند $f(x)$ است که بتواند یک لیست مرتب شده π بر حسب ویژگی‌های x (ویژگی‌های استخراج شده ازپرس‌و-جوی q و مستند d مرتب شده برای آن) به وجود آورد. برای بررسی ارزیابی این لیست ایجاد شده یک تابع $E(\pi, y)$ تعریف می‌شود. تابع رتبه‌بند باید دقت را بر روی تابع ارزیابی بیشینه کند که این مستلزم کمینه کردن تابع خطا است که در معادله۱ تعریف می‌شود.

معادله (۱) (لیست تمام معادلات در پیوست شماره ۱ ذکر شده است)

تابع $E(\pi, y)$ نشان دهنده ارزیابی لیست مرتب شده π تولید شده توسط تابع یادگیری و لیست اصلی y برای هر عضو از داده آموزشی است. پس $L(f)$ به عنوان تابع خطا روی مجموعه داده ورودی محسوب می‌شود.

حل این تابع خطا با روش‌های بهینه‌سازی مستقیم امکان‌پذیر نخواهد بود، زیرا تابع E در این روش که می‌تواند معیار MAP و یا NDCG باشد ناپیوسته و مشتق ناپذیر است، لذا سعی می‌شود تاحد بالای آن کمینه شود. بدین منظور مشابه [2] از معادله ۲ برای پیدا کردن این حد استفاده می‌شود. پس مسئله بهینه‌سازی، تبدیل به معادله ۳ می‌شود، یعنی به جای بهینه کردن تابع خطا $L(f)$ از حد بالای آن استفاده می‌شود.

معادله(۲) و (۳)

الگوریتم پیشنهادی به صورت تکراری سعی در حل این مسئله جدید بهینه‌سازی می‌کند.

با ورود نمونه‌های آموزشی و تعداد تکرارها (T)، الگوریتم مشابه الگوریتم AdaRank در هر تکرار یک رتبه‌بند ضعیف $h_t(t = 1, .., T)$ می‌سازد و با ترکیب خطی این رتبه‌بندها مدل مرتب‌سازی $f(x)$ را به وجود می‌آورد. تفاوت الگوریتم پیشنهادی با الگوریتم AdaRank در این است که به جای اینکه در هر تکرار از تمام نمونه‌های آموزشی در ساخت رتبه‌بند ضعیف بهره گرفته شود، تنها از R% از نمونه‌ها که وزن بالاتری دارند استفاده خواهد شد. این انتخاب سبب می‌شود تا تمرکز بر روی نمونه‌هایی قرار بگیرد که تا کنون درست رتبه‌بندی نشده‌اند و ضمن کاهش زمان یادگیری، دقت را نیز بالا می‌برد.

Input: $\acute{S} = \{x_i, y_i\}_{i=1}^{m}$ and parameters E and T

Set the Randomness Level R

Initialize $P_1(i) = {}^1/_m$

For t = 1,…, T

- Create weak ranker f_t with weighted distribution P_t from the top R percent of the training data S
- Choose α_t

$$\alpha_t = \frac{1}{2}.ln\frac{\sum_{i=1}^{m} P_t(i)\{1 + E(\pi_i, y_i)\}}{\sum_{i=1}^{m} P_t(i)\{1 - E(\pi_i, y_i)\}}$$

- Create f_t

$$f_t(x) = \sum_{k=1}^{t} \alpha_k f_k(x)$$

- Update P_{t+1}

$$P_{t+1}(i) = \frac{\exp\{-E(\pi_i, y_i)\}}{\sum_{j=1}^{m} \exp\{-E(\pi_j, y_j)\}}$$

End For

Output ranking model: $f(x) = f_T(x)$

شکل۲: الگوریتم پیشنهادی

در ادامه، به منظور بهینه کردن تابع خطا که حد بالایی از معیاری از MAP و یا NDCG تعریف می‌شود، توزیع داده‌های آموزشی P_t تغییر می‌کند. در ابتدا همه نمونه‌ها دارای وزن یکسان هستند. در هر تکرار وزن نمونه‌هایی که توسط تابع $f(x)$ به درستی مرتب نشده است، افزایش می‌یابد، و در نتیجه یادگیری در تکرار بعد تمرکز خود را بر روی ایجاد یک یادگیر ضعیف به منظور رتبه‌بندی نمونه‌های دشوارتر قرار می‌دهد. معیار ارزیابی این تابع یادگیر نیز کمینه کردن تابع خطا است. در هر مرحله به منظور ترکیب این توابع یادگیر ضعیف، به هر کدام از آنها یک وزن α_t اختصاص داده می‌شودکه نشان دهنده درجه اهمیت این تابع یادگیر است. پس روش ترکیب این توابع یادگیر برای ساخت یک مدل رتبه‌بندی، جمع وزن‌دار است.

معادله (۴)

۴- پیاده سازی و ارزیابی

در این بخش به ارزیابی الگوریتم پیشنهادی پرداخته خواهد شد. در ابتدا تنظیمات لازم برای پیاده‌سازی، شامل مجموعه داده مورد استفاده و معیار ارزیابی بیان می‌شود و سپس نتایج آزمایشات مورد بررسی قرار می‌گیرد.

۴-۱- مجموعه داده‌ها

برای ارزیابی الگوریتم پیشنهادی مجموعه داده LETOR نسخه ۳ مورد استفاده قرار می‌گیرد. مجموعه داده LETOR 0[3]از داده‌های TREC توسط محققان ماکروسافت بدست آمده است. بیشتر روش‌های یادگیری رتبه‌بندی از این مجموعه داده برای ارزیابی کارایی روش‌هایشان استفاده می‌کنند. برای ساخت نسخه سوم این مجموعه داده از دو مجموعه داده OHSUMED[21] وGOV. استفاده شده است. در مجموعه داده OHSUMED، ۱۰۶ پرس‌وجو موجود است که در حدود ۱۵۲ مستند به هر کدام از آنها تخصیص داده شده است. از این مجموعه داده در کل ۴۵ ویژگی استخراج شده است.

مجموعه داده GOV. از ترکیب نتایج جستجو در سه حوزه وب شامل Topic Distillation(TD)، HomePage Finding(HP) و Named Page Finding(NP) تشکیل شده است که در طی سال‌های ۲۰۰۳ و ۲۰۰۴ از مجموعه TRECجمع‌آوری شده است. تعداد پرس‌وجوها در هر زیرمجموعه متفاوت است، ولی به طور کلی این مجموعه شامل ۳۵۰ پرس‌وجو است.تعداد ویژگی‌های استخراج شده از این مجموعه داده برابر با ۶۴ ویژگی است.

۴-۲- معیارهای ارزیابی

معیارهایی که کارایی مدل رتبه‌بند را مورد ارزیابی قرار می‌دهند، به صورت مقایسه بین لیست مرتب خروجی از مدل با لیست مرتب شده واقعی موجود عمل می‌کنند. در زمینه بازیابی اطلاعات، معیارهای ارزیابی متفاوتی از جمله NDCG[13]، DCG[14]،MAP[15] مورد استفاده قرار می‌گیرند. در این مقاله از سه معیار NDCG وP@n و MAP استفاده می‌شود. علت انتخاب این معیارها، گستردگی استفاده از آنها در بین پژوهشگران این زمینه است.وقتی یک پرس‌وجوی q_i و مجموعه سندهای D_i تخصیص داده شده به آن به همراه لیست رتبه‌بندی π_i و برچسب y_i وارد می‌شود، می‌توان معیار DCG برای این پرس‌وجو در موقعیت k ام آن را به صورت معادله ۵ تعریف کرد [21].

معادله (۵)

(۱)

که در آن(۰) Gیک تابع بهره و D(۰) تابع تخفیف بر اساس موقعیت است و نیز $\pi_i(j)$ نشان دهنده موقعیت مستند $d_{i,j}$ در رتبه‌بندیπ_i ار است . نرمال شده این معیار به نام NDCGشناخته می‌شود.

معادله (۶)

13. Normalized Discounted Cumulative Gain
14. Discounted Cumulative Gain
15. Mean Average Precision

که در آن $DCG_{max}^{-1}(k)$ به عنوان عامل هنجارسازی معرفی می‌شود و بیشینه DCG در حالتی است که مستندات در بهترین ترتیب مناسب مرتب شده باشند.
متوسط دقت به صورت معادله ۷ تعریف می‌شود[23]
معادله (۷)

که در آن $y_{i,j}$ نشان دهنده سطح ارتباط سند $d_{i,j}$ است که دو مقدار ۰ یا ۱ را به خود اختصاص می‌دهد. $P(j)$ که به آن دقت تا موقعیت سند $d_{i,j}$ برای پرس‌وجوی q_i می‌گویند به صورت معادله ۸ تعریف می‌شود.
معادله (۸)

که در آن $\pi_i(j)$ نشان دهنده موقعیت سند $d_{i,j}$ در رتبه-بندی π_i است. برای محاسبه MAP میانگین AP روی مجموعه تمام پرس‌وجوها محاسبه می‌شود[23].
معیار P@n که نشان دهنده دقت در n سند بالا است، توسط معادله ۸ تعریف می‌شود که به جای j پارامتر n جایگزین می‌شود.

۴-۳- نتایج

برای ارزیابی الگوریتم‌ها از اعتبارسنجی متقاطع پنج دسته-ای[16] استفاده شده است، به این ترتیب که داده‌ها در هر مجموعه داده به ۵ قسمت تقسیم شد و سه دسته برای آموزش، یکی برای اعتبار سنجی و دیگری برای تست استفاده شد. برای ارزیابی نتایج، از تمام ویژگی‌های موجود در مجموعه داده‌ها استفاده شد. این ویژگی‌ها به صورت هنجار شده می‌باشد.
در ابتدا به طور کامل نتایج پیاده‌سازی برای مجموعه داده OHSUMED نشان داده می‌شود و سپس نتایج برای مجموعه داده GOV. مورد بررسی قرار می‌گیرد. این نتایج هم به صورت کلی برای این مجموعه داده و هم به صورت مجزا برای ۶ زیرمجموعه آن ارائه خواهد شد.
در انتها هم مقایسه‌ای از نظر انتخاب درصد نمونه‌های مورد نظر برای آموزش یادگیرهای ضعیف (R) بر روی مجموعه داده‌ها و نیز بررسی تاثیر زمانی اجرای این الگوریتم انجام می‌شود. این مقایسه به خوبی نشان می‌دهد که برای رسیدن به MAP بالا نیازی به استفاده صد درصد داده‌ها نیست و با داشتن درصدی از آنها می‌توان در کنار کاهش چشم گیر زمان به جواب خوبی رسید.
با اعمال الگوریتم پیشنهادی بر روی مجموعه داده OHSUMED در ۵ تکرار، نتایج P@n و MAP برای هر تکه محاسبه شد. این نتایج در جدول ۱ دیده می‌شود.
در جدول ۲، نتایج الگوریتم پیشنهادی و الگوریتم پایه آن یعتی AdaRank آمده است. در این جدول دیده می‌شود که MAP الگوریتم پیشنهادی بر روی این مجموعه داده برابر با ۰,۴۵۲۸ است که در مقایسه با الگوریتم پایه آن که برابر با ۰,۴۳۶۶ بوده است، رشدی برابر با ۳,۵ درصد داشته است. روند رشد الگوریتم پیشنهادی در مورد دقت نیز مشاهده می‌شود.
همین روند در مورد معیار NDCG هم صورت گرفت، به این صورت که با اعمال الگوریتم پیشنهادی بر روی مجموعه داده OHSUMED در ۵ تکرار نتایج NDCG@K برای هر دسته ذخیره شد. این نتایج در جدول ۳ دیده می‌شود نتایج این جدول نشان می‌دهد که در این معیار نیز الگوریتم پیشنهادی به خوبی عمل کرده و توانسته است برتری مطلقی بر الگوریتم AdaRank پیدا کند. نتایج موجود در جدول ۴ به خوبی گواه این مطلب است. همان‌طور که در مورد مجموعه داده OHSUMED عنوان شد، نتایج بدست آمده از الگوریتم پیشنهادی به خوبی توانسته است در معیارهای MAP، P@K و NDCG@K برتری نسبت به نتایج الگوریتم AdaRank داشته باشد. در نتیجه می‌توان گفت این ایده که در هر یادگیری به جای انتخاب همه داده‌ها، بر روی درصدی از آنها که وزن بالاتری دارند، آموزش صورت بگیرد، می‌تواند ما را به نتایج بهتری برساند.
علاوه بر مقایسه الگوریتم پیشنهادی با الگوریتم پایه AdaRank، الگوریتم پیشنهادی با الگوریتم‌های دیگر یادگیری رتبه‌بندی که مبتنی بر یادگیری جمعی هستند نیز مورد مقایسه قرار گرفت. به طور خاص الگوریتم پیشنهادی با الگوریتم‌های مطرح دیگری به نام‌های RankBoost

16. 5-fold cross validation

[14]و RankingForest 0 [15]و MART [11]از نظر معیار MAP بر روی مجموعه داده OHSUMED مورد مقایسه قرار گرفت. نتایج این مقایسه درشکل ۳ نشان داده شده است.

جدول۱: نتایج MAP و Precision الگوریتم پیشنهادی در مجموعه داده OHSUMED

	P@1	P@3	P@5	P@10	MAP
Fold1	۰,۶۳۶۴	۰,۵۳۰۳	۰,۴۷۲۷	۰,۳۸۱۸	۰,۳۵۷۶
Fold2	۰,۷۱۴۳	۰,۵۷۱۴	۰,۵۳۳۳	۰,۵۰۴۸	۰,۴۵۲۹
Fold3	۰,۷۶۱۹	۰,۶۶۷۰	۰,۶۷۵۲	۰,۵۵۷۱	۰,۴۷۲۵
Fold4	۰,۷۱۴۳	۰,۶۳۴۹	۰,۶۰۹۵	۰,۵۷۶۲	۰,۵۱۸۹
Fold5	۰,۷۱۴۳	۰,۶۸۲۵	۰,۶۳۸۱	۰,۵۷۱۴	۰,۴۶۲۳
Avg.	۰,۷۰۸۲	۰,۶۱۷۲	۰,۵۸۵۸	۰,۵۱۸۳	۰,۴۵۲۸

جدول۲ : مقایسه نتایج MAP و Precision الگوریتم پیشنهادی با الگوریتم AdaRankدر مجموعه داده OHSUMED

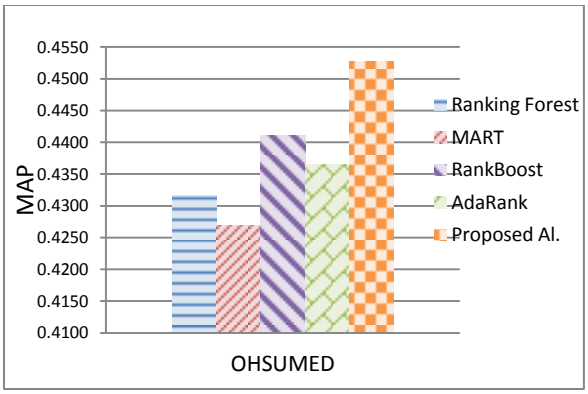
	P@1	P@3	P@5	P@10	MAP
AdaRank	۰,۵۶۸۴	۰,۵۴۶۱	۰,۵۰۵۳	۰,۴۸۶۲	۰,۴۳۶۶
Proposed algorithm	۰,۷۰۸۲	۰,۶۱۷۲	۰,۵۸۵۸	۰,۵۱۸۳	۰,۴۵۲۸

جدول۳: نتایج NDCG الگوریتم پیشنهادی در مجموعه داده OHSUMED

	@1	@3	@5	@10
Fold1	۰,۴۸۴۸	۰,۴۲۷۳	۰,۳۹۸۰	۰,۳۷۹۷
Fold2	۰,۵۸۷۳	۰,۴۸۸۸	۰,۴۸۶۴	۰,۴۶۸۹
Fold3	۰,۶۱۹۰	۰,۵۱۸۱	۰,۵۱۴۶	۰,۴۶۰۵
Fold4	۰,۶۱۹۰	۰,۵۱۳۴	۰,۴۹۸۰	۰,۴۸۸۵
Fold5	۰,۶۱۹۰	۰,۵۹۱۲	۰,۵۴۷۷	۰,۵۰۵۴
Avg.	۰,۵۸۵۸	۰,۵۰۷۸	۰,۴۸۸۹	۰,۴۶۰۶

جدول۴: مقایسه نتایج NDCG الگوریتم پیشنهادی با الگوریتم AdaRankدر مجموعه داده OHSUMED

	@1	@3	@5	@10
AdaRank	۰,۴۷۳۱	۰,۴۴۶۰	۰,۴۲۱۱	۰,۴۲۶۳
Proposed algorithm	۰,۵۸۵۸	۰,۵۰۷۸	۰,۴۸۸۹	۰,۴۶۰۶



شکل۳ : مقایسه نتایج MAP الگوریتم پیشنهادی با سایر الگوریتم‌ها در OHSUMED

همان‌طور که در شکل ۳ دیده می‌شود، الگوریتم پیشنهادی در مجموعه داده OHSUMED به MAP بالاتری نسب به سایر الگوریتم‌ها دست یافته است.در گام بعدی آزمایشات، این روند در مورد مجموعه داده GOV. نیز صورت پذیرفت. با اعمال الگوریتم پیشنهادی بر روی این مجموعه داده نتایج P@n و MAP با الگوریتم پایه AdaRank مورد مقایسه قرار گرفت. نتایج بدست آمده در جدول ۵ نشان می‌دهد که روند رشد الگوریتم پیشنهادی در مورد این مجموعه داده نیز وجود دارد. همین روند در مورد معیار NDCG هم صورت گرفت و نتایج ارائه شده در جدول ۶ نشان دهنده برتری الگوریتم پیشنهادی نسبت به الگوریتم AdaRank است.

نتایج بدست آمده در این مجموعه داده نیز نشان می‌دهد الگوریتم پیشنهادی به خوبی توانسته است در معیارهایMAP، P@K و NDCG@K برتری نسبت به نتایج الگوریتم پایهAdaRank داشته باشد.

در نهایت برای بررسی کارایی الگوریتم پیشنهادی، مقایسه‌ای بین این الگوریتم با الگوریتم‌های دیگر یادگیری رتبه‌بندی که مبتنی بر یادگیری جمعی هستند از نظر معیار MAP صورت گرفت. نتایج برای مجموعه داده استاندارد LETOR3 که از دو مجموعه داده OHSUMED و GOV. است، درجدول ۷ نشان داده شده است. همان‌طور که در جدول ۷ دیده می‌شود، الگوریتم پیشنهادی در مجموعه داده OHSUMED و GOV. نسبت به دیگر الگوریتم‌ها به MAP بالاتری دست یافته است.

مجموعه داده GOV. از ۶ زیر مجموعه تشکیل شده است. برای بررسی دقیق‌تر، نتایج برای این زیر مجموعه داده‌ها نیز به تفکیک بیان خواهد شد. بدین منظور در گام بعدی آزمایشات، برای هر شش زیر مجموعه داده معرفی شده، کارایی الگوریتم پیشنهادی در مقایسه با الگوریتم‌های پایه یادگیری جمعی در یادگیری رتبه‌بندی(RankingForest و MART و RankBoost و AdaRank) مورد ارزیابی قرار گرفت و نتایج آن نظر معیار MAP در جدول ۸ نشان داده شده است.

نتایج نشان می‌دهد که در همه این زیر مجموعه داده‌ها، الگوریتم پیشنهادی از الگوریتم پایه خود، یعنی AdaRank بهتر عمل کرده و به MAP بالاتری رسیده است. این روند در مقایسه با الگوریتم MART نیز مشاهده می‌شود. در مقایسه با الگوریتم‌هایRankBoost و RankingForest، به جز دو مجموعه داده TD2004 و NP2003، الگوریتم پیشنهادی همچنان MAP بالاتری بدست آورده است. لذا برای ادامه کار کارایی این الگوریتم‌ها بر روی دو مجموعه مورد بحث بالا از نظر معیار NDCG هم مورد بررسی قرار می‌گیرد.

در جدول ۹ معیار NDCG@K برای مجموعه داده TD2004 از منظر الگوریتم RankBoost و AdaRank و الگوریتم پیشنهادی مشاهده می‌شود.

جدول۵ : مقایسه نتایج MAP و Precision الگوریتم پیشنهادی با الگوریتم AdaRankدر مجموعه داده GOV.

	P@1	P@3	P@5	P@10	MAP
AdaRank	۰,۴۹۵۰	۰,۲۷۵۱	۰,۱۹۹۳	۰,۱۳۰۵	۰,۵۱۸۵
Proposed algorithm	۰,۵۱۶۷	۰,۲۸۱۱	۰,۲۰۳۳	۰,۱۴۸۲	۰,۵۳۶۹

جدول۶: مقایسه نتایج NDCG الگوریتم پیشنهادی با الگوریتم AdaRankدر مجموعه داده GOV.

	@1	**@3**	**@5**	**@10**
AdaRank	۰,۵۰۳۳	۰,۵۵۴۸	۰,۵۶۵۸	۰,۵۸۰۰
Proposed Algorithm	۰,۵۱۶۷	۰,۵۶۶۸	۰,۵۷۹۶	۰,۵۸۹۴

جدول۷: مقایسه نتایج MAP الگوریتم پیشنهادی با سایر الگوریتم‌ها به صورت تجمیع در مجموعه داده LETOR 3

	OHSUMED	.GOV
Ranking Forest	۰,۴۳۱۶	۰,۵۲۰۹
MART	۰,۴۲۶۹	۰,۴۵۷۸
RankBoost	۰,۴۴۱۱	۰,۴۹۸۸
AdaRank	۰,۴۳۶۶	۰,۵۱۸۵
Proposed Algorithm	۰,۴۵۲۸	۰,۵۳۶۹

جدول۸: مقایسه نتایج MAP الگوریتم پیشنهادی با سایر الگوریتم‌ها در زیر مجموعه داده‌های GOV.

	Ranking Forest	MART	RankBoost	AdaRank	Proposed Algorithm
TD2003	۰,۲۲۰۵	۰,۱۸۷۷	۰,۲۲۳۵	۰۲۴۵۲	۰,۲۵۱۲
TD2004	۰,۲۵۷۲	۰,۱۸۷۶	۰,۲۱۷۸	۰,۱۹۱۴	۰,۲۰۰۴
NP2003	۰,۶۹۵۵	۰,۶۲۲۳	۰,۶۴۹۵	۰,۶۱۸۲	۰,۶۳۸۳
NP2004	۰,۵۸۸۰	۰,۵۱۸۸	۰,۵۵۵۹	۰,۶۰۰۸	۰,۶۵۳۱
HP2003	۰,۷۳۶۹	۰,۷۳۰۹	۰,۷۲۰۴	۰,۷۳۲۳	۰,۷۴۳۴
HP2004	۰,۶۲۷۴	۰,۴۹۹۵	۰,۶۲۵۹	۰,۷۲۲۹	۰,۷۳۵۲

جدول۹:مقایسه نتایج NDCG الگوریتم پیشنهادی با سایر الگوریتم‌ها در مجموعه داده TD2004

	@1	@3	@5	@10
RankBoost	۰,۴۴۰۰	۰,۳۵۷۵	۰,۳۲۲۸	۰,۳۰۶۳
AdaRank	۰,۳۶۰۰	۰,۳۳۹۰	۰,۳۰۷۰	۰,۲۸۲۳
Proposed Algorithm	۰,۴۰۰۰	۰,۳۶۲۲	۰,۳۳۱۴	۰,۳۱۲۲

جدول۱۰:مقایسه نتایج NDCG الگوریتم پیشنهادی با سایر الگوریتم‌ها در مجموعه داده NP2003

	@1	@3	@5	@10
RankBoost	**۰,۵۴۰۰**	**۰,۶۳۴۴**	**۰,۶۷۸۴**	**۰,۶۹۶۴**
AdaRank	**۰,۵۰۶۷**	**۰,۶۰۶۵**	**۰,۶۴۴۵**	**۰,۶۷۰۲**
Proposed Algorithm	**۰,۵۲۶۷**	**۰,۶۱۸۱**	**۰,۶۷۰۰**	**۰,۶۸۸۸**

در این مجموعه داده دیده می‌شود که همان طور که MAP الگوریتم پیشنهادی از الگوریتم AdaRank بیشتر است، در معیار NDCG@k هم باز الگوریتم پیشنهادی توانسته به خوبی با این الگوریتم رقابت کند. در مورد الگوریتم RankBoost مشاهده می‌شود که در معیار NDCG، الگوریتم پیشنهادی توانسته است به کارایی بهتری دست پیدا کند و به جز در NDCG@1 در بقیه برتری با الگوریتم پیشنهادی بوده است. پس در این مجموعه داده، اگر چه الگوریتم پیشنهادی در MAP نسبت به RankBoost بالاتر قرار نگرفت، ولی در معیار NDCG توانست به خوبی با الگوریتم RankBoost رقابت کند.

همین آزمایش در مورد مجموعه داده NP2003 هم انجام شده است که نتایج در جدول ۱۰ مشاهده می‌شود.

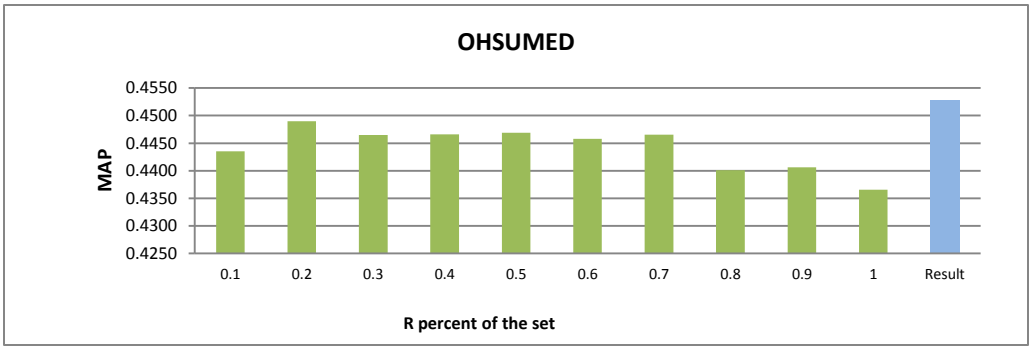
در این مجموعه داده مشاهده می‌شود که همان طور که MAP الگوریتم پیشنهادی از الگوریتم AdaRank بیشتر است، در معیار NDCG@k هم باز الگوریتم پیشنهادی توانسته به خوبی با این الگوریتم رقابت کند. در مورد الگوریتم RankBoost مشاهده می‌شود که در معیار NDCG، باز هم نتوانسته با الگوریتم RankBoost رقابت کند. البته مشاهده می‌شود که این اختلاف بسیار کم است.

شایان ذکر است که این روند زیاد دور از انتظار نیست. زیرا همان‌طور که در[24] نیز عنوان شده است، ماهیت این زیرمجموعه‌ها به علت تعداد کم بودن پرس‌وجوها با سایر زیرمجموعه‌ها متفاوت است.

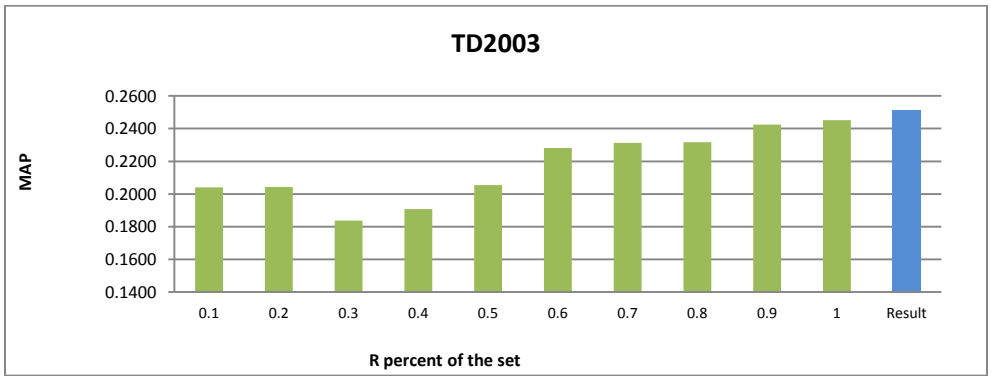
علاوه بر این در تحلیلی که بر روی مجموعه داده استاندارد LETOR3 انجام گرفت، عنوان شد که دو مجموعه داده TD2004 و NP2003 دارای رفتار یکسانی برای تمام الگوریتم‌ها نمی‌باشند. به عبارتی مابقی مجموعه داده‌ها رفتار مشابهی از نظر روند رشد در مورد الگوریتم‌ها نشان می‌دهند و در بازه نزدیک قرار می‌گیرند، ولی این دو مجموعه داده دارای اختلاف MAP بالایی هستند[25].

پس همان‌طور که در جدول ۷ و ۸ دیده می‌شود، الگوریتم پیشنهادی اگر چه بر روی دو زیر مجموعه موفق نبوده است، اما بر روی مجموعه داده LETOR 3 به صورت تجمیع، به بالاترین MAP نسبت به سایر الگوریتم‌ها دست یافته است.

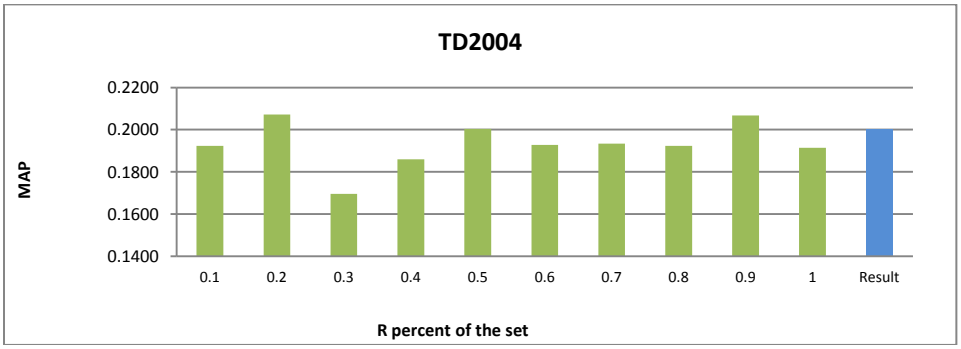
فاز بعدی، بررسی انتخاب معیار R در مجموعه داده‌ها است. بدین منظور در مجموعه داده‌ها، ابتدا آزمایش‌ها بر روی کل مجموعه داده‌ها با Rهایی با فاصله 0.1 انجام شد و سپس برای هر مجموعه داده و با استفاده از مجموعه داده Validation بهترین R انتخاب شد. نتایج در شکل‌های ۴ تا ۱۰ دیده می‌شود.



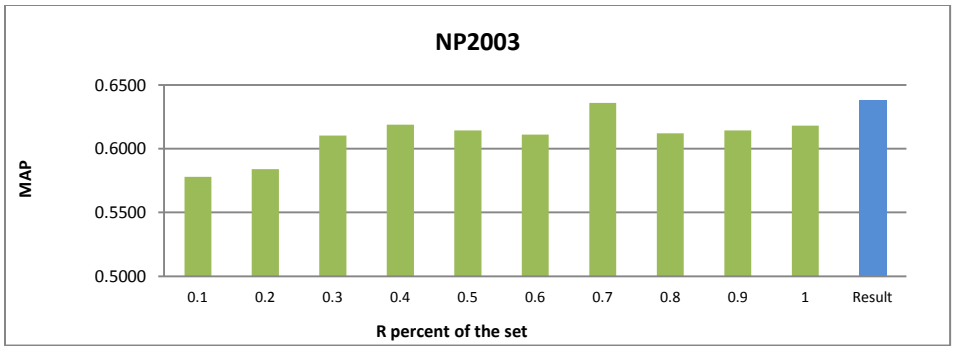
شکل۴: نتایج Rهای متفاوت در مجموعه داده OHSUMED



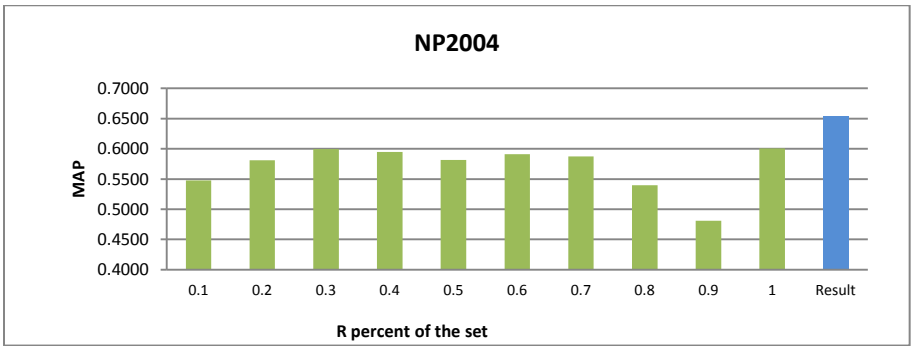
شکل۵: نتایج Rهای متفاوت در مجموعه داده TD2003



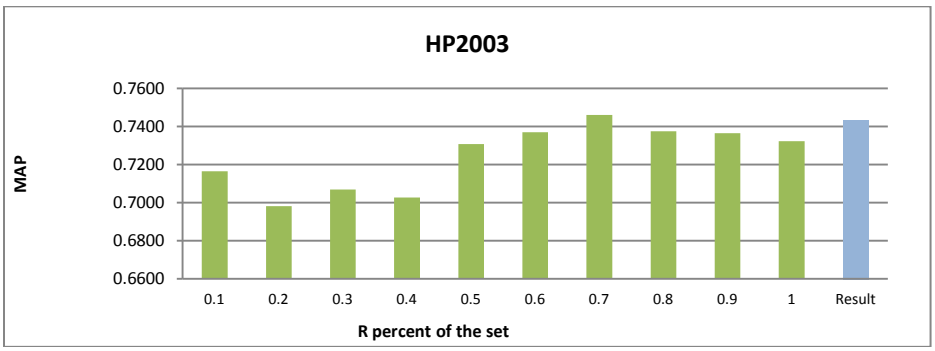
شکل۶: نتایج Rهای متفاوت در مجموعه داده TD2004



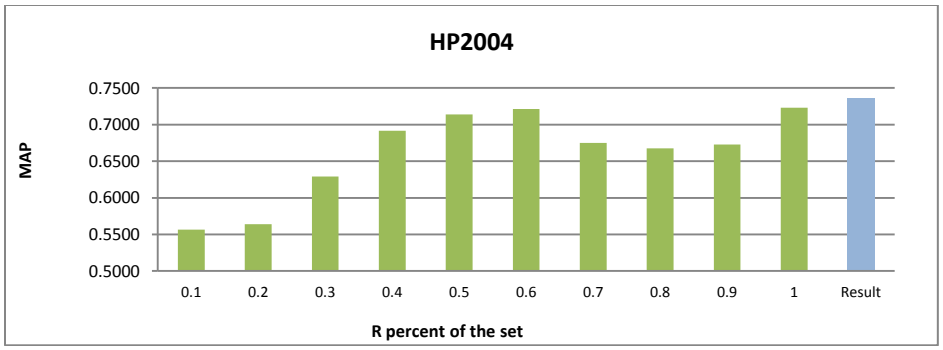
شکل۷: نتایج Rهای متفاوت در مجموعه داده NP2003



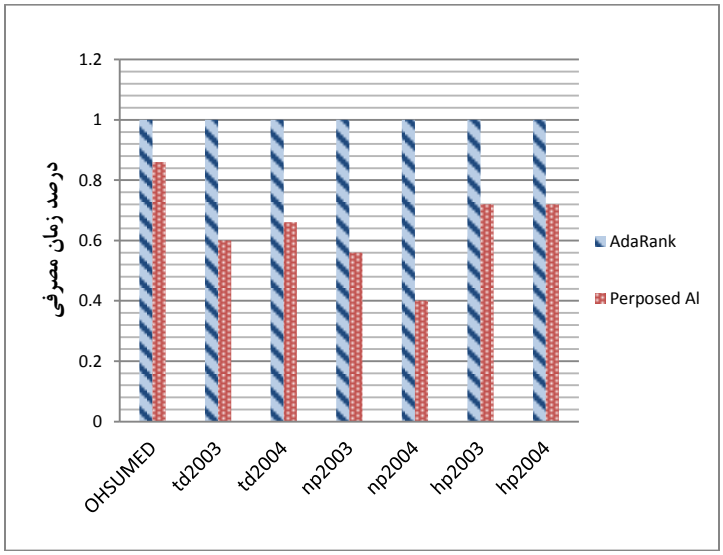
شکل۸: نتایج Rهای متفاوت در مجموعه داده NP2004



شکل۹: نتایج Rهای متفاوت در مجموعه داده HP2003



شکل۱۰: نتایج Rهای متفاوت در مجموعه داده HP2004



شکل۱۱: مقایسه نسبت هزینه زمانی الگوریتم پیشنهادی و الگوریتم AdaRank

همان‌طور که در مجموعه داده OHSUMED دیده می شود، برای بدست آوردن MAP بالا استفاده از R=1 یعنی ساخت تابع یادگیر بر روی کل مجموعه داده‌ها نیاز نیست و تنها بهره‌گیری از درصد کمتری از داده‌ها MAP را افزایش خواهد داد.

از این نمودارها دو نتیجه بسیار مهم گرفته می‌شود، یکی این که برای بدست آوردن MAP بالا نیاز نیست کل داده‌ها مورد استفاده قرار بگیرد و می‌توان در هر مرحله مطابق الگوریتم پیشنهادی بهترین داده‌ها نگه داشته شوند و الگوریتم بر روی آنها کار را به جلو ببرد.

این نتیجه از این جهت مطلوب است که در روش‌های جستجو، کار بر روی حجم عظیمی از داده‌ها بسیار سنگین و وقت‌گیر خواهد بود.

برای نمونه در شکل ۱۱ نسبت زمان مصرفی الگوریتم پیشنهادی در مقایسه با الگوریتم AdaRank دیده می‌شود. برای نمونه اگر مجموعه داده NP2004 توسط الگوریتم AdaRank زمانی برابر با ۱ واحد مصرف کند، این زمان توسط الگوریتم پیشنهادی حدود ۰,۴ خواهد بود. برای نمایش بهتر، نسبت زمان الگوریتم پیشنهادی به الگوریتم AdaRank مد نظر قرار گرفته است

.همان‌طور که مشاهده می‌شود در اکثر این مجموعه داده‌ها، به عنوان نمونه در مجموعه داده OHSUMED، اگر از همه مجموعه داده استفاده شود، نه تنها جواب خوبی حاصل نمی‌شود، بلکه MAP پایین‌تری به دست می‌آید، در حالی که انتخاب مثلا ۲۰٪ داده‌ها جواب مناسبی را بدست خواهد آورد.علت می‌تواند این باشد که اگر از تمام داده‌ها در ساخت یادگیر ضیعف استفاده شود، با وجود وزن‌دار بودن باز هم تمرکز بر روی تمام داده‌ها قرار می‌گیرد، ولی اگر از درصدی از داده‌ها استفاده شود، این تمرکز بر روی همان درصد در ساخت یادگیر خواهد بود.

نتیجه مهم دیگری که از این آزمایش‌ها بدست آمد این بود که بهتر است به جای اینکه از R% مشخص از مجموعه داده‌ها استفاده شود، در هر مرحله از آزمایش بر حسب مجموعه اعتبار سنجی این R انتخاب و بر طبق آن الگوریتم ادامه یابد. مشاهده می‌شود که در اکثر این مجموعه داده‌ها، نتایج (Result1) به MAP بالاتری دست پیدا کرده است. این نتایج حاصل از انتخاب R مناسب برای هر قسمت از مجموعه داده است.

۵- نتیجه‌گیری و کارهای آینده

مسئله رتبه‌بندی از جایگاه ویژه‌ای برخوردار است و کاربردهای متنوعی در بازیابی اطلاعات، موتورهای جستجو و پردازش زبان دارد و لذا امروزه مورد توجه بسیار از پژوهشگران واقع شده است. یکی از مباحث جدید در رتبه-بندی، استفاده از الگوریتم‌های یادگیری ماشین برای یادگیری رتبه‌بندی است، به این صورت که با استفاده از یک سری داده‌آموزشی که نشان دهنده رتبه‌بندی یک سری اشیا است، سعی شود یک مدل رتبه‌بندی برای مرتب‌سازی اشیا جدید ارائه شود که به بهترین نحوه ممکن آنها را رتبه‌بندی کند.اکثر الگوریتم‌های یادگیری ماشین که در دسته‌بندی داده‌ها مورد استفاده قرار می‌گیرند می‌توانند در یادگیری رتبه‌بندی مورد استفاده واقع شوند. دسته‌ای از این الگوریتم-ها، الگوریتم‌هایی هستند که مبتنی بر یادگیری جمعی هستند، یعنی برای دسته‌بندی داده‌ها از چندین دسته‌بند و ترکیب آنها استفاده می‌کنند.

این الگوریتم‌ها به علت بهره‌گیری از چندین دسته‌بند، در اکثر مواقع نتایج دقیق‌تر و مقاوم‌تری تولید می‌کنند. لذا استفاده از این الگوریتم‌ها در یادگیری رتبه‌بندی مورد توجه واقع شده است.

مبتنی بر الگوریتم‌های موجود در دسته لیست، الگوریتم پیشنهادی ارائه شد که این الگوریتم به صورت تکراری یادگیرهای ضعیفی بر روی داده‌های آموزشی که توزیع آنها بر اساس یادگیر قبلی تغییر یافته است، ساخته می‌شود و جمعی از یادگیرهای ضعیف را برای دسته‌بندی تولید می-کند. در مرحله ساخت رتبه‌بندهای ضعیف به جای استفاده از کل داده‌های آموزشی، از درصدی از آنها استفاده می‌شود. با بررسی این الگوریتم بر روی مجموعه داده LETOR3 مشاهده می‌شود که این الگوریتم با ساختن رتبه‌بند بر روی درصدی از داده‌ها، سبب افزایش دقت و کاهش زمان می‌شود. هدف اصلی ارائه این الگوریتم نیز رسیدن به زمان کمتر در حین نگه داشتن دقت بالا بود. نتایج نشان می‌دهد که الگوریتم پیشنهادی نسبت به الگوریتم پایهAdaRank بر روی مجموعه داده LETOR3 به MAP و NDCG بالاتری دست یافته است.

مقایسه الگوریتم پیشنهادی با الگوریتم‌های RankBoost، Ranking Forest و MART که جزو الگوریتم‌های رتبه‌بندی مبتنی بر یادگیری جمعی هستند، نشان می‌دهد که الگوریتم پیشنهادی اگر چه در برخی از زیرمجموعه‌های GOV. مانند TD2003 و NP2003 به MAP بالاتری دست نیافته است، ولی توانسته است میانگین MAP بدست آمده برای مجموعه GOV. را نسبت به الگوریتم‌های دیگر افزایش دهد. همین روند رو به رشد در مجموعه داده OHSUMED دیده می‌شود.

پس به طور کلی می‌تواند عنوان کرد که الگوریتم پیشنهادی ضمن کاهش زمان توانسته به MAP بالاتری در مجموعه داده LETOR3 دست پیدا کند.از جمله کارهایی که در آینده می‌توان انجام داد این است که بتوان روشی برای انتخاب هوشمندانه پارامتر R ایجاد کرد. علاوه بر آن می-توان این روش را در مورد الگوریتم‌های دیگر در زمینه یادگیری رتبه‌بندی هم اعمال کرد و نتایج را بررسی نمود.

پیوست شماره ۱ : معادلات ذکر شده در متن مقاله

$L(f) = \sum_{i=1}^{m}(E(\pi_i^*, y_i) - E(\pi_i, y_i)) = \sum_{i=1}^{m}(1 - E(\pi_i, y_i))$	معادله ۱
$\exp(-x) \geq 1 - x \Rightarrow$	معادله ۲
$\sum_{i=1}^{m} exp(-E(\pi_i, y_i)) \geq \sum_{i=1}^{m}(1 - E(\pi_i, y_i))$	معادله ۳
$f_t(x) = \sum_{k=1}^{t} \alpha_k f_k(x)$	معادله ۴
$DCG(k) = \sum_{j:\pi_i(j)\leq k} G(j)D(\pi_i(j))$	معادله ۵
$NDCG(k) = DCG_{max}^{-1}(k) \sum_{j:\pi_i(j)\leq k} G(j)D(\pi_i(j))$	معادله ۶
$AP = \frac{\sum_{j=1}^{n_i} P(j).y_{i,j}}{\sum_{j=1}^{n_i} y_{i,j}}$	معادله ۷
$P(j) = \frac{\sum_{k:\pi_i(k)\leq\pi_i(j)} y_{i,k}}{\pi_i(j)}$	معادله ۸

منابع

1.Hang Li, Learning to Rank for Information Retrieval and Natural Language Processing, Synthesis Lectures on Human Language Technologies, Morgan & Claypool Publishers, 2011.

2.Jun Xu and Hang Li, Adarank: a boosting algorithm for information retrieval. In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, pages 391–398, 2007.

3.Tao Qin, Tie-Yan Liu, Jun Xu, and Hang Li, LETOR: A benchmark collection for research on learning to rank for information retrieval, Information Retrieval, vol. 13(4), pages 346–374, 2010.

4.AmnonShashua and Anat Levin, Ranking with large margin principle: Two approaches,In Advances in Neural Information ProcessingSystems 15(NIPS 2002), pages 937-944, 2002.

5.Koby Crammer and Yoram Singer, Pranking with ranking, In Advances in Neural Information ProcessingSystems 14(NIPS 2001), pages 641–647, 2001.
6.Ping Li, Christopher Burges, and Qiang Wu, Mcrank: Learning to rank using multiple classification and gradient boosting, In Advances in Neural Information Processing Systems 20(NIPS 2008), pages 897–904, 2008.
7.Koby Crammer and Yoram Singer, Pranking with ranking, In Advances in Neural Information ProcessingSystems 14(NIPS 2001), pages 641–647, 2001.
8.Ralf Herbrich, ThoreGraepel, and KlausObermayer, Large Margin rank boundaries for ordinal regression,Advances in Large Margin Classifiers, pages 115-132, 2000.
9.Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender, Learning to rank using gradient descent. In Proceedings of the 22nd international conference on Machine learning, pages 89–96, 2005.
10.Christopher J.C. Burges, R. Ragno, and Q.V. Le. Learning to rank with nonsmooth cost functions. In Advances in Neural Information Processing Systems 18(NIPS 2006), pages 395–402, 2006.
11.Jerome .H. Friedman, Greedy function approximation: A gradient boosting machine, The Annals of Statistics, vol. 29, pages 1189-1232, 2001.
12.Qiang Wu, Christopher J. C. Burges, Krysta Marie Svore, and JianfengGao, Adapting boosting for information retrieval measures. Information Retrieval, vol. 13(3), pages 254–270, 2010.
13.ZhaohuiZheng, HongyuanZha, Tong Zhang, Olivier Chapelle, Keke Chen and Gordon Sun, A general boosting method and its application to learning ranking functions for web search. In Advances in Neural Information Processing Systems 20(NIPS 2008), pages 1697–1704, 2008.
14.Yoav Freund, Raj D. Iyer, Robert E. Schapire, and Yoram Singer, An efficient boosting algorithm for combining preferences. The Journal of Machine Learning Research, vol. 4, pages 933–969, 2003.
15.St´ephanCl´emencon, Marine Depecker, Nicolas Vayatis ,Ranking Forests,The Journal of Machine Learning Research, vol. 14,pages 39-73, 2013.
16.YisongYue,Thomas Finley, FilipRadlinski and Thorsten Joachims, A support vector method for optimizing average precision, In Proceedings of the 30th annual international ACMSIGIR conference, pages 271–278, 2007
17.Michael Taylor, John Guiver, Stephen Robertson, and Tom Minka, Softrank: optimizing non-smooth rank metrics. In Proceedings of the international conference on Web search and web data mining, pages 77–86, 2008.
18.Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai and Hang Li, Learning to rank: from pairwise approach to listwise approach. In Proceedings of the 24th international conference on Machine learning, pages 129–136, 2007.
19.Y. Freund and R. E. Schapire, A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting, Journal of Computer and System Sciences, vol. 55(1), pages 119-139, 1997.
20.Binxuan SUN, Jiarong LUO, Shuangbao SHU and Nan YU, Approaches to Combine Techniques Used by EnsembleLearning Methods, Journal of Computational Information Systems, vol. 8(1), pages 305–312, 2012.
21.William R. Hersh, Chris Buckley, T. J. Leone and David H. Hickam, OHSUMED: an interactive retrieval evaluation and new large test collection for research, In Proceedings of the 17rd annual international ACMSIGIR conference on Research and development in information retrieval, pages 192–201, 1994.

22.KalervoJärvelin and JaanaKekäläinen, In evaluation methods for retrieving highly relevant documents, In Proceedings of the 23rd annual international ACMSIGIR conference on Research and development in information retrieval, pages 41–48, 2000.
23.Ellen M. Voorhees and Donna Harman, TREC: Experiment and Evaluation in Information Retrieval, MIT, 2005.
24.Jun Xu, Tie-Yan Liu, Min Lu, Hang Li, Wei-Ying Ma: Directly optimizing evaluationmeasures in learning to rank, In Proceedings of the 31rd annual international ACM SIGIR conference on Research and development in information retrieval, pages 107-114, 2008.
25.Guilherme de Castro Mendes Gomes, Vitor Campos de Oliveira, Jussara M. Almeida,Marcos André Gonçalves: Is Learning to Rank Worth it? A Statistical Analysis of Learningto Rank Methods in the